PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS (PERMIS) WORKSHOP

National Institute of Standards and Technology, Gaithersburg, Maryland USA
September 21- 23, 2009

Table of Contents

Technical Sessions

MON-AM1 Model-based Performance Evaluation

MON-AM2 Special Session I: Performance Metrics for Sustainable Manufacturing

MOM-PM1 Performance Assessment and Reliability of Unmanned Syst

MON-PM2 Special Session II: Test and Evaluation of Unmanned and Autonomous Systems

TUE-AM1 Metrics & Measures

TUE-AM2 Special Session III: Is an Agent Theory of Mind Valuable for Adaptive, Intelligent Systems?

TUE-PM1 Ground Truth and Testbeds for Performance Testing

TUE-PM2 Special Session IV: An Ontology for Robotics Science and Systems

WED-AM1 Performance Measures for Mobile Robots

WED-AM2 Special Session V: TRANSTAC: Performance Evaluation of Speech Translation Systems for Military Applications

WED-PM1: Issues in Designing Intelligent Systems

WED-PM2 Special Session VI: Performance Measurements Towards Improved Forklift Safety

*Note: * Presentation Only*

FOREWORD

Welcome to PerMIS'09!

The Performance Metrics for Intelligent Systems (PerMIS) workshop is dedicated to defining measures and methodologies of evaluating performance of intelligent systems. As the only workshop of its kind, PerMIS has proved to be an excellent forum for sharing lessons learned and discussions as well as fostering collaborations between researchers and practitioners from industry, academia and government agencies.

The main theme of the ninth iteration of the workshop, PerMIS'09, seeks to address the question: "***Does performance measurement accelerate the pace of advancement for intelligent systems?***" In addition to the main theme, as in previous years, the workshop will focus on applications of performance measures to practical problems in commercial, industrial, homeland security, and military applications.

The PerMIS'09 program consists of six plenary addresses and six general and special sessions. The topics that are to be discussed by the speakers cover a wide array of themes centered on many intricate facets of intelligent system research. The presentations will emphasize and showcase the interdisciplinary nature of intelligent systems research and why it is not straightforward to evaluate such interconnected system of systems. The three days of twelve sessions will span themes from manufacturing, mobile robotics, human-system interaction, theory of mind, testing and evaluation of unmanned systems, to name a few.

PerMIS'09 is sponsored by NIST, DARPA and NSF, with technical co-sponsorship of the IEEE Washington Section Robotics and Automation Society Chapter, and in-cooperation with the Association for Computing Machinery (ACM) Special Interest Group on Artificial Intelligence (SIGART). The Defense Advanced Research Projects Agency Information Processing Technology Office graciously provided funding to help support the workshop. Special thanks are due to the National Science Foundation for providing funding to allow undergraduate and graduate students to attend PerMIS this year. We also thank Professor Holly Yanco of the University of Massachussetts – Lowell for organizing the student support grants program. We gratefully acknowledge the support of our sponsors.

We thank the special session organizers for proposing interesting topics and assembling researchers related to their sessions. These focused sessions provide an opportunity to delve deeper into specialized topics and to hear from experts in the field. Our thanks are also due to the Program Committee members for publicizing the workshop and the reviewers for providing feedback to the authors, and for helping us to put together an exciting program.

The proceedings of PerMIS will be indexed by INSPEC, Compendex, ACM's Digital Library, and are released as a NIST Special Publication. Selected papers from last year's PerMIS have been published as an edited book volume by Springer Publishers entitled *Performance Evaluation and Benchmarking of Intelligent Systems* (Eds. Raj Madhavan, Edward Tunstel and Elena Messina). The book presents a detailed and coherent picture of state-of-the-art, recent developments, and further research areas in intelligent systems by drawing from the experiences and insights of experts gained both through theoretical development and practical implementation in a variety of diverse application domains. The book will be available for your perusal during the workshop.

It is our sincere hope that you enjoy the presentations, the social programs, renew old relationships, and forge new ones at PerMIS'09!

Raj Madhavan
Program Chair

Elena Messina
General Chair

SPONSORS

NIST
National Institute of Standards and Technology

DARPA

NSF

IEEE

acm

PROGRAM COMMITTEE

General Chair:

Elena Messina (Intelligent Systems Division, NIST, USA)

Program Chair:

Raj Madhavan (Oak Ridge National Laboratory/NIST, USA)

Publicity Chair:

Edward Tunstel (Johns Hopkins University-APL, USA)

S. Balakirsky (NIST USA)

G. Berg-Cross (EM & I USA)

F. Bonsignorio (Heron Robots Italy)

R. Bostelman (NIST USA)

S. Carpin (UC Merced USA)

P. Courtney (PerkinElmer UK)

G. Dimitoglou (Hood College USA)

J. Evans (John Evans, LLC USA)

P. Evans (SWRI USA)

J. Gunderson (GammaTwo USA)

L. Gunderson (GammaTwo USA)

S. K. Gupta (UMD USA)

J. Hallam (SDU Denmark)

T. Kalmar-Nagy (Texas A & M USA)

R. Lakaemper (Temple Univ USA)

G. Moses (UMD USA)

A. del Pobil (Univ. Jaume-I Spain)

T. Potok (ORNL USA)

E. Prassler (UApp Sci-BRS Germany)

F. Proctor (NIST USA)

D. Prokhorov (Toyota USA)

L. Reeker (NIST USA)

M. Reggiani (UPadua Italy)

C. Schlenoff (NIST USA)

M. Shneier (NIST USA)

J. Steele (CSM USA)

PLENARY SPEAKER

Prof. Raffaello D'Andrea

ETH, Zurich, Switzerland

Towards a Ten Thousand Mobile Robot Warehouse

Mon. 8:30 am

ABSTRACT

Order fulfillment is a multi-billion dollar business. Existing solutions range from the highly automated, whose cost effectiveness is inversely related to their flexibility, to people pushing carts around in warehouses manually filling orders, which is very flexible but not very cost effective. In this talk I will describe a radical new approach to order fulfillment that is both flexible and cost effective. The key idea is to use hundreds of networked, autonomous mobile robots that carry inventory-storing pods to human operators. The result is a distribution facility that is dynamic, self-organizing, and adaptive.

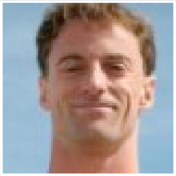
Various challenges had to be overcome in order to make this an economically viable system, ranging from design of robust autonomous mobile robots, real-time wireless control of hundreds of moving agents, the coordination of these agents, and the design of various algorithms that allow the system to adapt and reconfigure itself based on the environment and operating conditions. I will discuss these challenges and how they scale to future warehouses with thousands—not just hundreds—of mobile robots.

BIOGRAPHY

Raffaello D'Andrea received the B.Sc. degree in Engineering Science from the University of Toronto in 1991, and the M.S. and Ph.D. degrees in Electrical Engineering from the California Institute of Technology in 1992 and 1997. He was an assistant, and then an associate, professor at Cornell University from 1997 to 2007. He is currently a full professor of automatic control at ETH Zurich. He is also a founder of, and chief scientific advisor for, Kiva Systems.

He is a co-recipient of the 2008 IEEE/IFR Invention and Entrepreneurship Award, a United States Presidential Early Career Award for Science and Engineering, and was the faculty advisor and system architect of the Cornell Robot Soccer Team, four-time world champions at the international RoboCup competition in Sweden, Australia, Italy, and Japan. He is a recipient of two best paper awards from the American Automatic Control Council and the IEEE, a National Science Foundation Career Award, and several teaching awards in the area of project-based learning. A creator of dynamic sculpture, his work has appeared at various international venues, including the National Gallery of Canada, the Venice Biennale, the Luminato Festival, Ars Electronica, and ideaCity.

PLENARY SPEAKER



Mr. David Bruemmer

5D Robotics, Inc., USA

Measuring the Benefits of Intelligent Behavior for Robotic Threat Detection

Mon. 2:00 pm

ABSTRACT

For robotic applications in hazardous, critical environments, the intelligence needed to provide functional value (i.e. reduced time, increased probability of detection, increased hazard source localization accuracy) cannot be derived from a single behavior (such as obstacle avoidance, mapping, or mine detection). Rarely do we find an integrated suite of capabilities that is capable of accomplishing an end-to-end mission. Intelligence requires not simply behavior, but also the ability to use behaviors effectively towards a highly complex set of real-world, mission-level requirements. If the level of robot initiative and autonomy used in real-world missions is to increase, the underlying mechanisms for behavior composition and human interaction must also change.

Many approaches to creating behaviors as well as orchestrating them have been offered by the community including a variety of machine learning based techniques. These methods and algorithms are often highly elegant, formalized methods intended to streamline the development and testing methodologies. Unfortunately, these all too often fail to provide truly intelligent systems that provide value in the real world. Why is this?

One clue may be found if we consider biology. Is there anywhere in biology where we can find an elegant, formalized, understandable method for behavior composition? Functional intelligence may be, in part, derived from many interwoven heuristics for sequencing and interleaving behavior. In the brain these heuristics are learned over time through experience and perhaps not in an elegant fashion. Artificial Neural Networks (ANNs) are intended to model the behavior derivations we find in biology, but although ANNs allow us to effectively capture particular perceptual and action pairings, we are still left with the fundamental problem of how to sequence and compose behaviors to get a real job done. Without this behavior composition, we may have capability, but enjoy meager intelligence.

Although this talk will not submit a solution to this fundamental challenge, I would like to share a variety of experiments which, over the past few years, have allowed us to metric various components of intelligence for mobile robots used in a variety of real world missions. These missions include chemical plume localization, radiological characterization, urban search and rescue, mine detection and defeat of improvised explosive devices. To accomplish end-to-end missions in the hands of operators with no or little experience with robots requires a means to fuse components of robot intelligence while hiding the behavioral complexity from the user.

The Robot Intelligence Kernel (RIK) is being used to coalesce software components for perception, communication, behavior, world modeling, and human interaction into a single behavior architecture that can be easily transferred for use with a wide variety of robots and sensor-suited, low-level proprietary controls. This talk will discuss implementation strategies employed to integrate these components into a functional system that provides high-performance utility for various real-world tasks. Of particular interest is the cognitive glue, a fuzzy logic rule base, used to sequence and blend these behaviors into mission-level capabilities, such as minesweeping or radiological characterization. Lastly, the paper discusses agents within the interface that fuse various forms of robot and world representation. The interface agents also filter and interpret human input in order to incorporate it seamlessly into the behavioral intelligence of the robotic system. Our strategy is to hide sensor and behavior complexity while providing a means to integrate human intelligence at an appropriate level. In reviewing the benefits and limitations of the RIK approach, the talk will provide system-oriented results from recent hazard detection experiments. In particular, the talk will detail a number of measurements focused on the complete (i.e. human + robot + software + interface) system metrics as well as various component measurements.

BIOGRAPHY

Mr. David J. Bruemmer is Vice President for Research and Development at 5D Robotics, Inc. where he is also a founder and board member. Prior to joining 5D Mr. Bruemmer was Technical Director for Unmanned Vehicles at the Idaho National Laboratory (INL.) For more than 14 years Mr. Bruemmer has enjoyed finding ways to fuse emerging science and engineering into innovative technologies that can change the way robots interact with humans and their environment. He has authored over 50 peer reviewed journal articles, book chapters and conference papers in the area of intelligent robotics. Mr. Bruemmer has been recognized by the President's Office of Science and Technology Policy for his work to forge effective interagency research collaborations across the Federal government (e.g. NASA, Dept. of Energy, Dept. of Defense, Dept. of Commerce, Dept. of Homeland Defense). He is a winner of the R & D 100 Award, the Stoel Reeves Idaho Innovation Award and the Federal Lab Consortium Award for Excellence in Technology Transfer.

The Robot Intelligence Kernel (RIK), developed by Mr. Bruemmer and his team, is being used as a framework for integrating robot software into a standardized, interoperable architecture. Mr. Bruemmer has developed robot behaviors used for a wide variety of robots for applications including remote characterization of high radiation environments, mine sweeping operations, military reconnaissance, IED defeat, chemical plume tracing and search and rescue operations. These efforts have yielded 11 Patents (Issued and Pending) and 10 copyrighted software inventions. His research in the area of countermine operations has demonstrated a four fold decrease in time necessary to find landmines and an improvement of over 20% in probability of detection when compared with the current military baseline. Before working at the INL, Mr. Bruemmer served as a consultant to the Defense Advanced Research Projects Agency, where he worked to coordinate development of autonomous robotics technologies across several offices and programs.

PLENARY SPEAKER

Prof. Ben Kuipers

University of
Michigan, USA

Evaluating the Robot Cognitive Mapper

Tues. 8:30 am

ABSTRACT

A robot observes the space within range of its sensors. In this "small-scale" space, it detects hazards and makes local motion plans. As it explores its global environment, it knits local spatial models together to build a cognitive map—a representation of the global structure of "large-scale" space that extends beyond the sensory horizon of the robot at any given time.

We have developed the Hybrid Spatial Semantic Hierarchy (HSSH), a model of the cognitive map that covers both large-scale and small-scale space, as experienced by the exploring robot. The key idea behind the HSSH is to combine the strengths of multiple different representations (ontologies) for space, each relatively simple: the Local Metrical, Local Topological, Global Topological, and Global Metrical maps.

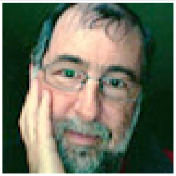
This hierarchy of representations supports a relatively simple and robust way for the robot to construct a useful cognitive map from exploration experience. It also supports robust and efficient planning of routes from one place to another, as well as multiple ontologies for communication between a robot and a human directing it in how to reach a desired destination.

The structure of the HSSH allows us to factor the evaluation task into simpler elements. Each level of the hierarchy can be evaluated according to its ability to meet the needs of the other levels, and the hierarchy as a whole is evaluated according to the different ways it can meet the needs of the robot agent, and how well each of those ways is accomplished. As a result of this factoring, each component is easier to evaluate, and has a lower bar for successful performance.

BIOGRAPHY

Benjamin Kuipers joined the University of Michigan in January 2009 as Professor of Computer Science and Engineering. Prior to that, he held an endowed Professorship in Computer Sciences at the University of Texas at Austin. He received his B.A. from Swarthmore College, and his Ph.D. from MIT. He investigates the representation of commonsense and expert knowledge, with particular emphasis on the effective use of incomplete knowledge. His research accomplishments include developing the TOUR model of spatial knowledge in the cognitive map, the QSIM algorithm for qualitative simulation, the Algernon system for knowledge representation, and the Spatial Semantic Hierarchy model of knowledge for robot exploration and mapping. He has served as Department Chair at UT Austin, and is a Fellow of AAAI and IEEE.

PLENARY SPEAKER



Prof. Paul Cohen

University of Arizona, USA

Against Sophistication: Why Worry About Performance Assessment

Tues. 2:00 pm

ABSTRACT

The theme of the 2009 PerMIS is, “Does performance measurement accelerate the pace of advancement for intelligent systems?” Surely, performance measurement is necessary but not sufficient for the advancement of intelligent systems, and no measurement can compensate for badly designed performance tasks or for performance becoming an end in itself. AI is drunk on performing hard tasks at high levels. Given a choice between power and generality, most of us choose power. Our programs depend on designed exploits, or on designed search spaces in which programs can learn exploits. Divide-and-conquer, specific function, power over generality, and exploits are valuable engineering methods in many disciplines. They are apt to build machines that do one thing well. Human intelligence isn't that kind of machine.

Fixing the current situation will require a disciplined stand against sophistication. It will require investments in general, child-like intelligence, and the investors might not see a return—high performance from cognitive systems—for some time. I think this is a deal worth making, both because it is likely to succeed and because the pursuit of high performance returns low dividends.

BIOGRAPHY

Paul Cohen is Professor and Head of Computer Science at the University of Arizona. Before that he worked at UMass Amherst and the USC Information Sciences Institute. His research is on planning, learning, cognitive development and language. He wrote a textbook on empirical methods for computer science and has worked on the evaluations of several DARPA programs, most recently PAL, Coordinators and Machine Reading.

PLENARY SPEAKER

Dr. Lora Weiss

Georgia Institute of Technology, USA

Assessing Autonomous Systems As They Evolve

Wed. 8:30 am

ABSTRACT

Today, unmanned systems are operating in-theater with untested collaborative capabilities. The vehicles are heterogeneous, in that they are developed by different contractors, they have different levels of autonomy, they have different sensors and capabilities, and they are physically disparate. Unmanned air vehicles built by one contractor have never autonomously collaborated with unmanned sea surface vehicles built by another contractor, and no one knows how they would perform if deployed together today. Their integrated use, however, is rapidly growing in the military. As improvements in autonomy, sensing, and reasoning advance, collaborating, multi-vendor unmanned systems will be increasingly employed to support challenging, tactical operations. The anticipated increase in sophistication drives the need for an ability to robustly test, measure, and evaluate heterogeneous unmanned vehicles for full spectrum dominance and joint operations. We need to consider assessment methods to evaluate force-on-force and mission level the effectiveness of disparate unmanned systems collaborating in theater-wide scenarios. A key requirement for assessing autonomous unmanned systems is the realization that unmanned vehicles pose new challenges that are distinct from traditional approaches to assessing systems. These challenges stem from the upcoming capabilities of unmanned systems being able to autonomously collect and process data, turn it into valued information and knowledge, and then intelligently act upon it with little to no operator involvement. Autonomy at the individual vehicle level involves transitioning cognition into decisions that drive actions. Based on the mission or operational environment, these unmanned systems may execute behaviors that cannot be precisely predicted. Assessments need to support evaluation of autonomous vehicle actions and judge whether the actions are reasonable and acceptable, without having precisely quantifiable metrics. Evaluating these systems will focus more on capabilities and missions rather than mechanics. New approaches to measuring their effectiveness will be adopted to support advances in autonomy and cognition, where the metrics and methods evolve and adapt, just as the systems do.

BIOGRAPHY

Dr. Lora G. Weiss is a lab Chief Scientist at the Georgia Tech Research Institute, where she conducts research on the design, development, and implementation of autonomy and control for manned and unmanned systems. She has supported intelligent autonomy for unmanned underwater vehicles, unmanned air vehicles, and unmanned ground vehicles, and is currently engaged in research in exploring all aspects of the behavior of these systems. Dr. Weiss has chaired sessions at IEEE conferences, ASA conferences, and Navy Symposiums and currently chairs the ASTM Standards Development Subcommittee F41.01, on Unmanned Maritime Vehicle Autonomy and Control. Dr. Weiss is on the Board of Directors for AUVSI, the world's largest non-profit unmanned systems organization. She has developed a video for IEEE Educational Services and has received several publication awards. Dr. Weiss has been Principal Investigator on numerous DoD programs sponsored by offices such as DARPA, the Office of Naval Research, and various Navy Program Executive Offices. She has provided over 150 technical briefs to high-ranking DoD officers and DoD technology offices.

VIDEO SESSION

Dr. Gary Berg-Cross

Knowledge Strategies, USA

Developmental Robotics in Theory and Action: a new way to Understand Cognition and Build Robots with Adaptive Abilities?

Mon. 12:45 - 1:15 pm

This video session serves as an introduction to the topic of developmental robotics (DR). It also serves to discuss some topics in the broader field of cognitive development, which can be explored by the DR research program. DR is a newly emerging interdisciplinary field that builds on 2 of the best tools we have to study cognition –robots and computer modeling. DR studies how autonomous robots can acquire/construct skills, processes & knowledge on their own, strictly through their interactions with the surrounding environment. A core idea is that intelligence is not solely explained by innate mechanisms that modularly organize the human brain. Instead the hypothesis is that much of intelligence/cognition results from a much dynamic process constructing cognitive ability through a long personal development involving “embodied interactions” in rich environments.

FOOD FOR THOUGHT: RELEASE OF WHITE PAPER

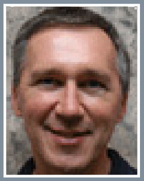
Prof. Erwin Prassler

UAppSci. Bonn-Rhein-Sieg, Germany

The Use of Reuse for Designing and Manufacturing Robots

Tues. 12:45 - 1:30 pm

BANQUET SPEAKER



Prof. Tom Mitchell

Carnegie Mellon University, USA

How does Brain Activity Represent Word Meanings?

Tues. 7:00 pm

Gaithersburg Hilton

ABSTRACT

How does the human brain represent meanings of words and pictures in terms of the underlying neural activity? This talk will present our research using machine learning methods together with fMRI brain imaging to study this question. One line of our research has involved training classifiers that identify which word a person is thinking about, based on their neural activity observed using fMRI. A more recent line involves developing a computational model that predicts the neural activity associated with arbitrary English words, including words for which we do not yet have brain image data. Once trained, the model predicts fMRI activation for any other concrete noun appearing in the text corpus, with highly significant accuracies over the 100 nouns for which we currently have fMRI data. Professor Mitchell’s research was recently featured on a CBS 60 Minutes story “Reading your Mind.”

BIOGRAPHY

Tom M. Mitchell is the E. Fredkin University Professor and head of the Machine Learning Department at Carnegie Mellon University. Mitchell is a past President of the American Association of Artificial Intelligence (AAAI), and a Fellow of the AAAS and of the AAAI. His general research interests lie in machine learning, artificial intelligence, and cognitive neuroscience. Mitchell's web home page is www.cs.cmu.edu/~tom.

September MONDAY 21

08:00	Welcome & Overview
08:30	***Plenary Presentation:*** ***Raffaello D'Andrea*** ***Towards a Ten Thousand Mobile Robot Warehouse***
09:30	Coffee Break
10:00	**MON-AM1 Model-based Performance Evaluation** *Chairs: M. Ani Hsieh & Paul Evans* • Validating Extended Neglect Tolerance Model for Humanoid Soccer Robotic Tasks with Varying Complexities [Rajesh Elara, Carlos Acosta Calderon, Changjiu Zhou, Wijerupage Sardha Wijesoma] • Modeling Multiple Human Operators in the Supervisory Control of Heterogeneous Unmanned Vehicles [Brian Mekdeci, Mary Cummings] • Internal Model Generation for Evolutionary Acceleration of Automated Robotic Assembly Optimization [Jeremy Marvel, Wyatt Newman] • Development of Top-Down Analysis of Distributed Assembly Tasks [Anthony Cowley, M. Ani Hsieh, C.J. Taylor] • Context-Based Object Recognition [Shaun Edwards, Meredith Wright, Ben Abbott]
12:30	Lunch 12:45 - 13:15 Video Session (Gary Berg-Cross)
14:00	***Plenary Presentation:*** ***David Bruemmer*** ***Measuring the Benefits of Intelligent Behavior for Robotic Threat Detection***
15:00	Coffee Break
15:30	**MON-PM1 Performance Assessment and Reliability of Unmanned Systems** *Chairs: Hui-Min Huang & Coire Maranzano* • A Mission Taxonomy-Based Approach to Planetary Rover Cost-Reliability Tradeoffs [David Asikin, John Dolan] • Towards a Systematic Assessment of the Functions of Unmanned Autonomous Systems [Robin Jaulmes, Eric Moline, Laurent Vielle] • Performance Measures Framework for Unmanned Systems (PerMFUS): Initial Perspective [Hui-Min Huang, Elena Messina, Adam Jacoff] • Optimum Combination of Full System and Subsystem Tests for Estimating the Reliability of a System [Coire Maranzano, James Spall]

September MONDAY 21

08:00	Welcome & Overview
08:30	***Plenary Presentation:*** ***Raffaello D'Andrea*** ***Towards a Ten Thousand Mobile Robot Warehouse***
09:30	Coffee Break
10:00	**MON-AM2 Special Session I: Performance Metrics for Sustainable Manufacturing** *Organizers: Kevin Lyons, Mahesh Mani & Ram Sriram* • Manufacturing Unit Process Life Cycle Inventories (Uplci) [Michael Overcash, Janet Twomey, Jacqueline Isaacs] • Conceptual Foundations of Energy Aware Manufacturing [Soundar Kumara] • Discrete Event Simulation to Generate Requirements Specification for Sustainable Manufacturing Systems Design [Björn Johansson, Anders Skoogh, Mahesh Mani, Swee Leong] • Towards A New Geometric Metric for Sustainability Assessment [Gaurav Ameta]
12:30	Lunch 12:45 - 13:15 Video Session (Gary Berg-Cross)
14:00	***Plenary Presentation:*** ***David Bruemmer*** ***Measuring the Benefits of Intelligent Behavior for Robotic Threat Detection***
15:00	Coffee Break
15:30	**MON-PM2 Special Session II: Test and Evaluation of Unmanned and Autonomous Systems** *Organizers: Mauricio Castillo-Effen & Nikita Visnevski* • Unmanned and Autonomous Systems Mission Based Test and Evaluation [Philipp Djang, Frank Lopez] • Modeling and Simulation for Unmanned and Autonomous System Test and Evaluation [Mauricio Castillo-Effen, Nikita Visnevski, Raj Subbu] • Evolutionary Framework for Test of Autonomous Systems [Raj Subbu, Nikita Visnevski, Philipp Djang] • Metrics for Co-evolving Autonomous Systems [Jack Ring]

September TUESDAY 22

08:15	Overview
08:30	***Plenary Presentation:*** ***Ben Kuipers*** ***Evaluating the Robot Cognitive Mapper***
09:30	Coffee Break
10:00	**TUE-AM1 The Role of Robotics Competitions in Advancing Intelligent Systems** *Chairs: Stephen Balakirsky & Jason Gorman* • The Role of Competitions in Advancing Intelligent Systems: A Practitioner's Perspective [Elena Messina, Raj Madhavan, Stephen Balakirsky] • Evaluating The RoboCup 2009 Virtual Robot Rescue Competition [Stephen Balakirsky, Stefano Carpin, Arnoud Visser] • RoboCupRescue Interleague Challenge 2009: Bridging the Gap between Simulation and Reality [Alexander Kleiner, Chris Scrapper, Adam Jacoff] • Mobile Microrobot Characterization through Performance-Based Competitions [Jason Gorman, Craig McGray, Richard Allen]
12:30	Lunch **12:45 - 13:30 Food for Thought: Release of White Paper** ***The Use of Reuse for Designing and Manufacturing Robots (Erwin Prassler)***
14:00	***Plenary Presentation:*** ***Paul Cohen*** ***Against Sophistication: Why Worry About Performance Assessment***
15:00	Coffee Break
15:30	**TUE-PM1 Ground Truth and Testbeds for Performance Testing** *Chairs: Tsai Hong & Barry Bodt* • Data Collection Test-Bed for the Evaluation of Range Imaging Sensors for ANSI/ITSDF B56.5 Safety Standard for Guided Industrial Vehicles [William Shackleford, Roger Bostelman] • Ground Truth Data Using 3D Imaging for Urban Search and Rescue Robots [Nicholas Scott, Alan Lytle] • Performance Measurements of Evaluating Static and Dynamic Multiple Human Detection and Tracking Systems in Unstructured Environments [Barry Bodt, Richard Camden, Harry Scott, Adam Jacoff, Tsai Hong, Tommy Chang, Rick Norcross, Anthony Downs, Ann Virts] • Mathematical Metrology for Evaluating a 6DOF Visual Servoing System [Mili Shah, Tommy Chang, Tsai Hong, Roger Eastman]
18:30	Banquet **19:00 - Banquet Speech** ***How does Brain Activity Represent Word Meanings? (Tom Mitchell)***

PERMIS

PROGRAM

September

TUESDAY 22

08:15	Overview
08:30	***Plenary Presentation:*** ***Ben Kuipers*** ***Evaluating the Robot Cognitive Mapper***
09:30	Coffee Break
10:00	**TUE-AM2 Special Session III: Is an Agent Theory of Mind Valuable for Adaptive, Intelligent Systems?** *Organizer: Gary Berg-Cross* • Is an Agent Theory of Mind (ToM) Valuable for Adaptive, Intelligent Systems? [Gary Berg-Cross] • Towards a Simple Robotic Theory of Mind [Kyung-Joong Kim, Hod Lipson] • Resilient Behavior through Controller Self-Diagnosis, Adaptation and Recovery [Juan Cristobal Zagal, Hod Lipson] • Neurodynamics of Cognition and Consciousness [Robert Kozma, Walter Freeman] • Theory of Mind, Computational Tractability, and Mind Shaping [Tad Zawidzki]
12:30	Lunch 12:45 - 13:30 Food for Thought: Release of White Paper *The Use of Reuse for Designing and Manufacturing Robots (Erwin Prassler)*
14:00	***Plenary Presentation:*** ***Paul Cohen*** ***Against Sophistication: Why Worry About Performance Assessment***
15:00	Coffee Break
15:30	**TUE-PM2 Special Session IV: An Ontology for Robotics Science and Systems** *Organizers: Erwin Prassler & Herman Bruyninckx* • Ontology Formalisms: What is Appropriate for Different Applications? [Craig Schlenoff] • Universal Core Semantic Layer: A Roadmap to Semantic Interoperability [Lowell Vizenor, Barry Smith]
18:30	Banquet 19:00 - Banquet Speech *How does Brain Activity Represent Word Meanings? (Tom Mitchell)*

September WEDNESDAY 23

08:15	Overview
08:30	**Plenary Presentation:** **Lora Weiss** ***Assessing Autonomous Systems As They Evolve***
09:30	Coffee Break
10:00	**WED-AM1 Performance Measures for Mobile Robots** *Chairs: Alan Bowling & Rolf Lakaemper* • Performance Measures of Agility for Mobile Robots [Alan Bowling, Shih-Chien Teng] • Measuring Robot Performance in Real-time for NASA Robotic Reconnaissance Operations [Debra Schreckenghost, Terrence Fong, Tod Milam, Hans Utz] • A Biologically Inspired Sensory Driven Method for Tracking Wind-Borne Odors [Brian Taylor, Brandon Rutter, Roger Quinn] • A Confidence Measure for Segment Based Maps [Rolf Lakaemper] • Evaluation of Robocup Maps [Benjamin Balaguer, Stefano Carpin, Stephen Balakirsky, Arnoud Visser]
12:30	Lunch
14:00	**WED-PM1 Issues in Designing Intelligent Systems** *Chairs: Danil Prokhorov & Satyandra Gupta* • Performance Measurement and Its Role in Advancement for Intelligent Systems: Discussion Points [Danil Prokhorov, Yasuo Uehara] • Collective Intelligence: Toward Classifying Systems of Systems [Alan Ramsbotham] • A Decision-Theoretic Formalism for Belief-Optimal Reasoning [Kris Hauser] • Evaluation of Automatically Generated Reactive Planning Logic for Unmanned Surface Vehicles [Max Schwartz, Petr Svec, Atul Thakur, Satyandra Gupta]
16:00	Coffee Break
16:30	Adjourn

September WEDNESDAY 23

08:15	Overview
08:30	***Plenary Presentation:*** ***Lora Weiss*** ***Assessing Autonomous Systems As They Evolve***
09:30	**Coffee Break**
10:00	**WED-AM2 Special Session V: TRANSTAC: Performance Evaluation of Speech Translation Systems for Military Applications** *Organizers: Craig Schlenoff & Brian Weiss* • Evaluating Speech Translation Systems: Applying SCORE to TRANSTAC Technologies [Craig Schlenoff, Brian Weiss, Michelle Steves, Greg Sanders, Frederick Proctor, Ann Virts] • Development and Internal Evaluation of Speech-to-Speech Translation Technology at BBN [David Stallard, Rohit Prasad, Prem Natarajan] • The Impact of Evaluation Scenario Development on the Quantitative Performance of Speech Translation Systems Prescribed by the SCORE Framework [Brian Weiss, Craig Schlenoff] • Probability of Successful Transfer of Low-Level Concepts via Machine Translation: A Meta-Evaluation [Greg Sanders, Sherri Condon] • Automated Metrics for Speech Translation [Sherri Condon, Mark Arehart, Christy Doran, Dan Parvaz, John Aberdeen, Karine Megerdoomian, Beatrice Oshika] • Utility Assessment in TRANSTAC: Using a Set of Complementary Methods [Michelle Steves, Emile Morse]
12:30	**Lunch**
14:00	**WED-PM2 Special Session VI: Performance Measurements Towards Improved Forklift Safety** *Organizer: Roger Bostelman* • Fork Lift Awareness [Mark Austin] • Where AGV's and Forklifts Roam: Preserving Operational Safety in a Shared Workspace [Richard Ungerbuehler] • Performance Measurements Towards Improved Manufacturing Vehicle Safety [Roger Bostelman, Will Shackleford] • White Paper: Towards Improved Forklift Safety [Roger Bostelman]
16:00	**Coffee Break**
16:30	**Adjourn**

AUTHOR INDEX

ACKNOWLEDGMENTS

These people provided essential support to make this event happen. Their ideas and efforts are very much appreciated.

Thank you PerMIS attendees!

Website and Proceedings

Debbie Russell

Local Arrangements

Jeanenne Salvermoser

Jennifer Peyton

Conference and Registration

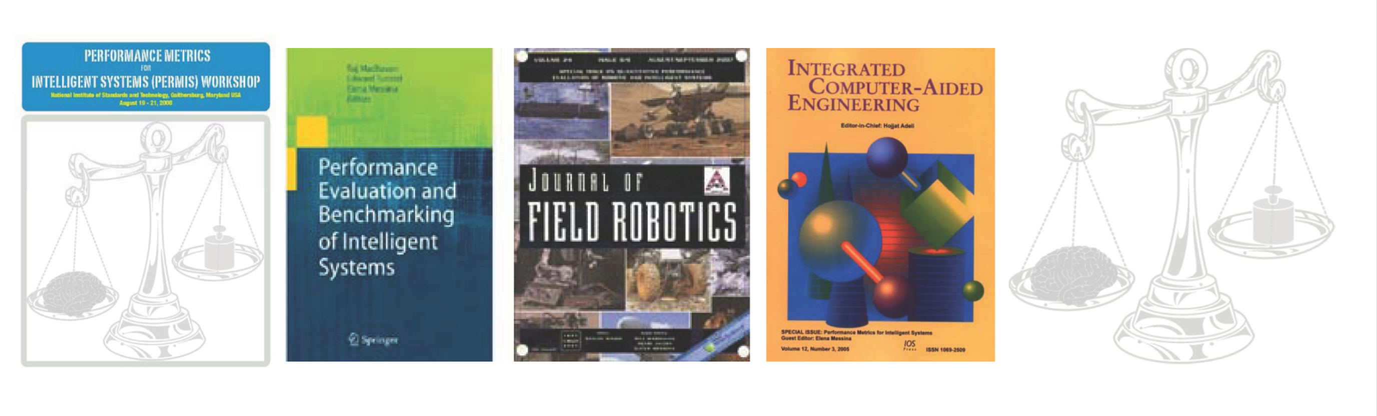
Mary Lou Norris

Angela Ellis

Teresa Vicente

Kathy Kilmer

Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
100 Bureau Drive, MS 8230
Gaithersburg, MD 20899-8230
http://www.nist.gov/mel/isd


PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS (PERMIS) WORKSHOP
National Institute of Standards and Technology, Gaithersburg, Maryland USA
August 19 - 21, 2008
Performance Evaluation and Benchmarking of Intelligent Systems
Springer
JOURNAL OF FIELD ROBOTICS
INTEGRATED COMPUTER-AIDED ENGINEERING
Editor-in-Chief: Hojjat Adeli
SPECIAL ISSUE: Performance Metrics for Intelligent Systems
Guest Editor: Elena Messina
Volume 12, Number 3, 2005
IOS Press
ISSN 1069-2509

Validating Extended Neglect Tolerance Model for Humanoid Soccer Robotic Tasks with Varying Complexities

Mohan Rajesh Elara
Carlos A. Acosta Calderon, Changjiu Zhou
Advanced Robotics & Intelligent Control Centre,
Singapore Polytechnic,
Singapore 139651
MohanRajesh@sp.edu.sg

Wijerupage Sardha Wijesoma
Division of Control & Instrumentation,
School of Electrical & Electronics Engineering,
Nanyang Technological University,
Singapore 639798
eswwijesoma@ntu.edu.sg

ABSTRACT

Estimating robot performance in human robot teams is a vital problem in human robot interaction community. In previous work, we presented extended neglect tolerance model for estimation of robot performance, where the human operator switches control between robots sequentially based on acceptable performance levels, taking into account any false alarms in human robot interactions. Task complexity is a key parameter that directly impacts the robot performance as well as the false alarms occurrences. In this paper, we validate the extended neglect tolerance model for two soccer robotic tasks of varying complexity levels. We also present the impact of task complexity on robot performance estimations and false alarms demands. Experiments were performed with real and virtual humanoid soccer robots across tele-operated and semi-autonomous modes of autonomy. Measured false alarm demand and robot performances were largely consistent with the extended neglect tolerance model predictions for both real and virtual robot experiments. Experiments also showed that the task complexity is directly proportional to false alarm demands and inversely proportional to robot performance.

Keywords

Human robot teams, Robot performance, Task complexity, False alarm demand, Humanoid soccer robots, and Autonomy modes.

1. INTRODUCTION

Growing popularity and increasing viable application domains has contributed to greater presence of robots in the commercial marketplace. Many of these applications require humans and robots to interact closely and work together towards a common goal. Some real life examples of such scenarios include edutainment, service, rescue and surgical robots [1]-[3]. Robot autonomy is an essential component in these applications as it exempts human operators from the time intensive control and decision making processes. Most robotic applications for the commercial market can be categorized into tele-operation and semi-autonomous modes of autonomy. In tele-operation mode, the human operator guides the robot continuously until the given goal is accomplished. This mode requires complete attention of the human operator during the whole operation and every single decision is made by the operator and the robot has zero intelligence [4]. In semi-autonomous mode, the human and the robot collaboratively control parts of the functions required to accomplish the goal. The amount of functions left to the robot depends on the level of robot intelligence, in most scenarios the repetitive, low level tasks are handled by robots and only few high level tasks and decisions making steps are handled by humans [5]. Therefore, the operator work load is greatly reduced in semi-autonomous mode as compared to tele-operation mode. The experiments presented in this paper were performed across tele-operation and semi-autonomous modes of autonomy.

Robot performance in human robot teams is complex and multi-faceted reflecting the capabilities of the robot(s), the operator(s) and the quality of interactions [6]. Neglect tolerance model presented in [7] is used as a general index for estimating robot performance in relation to autonomy in human robot interaction community. This model is employed in [8] to predict the optimized number of robots that should be utilized in human robot teams and robot system effectiveness. Neglect tolerance model is applied in [9] to estimate instantaneous robot performance, evaluate and compare three human robot interaction systems. Neglect tolerance model is applied in [10] to evaluate human robot interaction systems with special focus on the role of a collaborative workspace in enabling mixed initiative interaction between humans and heterogeneous teams of robotic vehicles. Neglect tolerance model is also adopted in [11] to derive model that approximates absolute autonomy and power in agent systems.

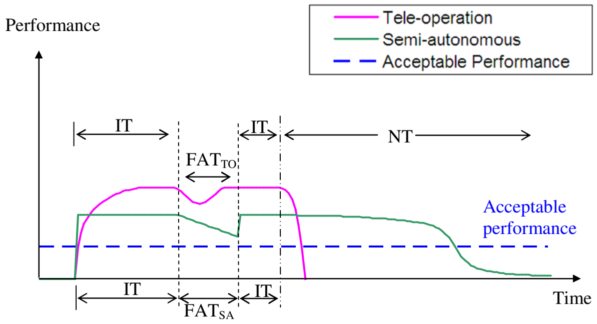
Neglect tolerance model is extended in [12] to investigate human interaction in cooperating human robot teams within a realistically complex environment. Neglect tolerance model assumes ideal conditions while estimating performances, ignoring any false alarms due to erroneous interactions between the human operator and robot. But, in most real life applications erroneous interactions between the human operator and the robot are common due to uncertainties in both human operators as well as in robots. These erroneous interactions lead to false alarms which can be classified into two categories namely, the false positives wherein a robot rejects a "correct" interaction and false negatives wherein a robot fails to reject an "incorrect" interaction. False alarms negatively impact the performance of human robot teams. This zero false alarm assumption results in a less accurate estimation of robot attention demand and robot performance, not only leading to the operator's failure in accomplishing the task as scheduled due to higher attention demands in actual situation, but leading to operator's inability to achieve the performance level set for that task due to the drop in performance attributed to the false alarms. In our earlier work [13] [14], we presented the extended neglect tolerance model to estimate robot performance taking into account the additional demands required due to false alarms. We also showed that extended neglect tolerance model offers better estimations of robot attention demand, and robot performances as compared to neglect tolerance model. For any robotics applications, task complexity is one of the critical factors directly impacting the performance of the robot and occurrence of false alarms in human robot teams. In our previous work, we only experimented and estimated robot performance using the extended neglect tolerance model over a single task complexity and the relationship to task complexity was ignored. But, extended neglect tolerance model can be applied to any robotics application irrespective of the task complexity, robot platform, or domain. Neglect tolerance and interface efficiency are proportional to the task complexity, therefore deriving the influence of latter on robot performance and false alarm demands is necessary for the robot operators to better gauge and optimize resources for the robot task on hand. In this paper, we validate the extended neglect tolerance model for two real and virtual humanoid soccer robotic tasks with varying complexity levels across two levels of robot autonomy. We also present the impact of task complexity on robot performance estimations and false alarms demands.

In this paper, we will first discuss the extended neglect tolerance model. In Section III, we will present a brief description of our real and virtual Robo-Erectus Junior humanoid robots used in the experiments. In Section IV, we will present the experiments involving twenty test subjects to validate the extended neglect tolerance model for two tasks with varying level of complexities across two autonomy modes. Finally, Section V presents some concluding ideas.

2. EXTENDED NEGLECT TOLERANCE MODEL

Neglect tolerance model exploits neglect tolerance and interface efficiency parameters for estimating robot performance in human robot teams [15]-[17]. Neglect tolerance is a measure of how the robot's performance drops over time when the robot is neglected by the user. Interface efficiency is a measure of how the robot's performance varies over time when the robot is being serviced by the human operator. Neglect tolerance model assumes zero erroneous interactions during robot operation while estimating robot performance in human robot teams. But, in most real life situations uncertainties in both human operator and the robots result in erroneous interactions. For example, in a manipulator control task the human operator may select a incorrect co-ordinate points leading to a "false negative" as the manipulator would fail to reject the false interaction or there may be cases where the human operator select a correct co-ordinate points but the robot chooses a wrong co-ordinate points/ignores the human operator controls due to uncertainties in robot software/hardware leading to a "false positive" as the robot rejects a true interactions.

To incorporate the demands due the false alarms, in our earlier work we extended the neglect tolerance model by introducing the notions false alarm time (FAT) and false alarm demand (FAD) as illustrated in Fig. 1. FAT is defined as the time spent over false alarm identification and robot performance recovery to the pre-false alarm level.



IT : Interaction Time
NT : Neglect Time
FAT_{TO} : False Alarm Time (Tele-operation)
FAT_{SA} : False Alarm Time (Semi-autonomous)

Figure 1.Extended Neglect Tolerance Model for Measuring Robot Performance in Human Robot Teams

The scenario depicted in Fig. 1 starts just after the operator starts to service the target robot. The robot performance increases with human operator servicing the robot over time and saturates at some point for both tele-operation and semi-autonomous autonomy modes. In Fig. 1, IT is the period of interaction between human operator and the robot, NT is the period of neglect where the human operator ignores the robot, FAT_{TO} is the time spent over false alarm identification and recovery for tele-operation mode and FAT_{SA} is the time spent over false alarm identification and recovery for semi-autonomous mode. Acceptable performance is the minimum performance level that can be tolerated by the operator for a given task.

False alarms, both false negatives as well as the false positives negatively affect the robot performance, but the FATs are larger for semi-autonomous mode as compared to tele-operation mode due to the delay in the false alarm identification process in the latter. In tele-operation mode, as the operator controls the robot continuously any occurrence of false alarm is identified and rectified in a shorter time period whereas in semi-autonomous mode the operator controls the robot by specifying waypoints for the latter to navigate and so any false alarm that occurs during the neglect period can only be identified and rectified during next period of service thereby resulting in larger FAT.

FAD is the additional demand placed on the robot operator due to false alarms, it is defined as:

$$FAD = \frac{\sum FAT}{IT + \sum FAT} \qquad (1)$$

FAT in Eq. 1 can be expanded as:

$$FAD = \frac{\sum FAT_P + \sum FAT_N}{IT + \sum FAT_P + \sum FAT_N} \qquad (2)$$

where, FATp is the false alarm time contributed by a false positive and FATn is the false alarm time contributed by a false negative. Robot attention demand (RAD) is the robot's average performance over an interaction cycle [23] and extended neglect tolerance model redefines RAD as:

$$RAD = \frac{IT + \sum FAT}{IT + NT + \sum FAT} \qquad (3)$$

Task complexity is a measure of difficulty level of the robot task and it remains as a key factor in deciding the robot performance level and number of occurrences of false alarms. In our experiments in this paper, the task complexity is only a function of static obstacle density. But, it can be further extended to include active obstacle density and terrain factors. We conducted experiments with our real and virtual humanoid soccer robots, Robo-Erectus Junior to validate the extended neglect tolerance model for two tasks with varying complexity levels across tele-operation and semi-autonomous modes of autonomy.

3. ROBO-ERECTUS JUNIOR- A SOCCER PLAYING HUMANOID ROBOT

This section introduces the Robo-Erectus Junior humanoid robot that we employed for our experiments for validating extended neglect tolerance model in this paper. Robo-Erectus Junior is one of the foremost leading soccer playing humanoid robots in the RoboCup humanoid leagues. The objective of the Robo-Erectus Junior development team is to develop a low cost humanoid platform for soccer robotics [18] and human robot interaction [19]. The mechanical structure, electronic control system and gait movement control of Robo-Erectus Junior has evolved through many stages to cope with the increasing complexity of the RoboCup humanoid leagues. Fig. 3 shows the physical design of Robo-Erectus Junior.

Robo-Erectus Junior has been designed to cope with the complexity of a 3 versus 3 soccer game. Robo-Erectus Junior is equipped with three processors each for vision, artificial intelligence and control. Table 1 shows the specification of the processors used in Robo-Erectus Junior. The robot platform is equipped with three sensors: an USB camera to capture images, a tilt sensor to detect a fall, and a compass to detect their direction. The servomotors used send back the feedback data including angular positions, speed, voltage, and temperature. To communicate with its teammates, Robo-Erectus Junior uses a wireless network connected to the artificial intelligence processor. The vision processor performs recognition and tracking of objects of interest including ball, goal, field lines, goal post teammate and the opponents based on a blob finder based algorithm [20]. The further processing of detected blobs, wireless communications and decision making are performed by the artificial intelligence processor which selects and implements the soccer skills (like walk to the ball, pass ball, and dive) the robot is to perform.

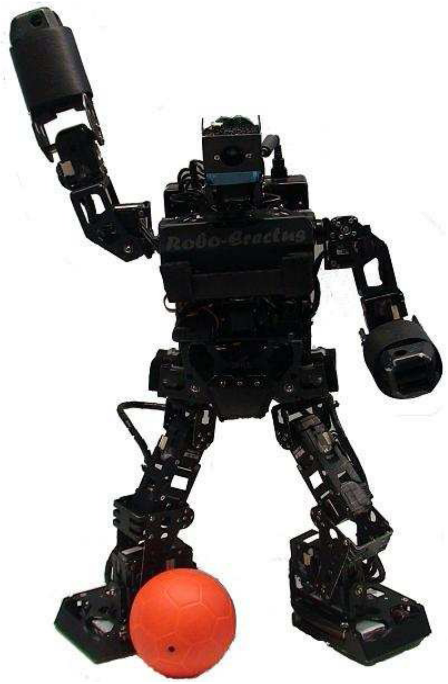


Figure 2.Robo-Erectus Junior, the Latest Generation of the Family Robo-Erectus

Table 1. Processor Specification of Robo-Erectus Junior

Features	**Artificial Intelligence**	**Vision Processor**	**Control Processor**
Processor	Intel ARM XScale	Intel ARM XScale	ATMEL ATmega-128
Speed	400Mhz	400Mhz	16Mhz
Memory	16MB	32MB	4KB
Storage	16MB	16MB	132KB
Interface	RS232, WIFI	RS232, USB	RS232, RS485

Finally, the control processor handles the low level control of motor, based on the soccer skill selected by the artificial intelligence processor. Table 2 shows the physical specifications of Robo-Erectus Junior. It is powered by two high-current Lithium polymer rechargeable batteries, which are located in each foot. Each battery cell has a weight of only 110g providing 12v which means about 15 minutes of operation.

Our Virtual-RE simulator was used to perform experiments with virtual Robo-Erectus Junior humanoid robots providing several possibilities of visualization and interaction with the simulated world [21]. Fig. 3 shows the virtual Robo-Erectus Junior humanoid robot and its environment. Virtual-RE simulator uses the Open Dynamics Engine (ODE) to simulate rigid body dynamics, which has a wide variety of features and has been used successfully in many other projects [22]. OpenGL libraries were used for both visualization and computation of imaging sensory information due to its effectiveness in accommodating modern hardware on a range of platforms. Client-server based architecture was adopted for the realization of the simulator as it allows halting and stepwise execution of the whole simulation without any concurrencies. It also permits detailed debugging of the executed robot software. The simulation kernel models the robots and the environment, simulates sensor readings, and executes commands given by the controller or the user. The graphic user interface not only serves as a tool for interaction between the robot and the user but also visualize the robot status and feedback information whereas the all behavioral controls are handled by the robot controllers.

In each simulation step, the controller reads the available sensors, plans the next action, and sets the actuators to the desired states. Virtual-RE provides each robot with a set of simulated sensors, i.e. tilt, compass, gyroscopes, camera images, and motor feedback. The motor states are also simulated as in the real robot with feedback information that includes the joint angles as well as the velocities of motors.

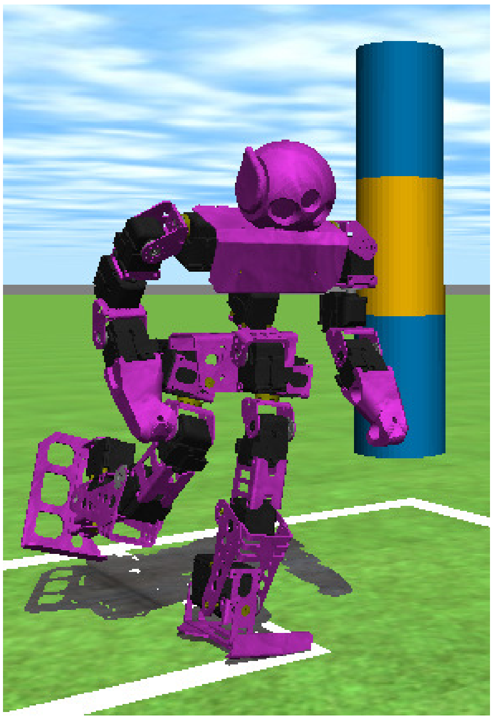


Figure 3. Robo-Erectus Junior in Virtual-RE simulator

Table 2. Processor Specification of Robo-Erectus Junior

Weight	Dimension			Speed
	Height	Width	Depth	Walking
3.2 Kg	480 mm	270 mm	150 mm	5 m/min

4. EXPERIMENTS

4.1 Experimental Design

In these experiments, we validated the extended neglect tolerance model for two different levels of task complexities with our real and virtual Robo-Erectus humanoid robots across tele-operation and semi-autonomous modes of autonomy. We selected the task of navigating Robo-Erectus Junior in the soccer field towards a ball position. The robot and the ball were randomly placed in the soccer field. Operator used the graphic user interface to control the robots so as to navigate to the ball. Upon reaching the ball, the robot was placed at the initial position and the ball at another random position on the field for the next session.

To validate the extended neglect tolerance mode for different task complexity levels, obstacles were placed in the path between the robot and the ball. The complexity was increased by increasing the number of obstacles. The secondary task for the operator was to control a second robot during the neglect time of the target robot so as to collect twice as much data per test session.

4.2 Instantaneous Performance

In this paper, we redefined the instantaneous performance presented in [23] to suit the task under study as ratio of current capability of the robot at a given time to the maximum capability of the robot. Instantaneous performance can take any value between 0 and 1. It is given by:

$$P_I(t) = \frac{C_C(t)}{M_C(t)} \qquad (5)$$

Where Cc(t) is the current capability of the robot at time t for a task, Mc(t) is the maximum capability of the robot or other objects at time t for the same task and PI(t) is the instantaneous performance at time t. In our experiments, the objective for the robot is to navigate to the ball position so the maximum capability would be the distance travelled by the robot moving optimally towards the ball position at top speed. We define maximum capability of the robot as:

$$M_C(t) = K.\delta_t \qquad (6)$$

where δ_t is a small interval of time and K is the maximum speed of the robot. Since, Robo-Erectus Junior humanoid robots can travel at the speed of 8.33cm per second, K value used was 8.33. The current capability of the robot is the actual distance travelled by the robot in the time δ_t,

$$C_C(t) = D_t - D_{t-\delta_t} \qquad (7)$$

where D_t is the distance travelled by the robot at time t and $D_{t-\delta_t}$ is the distance travelled by robot at time t-δ_t. The instantaneous performance is computed as:

$$P_I(t) = \frac{C_C(t)}{M_C(t)} = \frac{D_t - D_{t-\delta_t}}{K.\delta_t} \qquad (8)$$

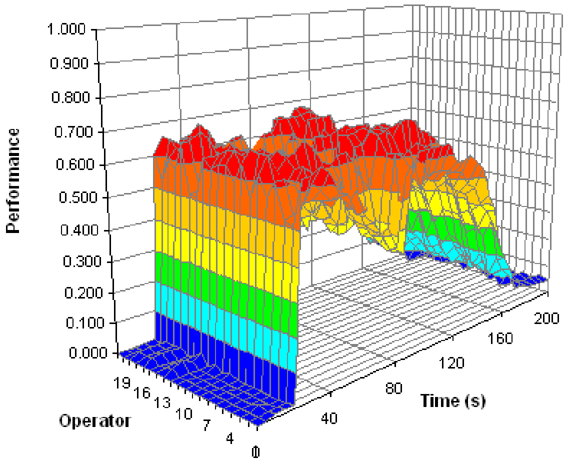
4.3 Participants & Procedure

The test subjects were first trained on the use of the graphic user interface to control the humanoid robots. Sufficient training was provided until the test subjects felt confident in using the user interface upon which the test session with real and virtual robots across tele-operation and semi-autonomous modes of autonomy for the two tasks were conducted. We recruited 20 test subjects aged between 18 and 51 and each of them took part in two 10 minute session with real and virtual robots, so a total of 80 test sessions were performed. Of the 40 sessions each with real and virtual robots, 20 were dedicated to the tele-operation mode and remaining 20 to the semi-autonomous mode.

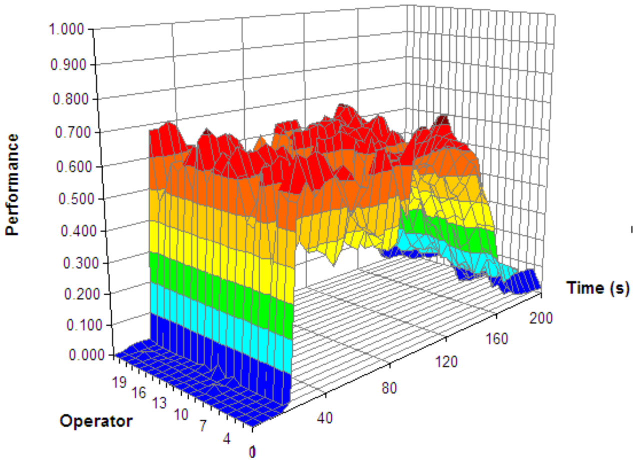
Out of the 20 test sessions for both the autonomy modes, 10 sessions involved task 1 where the human operator navigated the robot to the ball with five obstacles in its path, and the other 10 sessions were dedicated to task 2, where fifteen obstacles were placed in the path between the robot and the ball. In each test session, the operator first serviced the target robot to accomplish the task of navigating to the ball. After servicing the target robot, he/she switches to the secondary task of navigating the second robot to the ball. The operator performed the navigation task as many times as possible with the two robots during each ten minutes test session. The instantaneous performance measurements together with the time, operator controls, and robot state information were recorded for each test session.

4.4 Results

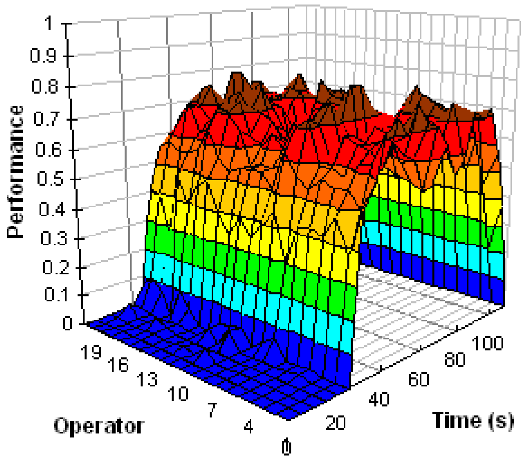
Fig. 4 shows the performance of the real and virtual robots for task 1 across tele-operation and semi-autonomous modes for all the twenty test subjects. False alarms were witnessed in 39 out of the total 40 test sessions with task 1 involving both real and virtual robots across tele-operation and semi-autonomous modes. A total of 68 false alarms were recorded out of which 22 were false negatives and 46 were false positives. Fig. 5 shows the performance of the real and virtual robots for task 2 across tele-operation and semi-autonomous modes for all the twenty test subjects. False alarms were witnessed in all the 40 test sessions with task 2.



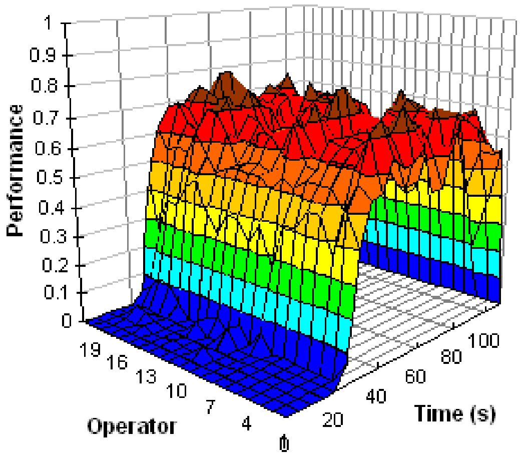
(a)



(b)



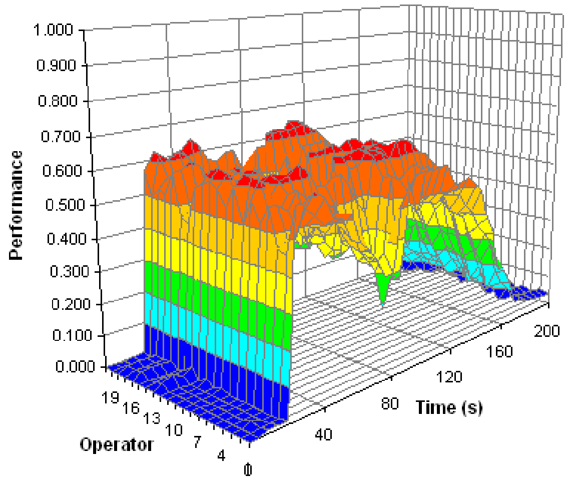
(c)



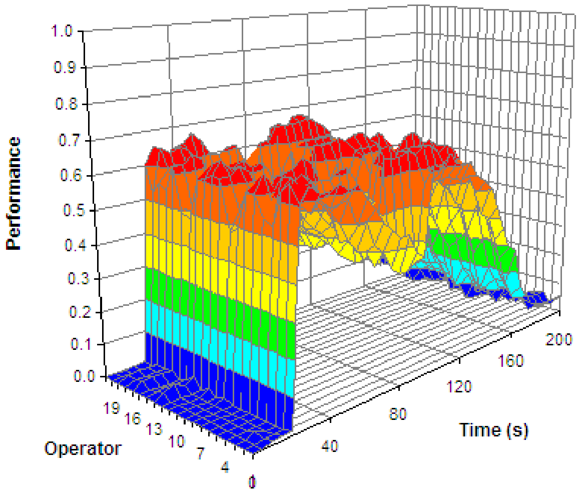
(d)

Figure 4.Robot Performance in Human Robot Teams for Task 1: (a) Real Robot in Semi-autonomous, (b) Virtual Robot in Semi-autonomous, (c) Real Robot in Tele-operation and (d) Virtual Robot in Tele-operation.

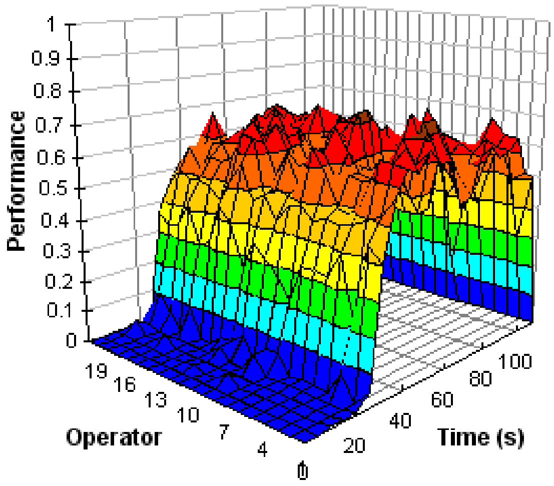
A total of 103 false alarms were recorded out of which 35 were false negatives and 68 were false positives. The false positives were mainly due to errors in the graphic user interaction scheme, software faults in robot's control and artificial intelligence modules, and hardware failures in sensor/actuator systems. The false negatives were mainly due to the human error pertaining to lack of understanding of the interaction scheme, and the task of interest. As postulated in the extended neglect tolerance model, the FADs for tele-operation were found to be shorter than those for semi-autonomous experiments for both task complexities. The performance results for real and virtual robots were found to follow similar pattern for both task complexities across both tele-operation and semi-autonomous modes. From the figures, it is evident that due to the increased complexity in task 2 attributed to the presence of additional ten more obstacles in the path of the robot, the performance has significantly dropped in both real and virtual experiments.



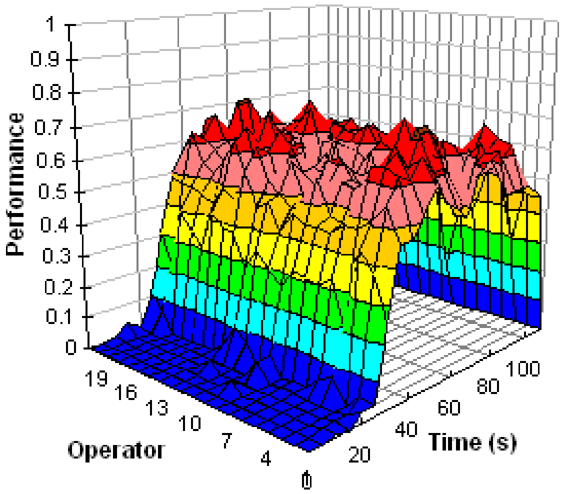
(a)



(b)



(c)



(d)

Figure 5. Robot Performance in Human Robot Teams for Task 2: (a) Real Robot in Semi-autonomous (b) Virtual Robot in Semi-autonomous. (c) Real Robot in Tele-operation Mode, and (d) Virtual Robot in Tele-operation Mode

For both the tasks complexities, tele-operation mode is efficient in increasing the performance of the robot upon servicing after a neglect period as compared to point to point mode. The robot performance dropped abruptly to zero within 2 seconds of neglect period for both the tasks across tele-operation experiments whereas the performance drop during neglect period was more gradual for both the tasks across semi-autonomous experiments. During the neglect period, the rate of performance drop was slower in semi-autonomous mode as compared to tele-operation mode. Occurrence of false alarm degrades performance in all the experimental cases. The performance drop due to false alarms are more prominent in semi-autonomous mode as compared to tele-operation mode as the period for performance recovery to pre-false alarm level is shorter in latter. The trend of the graphs in Fig. 4, and Fig. 5 validate the extended neglect tolerance model for varying levels of task complexities. From the figures, it is evident that irrespective of the task complexity the tele-operation mode requires the operator to interact continuously with the robot as in the case of neglect the robot performance drops rapidly to zero. We also computed and compared the FADs for real and virtual robot experiments across the two autonomy modes for both the tasks. Fig. 6 shows the average FADs for different experimental cases. The mean FAD for task 2 in semi-autonomous mode was highest for both real and virtual robot experiments. It is clear from the figure that increasing task complexity increases the number of occurrence of false alarms and therefore resulting in higher FAD. The number of false alarms increased from 68 in task 1 to 103 in task 2, in specific the increase in false positives was more prominent and the results for real and virtual robot experiments followed the similar patterns. FADs can be used as a performance metric to gauge the additional operator efforts required for tasks of varying complexities and scenarios. Table 3 presents the percentage performance drop in task 2 in comparison to task 1 for all experimental cases.

Table 3. Percentage Performance Drop in Task 2 in Comparison to Task 1 For All Experimental Cases

Experimental Cases	Performance Drop
Real Robot Tele-operation	6.78%
Virtual Robot Tele-operation	10.75%
Real Robot Semi-autonomous	12.71%
Virtual Robot Semi-autonomous	11.27%

From, the table it is evident that an increase in task complexity results in performance drop and deriving this relationship for task of interest can aid robot operator optimize resources and time.

5. CONCLUSION

In this paper, we validated the extended neglect tolerance model for two robot navigation tasks with different complexity levels across tele-operation and semi-autonomous modes of autonomy. Results of our experiments with real and virtual robots were largely consistent with the proposed extended neglect tolerance model predictions for both tasks across the two autonomy modes. FADs were found to be directly proportional to the task complexity, as the results showed that an increase in task complexity resulted in an increase in FAD. Irrespective of the task complexity, FADs were found to be higher in semi-autonomous mode as compared to tele-operation mode for experiments with real and virtual robots. Results from both the tasks showed that tele-operation mode offers higher robot performance than semi-autonomous mode but the latter requires lower RAD and offers better performance deterioration rate during neglect times. The experiments in this paper were limited to a human operator navigating a single robot towards a randomly placed ball. Future work would include extending these results to estimating robot performance in multi-robot teams involving homogeneous and/or heterogeneous robots working together with a human operator towards accomplishing tasks of varying complexities. A second possibility of future work is to use FAD as a metric to compare performances of robot platforms, autonomy modes and interaction schemes for tasks of varying complexities. Another possibility of future work is to study the effects of task complexities on robot performances and FADs for multi-tasking problems.

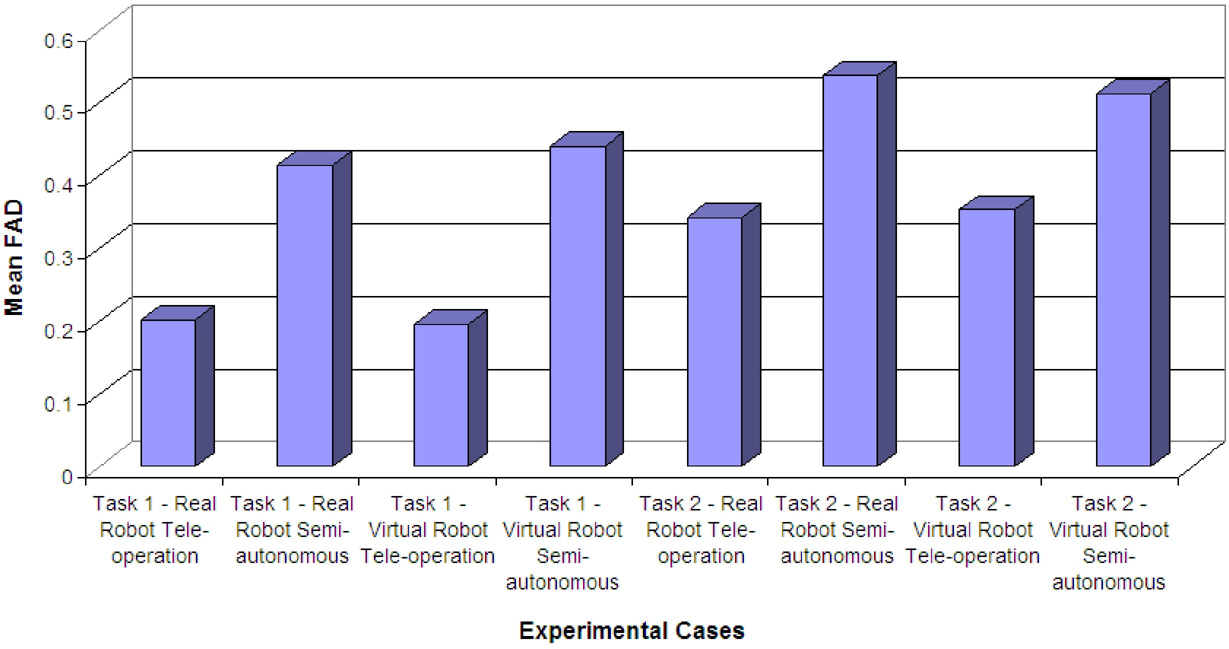


Figure 6. Average FADs for different experimental cases

6. REFERENCES

[1] R. E. Mohan, C. A. Acosta Calderon, C. Zhou, P. K. Yue, “Using Humanoid Robotics for Discovery Based Engineering Learning in Tertiary Institutions,” in Proc. Raffles International Conference on Education, Singapore, 2008.

[2] B. M. Hudock, B. E. Bishop and F. L. Crabbe, “Development of a Novel Urban Search and Rescue Robot,” United States Naval Academy Trident Scholar Report, U.S.A, 2003.

[3] G. S. Fischer, I. Iordachita, C. Csoma, J. Tokuda, S. P. DiMaio, C. M. Tempany, N. Hata, and G. Fichtinger, “MRI-compatible pneumatic robot for transperineal prostate needle placement”, IEEE/ASME Transaction on Mechatronics, pp. 295-305, vol. 3, 2008.

[4] K. W. Ong, G. Seet, and S. K. Sim, “An Implementation of Seamless Human-Robot Interaction for Telerobotics,” Journal of Advanced Robotic Systems, Vol. 5, No. 2, 2008.

[5] M. Paul, and A. Peter, “Shared autonomy in a robot hand teleoperation system,” in Proc. International Conference on Intelligent Robotics Systems, Munich, Germany, 1994.

[6] J. Wang, H. Wang, M. Lewis, P. Scerri, P. Velagapudi and K. Sycara, “Experiments in Coordination Demand for MultiRobot Systems”, in Proc. IEEE International Conference on Distributed Human-Machine Systems, Greece, 2008.

[7] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen, “Validating Human-Robot Interaction Schemes in Multi-Tasking Environments”, IEEE Transactions on Systems, Man, and Cybernetics -- Part-A, Vol. 35, No. 4, Pages 438-449, July 2005.

[8] J. W. Crandall, and M. L. Cummings, “Identifying Predictive Metrics for Supervisory Control of Multiple Robots” IEEE Transactions on Robotics, Vol. 23, No. 5, Oct 2007

[9] J. W. Crandall, and M. A. Goodrich, “Measuring the Intelligence of a Robot and its Interface”, in Proc. NIST's Performance Metrics for Intelligent Systems Workshop, U.S.A, 2003.

[10] D. J. Bruemmer and M. Walton, “Collaborative tools for mixed teams of humans and robots”, in Proc. International Workshop on Multi-robot Systems, Washington, U.S.A, 2003.

[11] H. Hexmoor , “A model of absolute autonomy and power: Toward group effects,” in Proc. AAAI Workshop Autonomy, Delegation, and Control: From Inter-Agent to Groups, Canada, 2002.

[12] J. Wang, “Human control of cooperating robots,” Ph. D dissertation, University of Pennsylvania, Pennsylvania, U.S.A, 2008.

[13] R. E. Mohan, W. S. Wijesoma, C. A. A. Calderon, C. Zhou, “Experimenting false alarm demand for human robot interactions in humanoid soccer robots” International Journal of Social Robotics, Accepted for publication.

[14] R. E. Mohan, W. S. Wijesoma, C. A. A. Calderon, C. Zhou, "False alarm demand: A new metric for measuring robot performance in human robot teams", in Proc. IEEE International Conference on Autonomous Robots and Agents, New Zealand, 2009.

[15] M. A. Goodrich, J. W. Crandall, J. L. Stimpson, " Neglect Tolerant Teaming: Issues and Dilemmas", in Proc. AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments, U.S.A, 2003.

[16] J. W. Crandall, C. W. Nielsen, and M. A. Goodrich, “Towards Predicting Robot Team Performance”, in Proc. IEEE International Conference on Systems, Man, and Cybernetics, 2003

[17] Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. A. Goodrich, “Common metrics for human robot interaction”, in Proc. International Conference on Human Robot Interaction, U.S.A, 2006.

[18] C. Zhou, and P. K. Yue, “Robo-Erectus: A low cost autonomous humanoid soccer robot,” Advanced Robotics, 2004.

[19] R. E. Mohan , C. A. Acosta Calderon, C. Zhou, P.K. Yue, and L. Hu, "Modelling human humanoid robot interaction in soccer robotics domain using NGOMSL", in Proc. 17th IEEE International Symposium on Robot and Human Interactive Communication, Germany, 2008.

[20] R. E. Mohan, C. A. Acosta Calderon, C. Zhou, P. K. Yue, L. Hu, and B. Iniya, "An embedded vision system for soccer playing humanoid robot: Robo-Erectus Junior", in Proc. IEEE International Conference on Signal Processing, Communications and Networking, India, 2008

[21] C. A. Acosta Calderon, R. E. Mohan, and C. Zhou, "A humanoid robotic simulator with application to robocup ", in Proc. IEEE Latin America Robotic Simposium/Congreso Mexicano de Robotica, Mexico, 2007.

[22] R. Smith. (2009, January 23). Open dynamics engine [Online]. Available: http://www.ode.org

[23] J. W. Crandall, “Towards developing effective human robot systems,” M.S. dissertation, Brigham Young University, U.S.A, 2003.

Modeling Multiple Human Operators in the Supervisory Control of Heterogeneous Unmanned Vehicles

Brian Mekdeci
Massachusetts Institute of Technology
77 Massachusetts Ave, Rm 33-407
Cambridge, MA, 02139
1-617-452-3044
mekdeci@mit.edu

M.L. Cummings
Massachusetts Institute of Technology
77 Massachusetts Ave, 33-311
Cambridge, MA, 02139
1-617-252-1512
missyc@mit.edu

ABSTRACT

In the near future, large, complex, time-critical missions, such as disaster relief, will likely require multiple unmanned vehicle (UV) operators, each controlling multiple vehicles, to combine their efforts as a team. However, is the effort of the team equal to the sum of the operator's individual efforts? To help answer this question, a discrete event simulation model of a team of human operators, each performing supervisory control of multiple unmanned vehicles, was developed. The model consists of exogenous and internal inputs, operator servers, and a task allocation mechanism that disseminates events to the operators according to the team structure and state of the system. To generate the data necessary for model building and validation, an experimental test-bed was developed where teams of three operators controlled multiple UVs by using a simulated ground control station software interface. The team structure and inter-arrival time of exogenous events were both varied in a 2x2 full factorial design to gather data on the impact on system performance that occurs as a result of changing both exogenous and internal inputs. From the data that was gathered, the model was able to replicate the empirical results within a 95% confidence interval for all four treatments, however more empirical data is needed to build confidence in the model's predictive ability.

Categories and Subject Descriptors

I.6.3 [**Computing Methodologies**]: Simulation and Modeling – *applications.*

General Terms

Performance, Experimentation, Human Factors.

Keywords

Discrete event simulation, human factors, modeling, team performance, supervisory control, unmanned vehicles.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
Copyright © 2009 ACM 978-1-60558-747-9/09/09...$10.00

1. INTRODUCTION

Unmanned vehicles (UVs) are currently in use for numerous military operations, but they are also being considered for many non-military applications as well, including mining, fighting forest fires, border patrol and supporting police [1]. Currently, several human operators are required to control many of today's UVs, but futuristic systems will invert the operator-to-UV ratio so that one operator can control multiple UVs [2]. To accomplish this goal, the level of automation will have to increase such that operators will give high-level, supervisory instructions to the UVs instead of manual control [3]. However, previous research has shown that even under supervisory control, there is a cognitive limit as to the number of UVs a single human operator can effectively manage [4, 5]. Large, complex, time-critical missions, such as disaster relief, will likely exceed that limit and will require multiple operators, each controlling multiple UVs, to combine their efforts. Since such systems do not currently exist, many questions arise, including: (1) How many operators are necessary to achieve a set of mission objectives? (2) How should the operators combine their efforts in the most effective way? (3) Will the group performance be more than, equal to, or less than the sum of the individual contributions?

2. RESEARCH OBJECTIVE

The goal of this research is to develop a quantitative model of a team of human operators, each performing supervisory control of multiple unmanned vehicles, in time-critical environments. This model would allow stakeholders, such as vehicle designers and battlefield commanders, to vary input parameters, such as vehicle speed and number of human operators, in order to determine their impact on system performance.

3. PREVIOUS RESEARCH

3.1 Queuing Model of Supervisory Control of Unmanned Vehicles

Supervisory control of unmanned vehicles involves an operator handling intermittent events via an automated system by giving high-level commands to UVs. As such, supervisory control of unmanned vehicles has been previously modeled as a queuing system where the vehicles requesting assistance are regarded as users and the human operators are regarded as servers [6]. For instance, in a simple surveillance scenario whose timeline is shown in Figure 1, an unidentified contact suddenly emerges at time *t*. This event, labeled A, requires that the operator perform a task, in this case, assign an UV to the contact location for further investigation. Since this event is not directly controllable by the operator or vehicle, it is considered to be an exogenous event to the system. Ideally, the operator would notice this event and start “servicing” it immediately by performing the associated task. However, because of inherent inefficiencies of human attention, the operator will inadvertently introduce a delay between the arrival of this event and the moment he starts to service it (marked by event B in the timeline). This delay is due to a combination of the Wait Time due to loss of Situational Awareness (WTSA) and the Wait Time due to Interaction (WTI) [4]. WTSA occurs when the operator is not aware that the event requires his attention, whereas WTI occurs when the operator has noticed the event, but has not measurably started the associated task yet (perhaps due to deciding between the right course of action from a number of options). Since it is extremely difficult to separate WTSA from WTI, the measured time between when an event emerges and when the operator starts the associated task (assuming the operator is not busy and has the resources available to service the event) will be considered WTOD – wait time due to operator delay. Cummings and Mitchell [4] have shown that this delay can be quite significant particularly when operators are controlling multiple vehicles simultaneously and have degraded situational awareness.

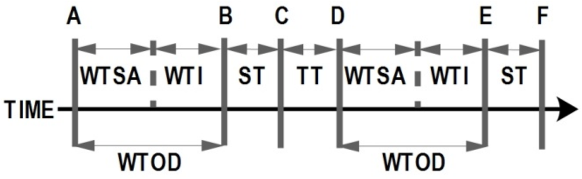


Figure 1: Timeline of events for simple UV scenario.

The task of assigning a vehicle to a location also takes a finite amount of time known as the Service Time (ST). At the moment when the operator finishes assigning a vehicle (C in Figure 1), that vehicle will begin to travel the assigned location. The time during which the vehicle is travelling is referred to as the Travel Time (TT) and in this scenario also represents the Neglect Time (NT) of the vehicle, since the vehicle acts autonomously during this period without requiring the operator’s attention [7]. After some time, the vehicle will eventually arrive at the contact location, denoted by event D. Similar to the time between A and B, the vehicle must wait a finite period of time before the operator begins to interact with the vehicle’s camera, denoted by event E. Finally, after another service time, the operator finishes identifying the contact (labeled event F) which may more may not spawn additional endogenous events, depending upon the scenario. If the final objective of the operator is to simply identify unknown contacts, then the difference in time between event F (when the final objective is met) and event A (when the contact emerged) is known as the Objective Completion Time (OCT). Since time is of the essence in many UV applications, the goal of many UV system designers and decision makers it to minimize the average OCT for a given scenario.

3.1.1 Multiple Event Handling

3.1.1.1 Wait Time due to Queuing

If an operator is busy interacting with a vehicle and another event emerges that requires the operator’s attention, then that event must wait for the operator to become available. This additional time, not represented in Figure 1, is known as the wait time due to queuing (WTQ) since the event is considered to be in the queue for the operator’s attention. Since vehicles tend to produce endogenous events (such as requiring new waypoints when they have reached the old ones), as the number of vehicles or exogenous events in the system increases, the probability of an event experiencing WTQ grows. Additionally, it has been shown that operators may take longer to respond to events as they emerge due to high workload and a loss of situational awareness [4]. Thus, as more events require the operator’s attention, the OCT will continue to grow until it reaches an unacceptable level, at which point a team of multiple operators will likely be required.

3.1.1.2 Switching Strategy

If more than one event is in the operator’s queue, the operator must select which event he will service next. There are several strategies an operator can use, including first-in-first-out (FIFO), highest-priority-first or even random selection. Switching strategy affects the total time tasks spend waiting for service not only because of the ordering of the tasks (queuing policy), but also because of the time required for the mental model change of the operator (switching cost) if the tasks are dissimilar [8]. It has been demonstrated that for operators of multiple, unmanned vehicles, the switching cost can be substantial [9].

3.2 Single Operator Discrete Event Simulation Model

Solving traditional queuing models can yield results of interest to the study of supervisory control such as the average time an event will spend waiting in a queue and server (operator) utilization. Although analytical solutions are possible for simple supervisory control systems, often the assumptions required for closed-form solutions, such as steady-state behavior and independent arrivals, are not met. Discrete event simulations (DES) overcome many of the limitations of analytical models by using computational methods that do not require such strict assumptions [10] and therefore allow a richer set of complex UV-operator systems to be modeled.

A single human operator controlling heterogeneous unmanned vehicles was successfully modeled using a Multi-UV Discrete Event Simulation (MUV-DES) model [8]. A Multi-UV, Multi-Operator Discrete Event Simulation (MUVMO-DES) model that builds upon this work, but also considers multiple operators combining their efforts, is the focus of this research. This new model consists of exogenous and internal inputs, operator servers and their interactions, and a task allocation mechanism that disseminates events to the operators according to the team structure and state of the system. The inputs to the model are both exogenous, such as the arrival rate of new contacts, and also internal, such as the length of time an operator spends interacting with a vehicle. These inputs are also stochastic due to the large amount of uncertainty in environmental conditions and human behavior.

4. METHODS

4.1 Multi-UV, Multi-Operator Discrete Event Simulation Model

Expanding the MUV-DES model to multiple operators required several new considerations, in particular a model of team communication, mutual performance monitoring and task allocation.

4.1.1 Modeling Communication

Geographically-disperse UV operators communicate through voice, chat or a combination of both. Voice communication is typically the fastest and allows operators the ability to control the UVs while simultaneously communicating via a headset. Voice communication is effective for small teams but can become problematic as the number of operators becomes large, due to multiple voice messages that occur simultaneously. Thus, voice communications are typically serial in nature, meaning only one operator can speak at a time. Chat messages allow operators to send messages to each other asynchronously and in parallel. Due to software's ability to parse text and apply sorting filters in real-time, chat communication often scales well with large teams. Chat messages also tend to be clearer than voice communication, in that they are not as susceptible to noisy communication channels, background noise, volume or operator accents. Furthermore, chat messages automatically create a real-time transcript of the communication, something that is typically not possible with voice. For the initial MUVMO-DES model, communications are assumed to be chat for data gathering purposes, but given the widespread use of chat by operational command and control personnel, this assumption also carries external validity. Modeling voice communications is left for future work.

4.1.2 Mutual Performance Monitoring

In addition to explicit communications, operators may also coordinate by mutual performance monitoring, recognized as one of the core components of teamwork [11]. Through a user interface, operators can typically view each other's vehicles and commands to gain situation awareness of what the team is doing. For instance, instead of explicitly communicating, an operator may take a quick look at the interface to see if any other operator's vehicles are already heading to a new contact before assigning their own. However, because this form of coordination is unilateral, teammates must make assumptions about the actions and intentions of other teammates which may or may not be valid.

4.1.3 Modeling Coordination

Communication and mutual performance monitoring can be represented by discrete endogenous events that the operators generate. For instance, in Figure 2, instead of servicing an event once it arrives (event A), an operator may choose to send a chat message to other operators by first starting a chat message, composing it for a finite period of time (labeled COORD) and then sending it before starting to service the task (event C). Similarly, an operator may perform a mutual performance monitoring task that also takes a finite period of time. However, if an operator is composing a chat message or monitoring the performance of other operators, then the operator is considered to be busy and as such, any event that is waiting for the operator's attention while he is communicating or monitoring will incur a WTQ for that period of time. This additional WTQ represents a quantitative measurement of the coordination cost (process loss) associated with the team performance.

The timeline shown in Figure 2 is a simple example of coordination but more complex coordination scenarios exist as well. For simple tasks, a single communication message may be all that is needed, such as claiming responsibility for a target that emerges. For more complex tasks, the communication may involve a conversation that spawns several iterations of communication messages. This initial model will only assume single communication messages and as such, will only be able to model simple coordination between the team members.

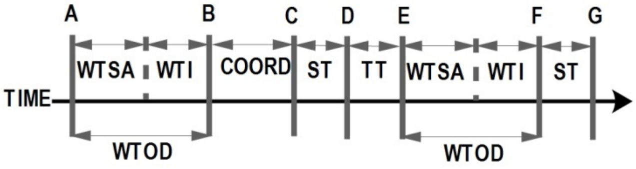


Figure 2: Timeline of events with coordination.

4.1.3.1 Coordination Strategies

Similar to switching strategies, an operator will also have a coordination strategy that dictates the type and timing of the coordination he will perform when faced with a task that can be serviced by more than one operator. One such strategy is to not coordinate at all, but this would require the team to have predefined roles and responsibilities (such as mechanistic teams) or run a high risk of task allocation errors. A task allocation error occurs when more than one operator or no operator attempts to service a particular task.

If an operator choose to coordinate her actions, she typically must choose the type of coordination first, i.e. whether or not to communicate, monitor or both. In addition to the type of coordination, the timing of the coordination is very important as well. A common strategy would be to coordinate first and then service the task. This type of coordination strategy is the least likely to incur task allocation errors. This coordination strategy was assumed for the initial MUVMO-DES model. However, other coordination strategies exist. For instance, an operator could service the task first and then send a courtesy message to other operators. This strategy allows the operator to give the fastest response to an event, but raises the possibility that another operator will also begin servicing the task before the first operator gets a chance to send the coordination message.

4.1.3.2 Team Structure and Task Allocation

Although the model was designed to be general and handle a variety of team structures, *mechanistic* and *organic* teams structures were chosen to be modeled initially since they represent two polar opposites of the organizational spectrum [12]. A mechanistic team is one where the operators have rigidly defined roles and responsibilities. For instance, when all of the vehicles of one type are assigned to one and only one operator, then that operator is given the full responsibility for performing the tasks that only that vehicle can do. If one of each vehicle type is allocated to each operator instead, then that team structure would be considered organic since any operator can perform any task that arises, provided that he has an appropriate vehicle available. Both team structures suffer from inefficiencies, or what Steiner [13] refers to as a "process loss" which is the differential between the performance of a team and the theoretical maximum achieved if the efforts of the individuals were combined ideally. In mechanistic teams, process loss occurs when task loads are uneven and some operators are too busy while others are idle. In organic teams, process loss occur when operators have to spend time coordinating how they will share the common queue and/or allocate the tasks amongst themselves in a sub-optimal manner.

Due to the clear task allocation roles, extending the MUV-DES model for mechanistic teams involved having a separate queue and server for each operator. Since each task was unique to an operator, every event that arose was automatically assigned to the appropriate operator.

For the organic team, a different task allocation mechanism was needed. Since the model is merely an abstraction of the actual scenario, the first attempt at an organic model randomly assigned the tasks to the operators based on who was available at that moment to service the event. If more than one operator was available, the event was randomly assigned to one of the available operators. If no operator was available, the event waited in a common queue (incurring a WTQ cost) until an operator became available. This form of modeling assumes that there will be no task allocation errors, i.e. one and only one operator will service or attempt to service any particular task. In real organic teams, this will likely only happen if the teams coordinate their actions through communication or mutual performance monitoring.

4.2 Data Gathering

The MUVMO-DES model utilizes stochastic processes to account for the uncertainty within the system. Therefore, random values are drawn for WTOD, service time, communication time, monitoring time, travel times and travel time in the model. These probability density functions (pdfs) need to be generated by binning empirical data into histograms and fitting an appropriate curve.

To generate the stochastic inputs necessary for model building and to validate the model's outputs against actual team performance metrics, real data must be gathered. Since there are no extant systems of teams of operators each controlling multiple unmanned vehicles, there is no "real world" data to collect. Hence, an experimental test-bed where teams of operators controlled multiple UVs was specifically developed and experimental trials were conducted to gather the data used for model building and validation.

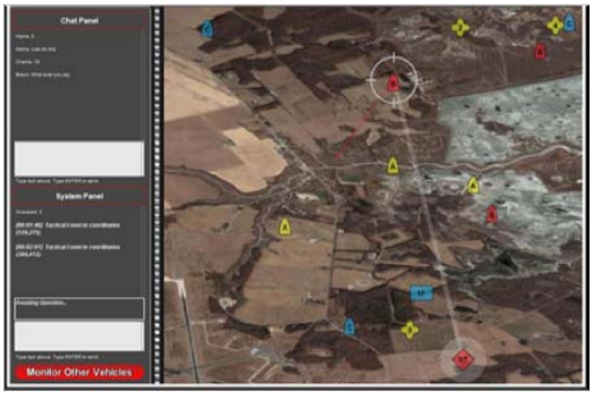


Figure 3: Main display of the ground control interface.

4.2.1 Experimental Test-Bed

The experimental test-bed consisted of a video game-like simulation of unmanned vehicle control by a team of operators. The simulation included three ground control stations, with one subject assigned to each station.

4.2.2 Ground Control Interface

Subjects interacted with the ground control stations via a computer monitor display using standard keyboard and mouse inputs. The main display of the ground control station featured three sections – a large map, a chat panel and a system panel (Figure 3). The map represented the geographical area that the operators were responsible for, as well as all the vehicles under their control and contacts that they needed to handle. Contacts and vehicles were represented using MIL-STD-2525B icons [14] and the operators assigned vehicles to contacts by clicking on the map interface with the mouse. The operators were also able to communicate with each other via instant messaging within the chat interface window. Operators would type messages into the chat, which would then appear on all the other operator's chat panels instantly. Chat messages were labeled with the operators unique IDs, which corresponded to the labels for each operator's vehicle icons. In addition to the map and chat display, there was also a system panel where the system would occasionally send messages to a particular operator, such as a confirmation message that the operator had assigned a particular vehicle to travel to a particular location.

4.2.3 Tasks

Each mission scenario required a team of operators to "handle" contacts that appear intermittently over the map. To do this, the team of operators needed to perform both assignment and payload tasks.

4.2.3.1 Assignment Tasks

Assignment tasks required the operators to send their vehicles to the contacts on the map as they emerged. Once assigned, the UV would start to travel to that particular contact location on the map in a straight line and would continue until either the vehicle reached its assigned destination or the operator re-assigned the vehicle elsewhere. There were no obstacles on any of the maps and no path-planning required.

Although assignments were done by individual operators, they can be considered a "team task" since the operators had to coordinate their assignments to ensure that one and only one vehicle was assigned to each and every contact. Furthermore, subjects were instructed that vehicles should be chosen in the interest of minimizing travel times, i.e. typically the closest available vehicle to the contact location.

4.2.3.2 Payload Task

Once a vehicle reached a contact, the operator performed a simple task by interacting with the vehicle's payload. This task was unique to the vehicle and contact type, but involved either visual identification (e.g., where is the red truck in the parking lot?) or a simple hand-eye coordination task. Since all three vehicles were aerial of some sort, all payload tasks involved a birds-eye view of the terrain. An example of a hand-eye coordination task is shown in Figure 4 where the operator must destroy a contact by centering the crosshairs over a stationary target on the ground and pressing the fire button three times. The difficulty in this task was that the crosshairs are subject to jitter due to the motion of the UV. The other hand-eye coordination task involved dropping aid packages to victims on the ground. This task was similar to the destruction task except that the crosshairs were steady but the projectiles were slow-falling and susceptible to the wind. Thus, players had to compensate for a light north-east wind, for instance, by aiming packages slightly to the southwest of the target location and pressing the drop button once. Payload tasks are considered an "individual task" as they do not require any coordination or assistance from any of the other operators.

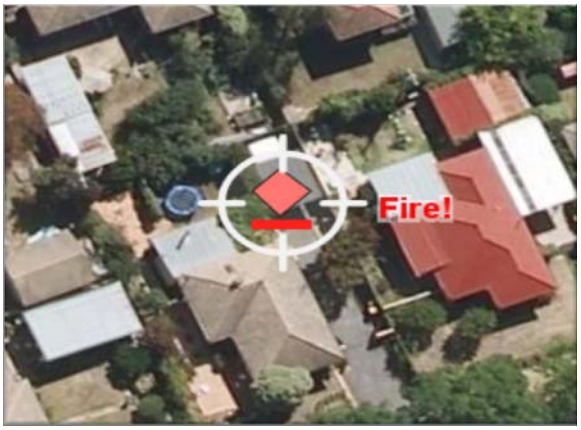


Figure 4: Missile firing payload task.

4.2.3.3 Scenario Objectives

The objective of each scenario was to identify all unidentified contacts and either rescue them (if friendly) or destroy them (if hostile) as quickly as possible. There were three vehicle types, one that handles each type of contact (unidentified, friendly, hostile) exclusively. Although any UV of the appropriate type could be assigned to a contact, only the first vehicle to start the payload task could successfully complete it. When a contact first appeared on the map, it was always of the unidentified type, which required a scouting UV (Type A). Once the scouting UV arrived, the operator performed a visual identification task which transformed the contact from unidentified to either hostile or friendly. If the contact was identified as being hostile, a tactical UV (Type B) was sent by an operator to the contact location to destroy it via the missile firing task. Similarly, if an unidentified contact was identified as being friendly, a rescue UV (Type C) was sent by an operator instead to drop aid packages to the contacts' location, thereby "rescuing" the contact. The time a contact spent in the system, from the moment it arrived, until the moment it was successfully handled, was the objective completion time. Since a scenario consisted of multiple contacts, the Average Objective Completion Time (AOCT) was the metric of interest, where the average was simply the mean of all the OCTs for that scenario.

4.2.3.3.1 Design of Experiments

A 2x2 repeated measures experiment was conducted where the independent variables were team structure (mechanistic, organic) and the inter-arrival time of unidentified contacts (constant, erratic). Ten teams of three participants each completed all four treatments. The order of trials was counter-balanced and randomly assigned to the teams. An alpha value of 0.05 was used for significance.

4.2.4 Independent Variables

4.2.4.1 Inter-Arrival Times of Exogenous Events

Previous research has demonstrated that optimal UV operator performance occurs when the operator has a utilization lower than 70% [15]. Thus, all scenarios were designed to have an operator utilization of about 50%, meaning that operators spent approximately 50% of their time, on average, performing assignment or payload tasks. This was achieved in pilot studies by fixing the payload tasks and manipulating the number of exogenous events and their inter-arrival times until the average operator utilization was about 50%.

The experimental trials had a total of 16 exogenous events (unidentified contacts emerging). The time between successive exogenous events (the inter-arrival time) was 30 seconds for the constant treatment. For the erratic factor level, the inter-arrival times were generated from a bimodal distribution where the means of the modes were set at 75 seconds and 225 seconds from the start of the trial, with a standard deviation of 15 seconds. In both the constant and bimodal treatments, the first exogenous event always appeared at time 0, thus only 15 events were drawn from the bimodal distribution for the erratic condition. The inter-arrival of exogenous events was varied between constant and erratic to determine if team structure had an effect on how operators performed under different task load distributions.

4.2.5 Participants

Participants were recruited via e-mail and paper advertisements and through word-of-mouth. All of the participants were between the ages of 18 and 35, with the mean age being 21.7. Some participants had military, video game or previous UV experiment experience. Due to scheduling concerns, some teams were composed of individuals who knew each other while most teams were composed of individuals who were randomly assigned. The level of inter-personal relationships between team members (stranger, casual acquaintance, friend, romantic, etc) was not recorded.

4.2.5.1 Training

Prior to the experimental trials, the participants completed an individual 20-minute PowerPoint® training session. Afterwards, the participants completed two practice scenarios (one mechanistic and one organic) as teams, each one taking about 10 minutes to complete. Thus, the total training time was approximately 40 minutes.

5. RESULTS

The order of the trials was checked to determine if a learning factor occurred across the four team sessions. Given that the training time was minimal, and previous research has shown that four or more training sessions is needed for teams to achieve stable performance [16], testing order was of concern, and showed a significant effect ($F(3, 24) = 4.12$, $p=.02$). Most teams did worse on the first trial, regardless of the treatment, than on subsequent trials (Figure 5). Thus, the final statistical model included a two factor, repeated measures ANOVA with blocking on the trial order.

Team structure was significant ($F(1, 24) = 1.484$, $p < 0.01$), with mechanistic teams performing better than organic teams overall, although there was no significant difference when the inter-arrival rate was erratic. Mechanistic teams performed worse when the inter-arrival rate was erratic as opposed to constant ($t(15.8) =$

2.47, p = 0.03). However the inter-arrival rate had no significant effect on the organic teams. The inter-arrival rate by itself was not significant, but the interaction of the independent variables was (F(1, 24) = 10.47, p = 0.04).

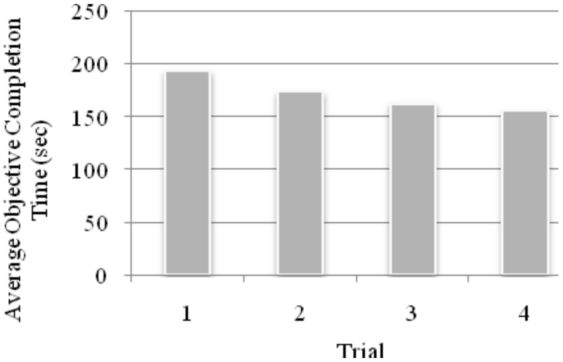


Figure 5: Effect of AOCT vs trial order.

5.1 Model Results

The model was run 1000 times for each treatment condition. For the organic team, the model predictions were within the 95% confidence interval of the empirical results for all four treatments (Figure 6). Since the mechanistic teams did not have to coordinate their actions due to their rigid role structure, they were initially modeled without any communication or monitoring behavior. In the erratic inter-arrival condition, the model predictions for the mechanistic team was within the 95% confidence interval, however for the constant inter-arrival condition, the model's predictions were low (Figure 6).

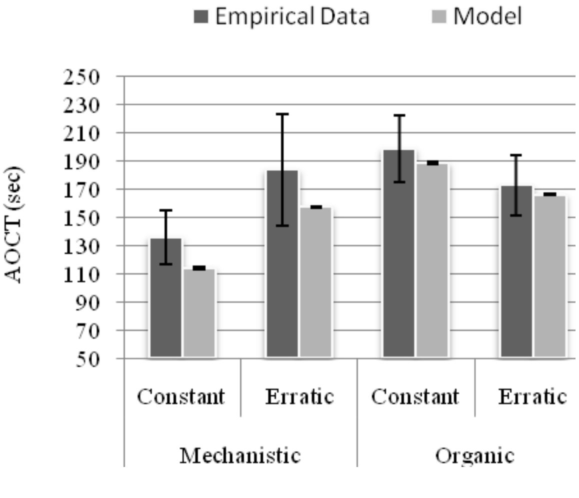


Figure 6: Initial empirical results.

Upon further investigation of the experimental transcripts, the mechanistic team did communicate and monitor each other's actions, even though it was not necessary. Thus, a coordination strategy similar to that used by the organic team was implemented in the mechanistic model and new outputs were generated. Not surprisingly, the additional cost associated with coordination increased the OCT of the mechanistic team. Thus, with the coordination strategy implemented in both teams, the model predictions were within the 95% confidence interval for all four treatment conditions (Figure 7).

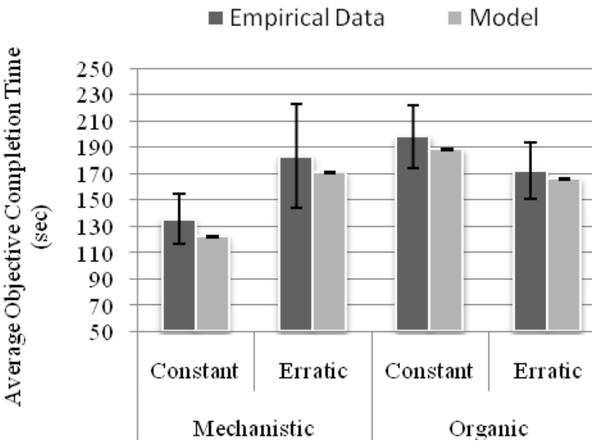


Figure 7: Revised empirical results.

6. DISCUSSION

It was not surprising that the mechanistic teams performed worse under erratic inter-arrival times than they did when the inter-arrival times were constant, since the erratic inter-arrival times caused events to arrive in batches, thereby increasing the queues. However, it was interesting that there was no significant difference in the performance of the organic team under the different inter-arrival rates of exogenous events. This suggests that even though events arrived in clusters during the erratic inter-arrival treatment, the organic team was able to handle the workload spike without increasing the AOCT. This suggests that the organic team is more robust to environmental uncertainty than mechanistic teams due to their flexible structure and the ability to spread tasks across the team.

It was predicted that mechanistic teams would perform better than organic teams, which they did, but not necessarily for the same reasons. Originally, mechanistic teams were thought to have an advantage over organic teams because they did not incur coordination costs. As shown in the results, mechanistic teams do incur coordination costs and without taking these costs into consideration, the performance predictions are too low in the constant inter-arrival case. This is interesting because the communications are theoretically unnecessary. However, this highlights the importance of understanding the intrinsic need for communication between team members, even if it is not necessary. Future work should look at how to mitigate such communication overhead.

So, if mechanistic teams are also incurring coordination costs, how are they still managing to perform better overall than organic teams? The answer to this question perhaps lies in the fact that the empirical data used to generated the pdfs for the different sources (e.g. travel times, WTOD, service times) was separated into the four different treatment conditions. Although there was no statistically significant difference between the values and the differences could be attributed to sampling error, there were small differences in nearly every input condition. Since the OCT is the sum of all of these individual times, then these differences (or errors) combine into a statistically significant result.

Other factors may play a role as well, such as the switching strategy of the operators. The switching strategy assumed for all of the operators was FIFO, although in many cases, operators did not adhere to this strategy. Thus, future analysis should determine the actual switching strategies observed in the experimental trials and implement those instead.

Another issue is that statistical significance for data such as WTOD was difficult to obtain due to a number of factors. First, the sample size of the experiment was small (n = 10) but this is not unusual for team studies since it takes multiple participants to form a single experimental unit. Increasing the sample size should reduce the standard error of the experimental results. Additionally, previous research has shown that UAV teams do not reach asymptotic performance levels until after they have completed around four sessions together [16]. Although this is likely to be highly contingent upon a number of factors such as the difficulty of the task, the inter-operability required for success and the length of the sessions, it does seem to be consistent with our results. Thus, to further reduce variability in the experimental results, additional practice sessions should be added. Finally, the experiment was not controlled for the skill level or the relationships of the individuals. Factors such as age, video game experience and military background could have had an effect on individual performance. If a reduction in the variability of the team's performance is desired, then future experiments could select for and block on particular individual traits. However, teams of futuristic UV operators may be just as diverse as the sample population, particularly if they are composed of individuals from different agencies or even nations operating via an interoperability standards [17]. These operators may have different levels of training, skills and attitudes which may result in significantly different levels of individual performance. Thus, it is not necessarily a flaw in the experimental design to have diversity in regards to the individual traits, as it can be argued that such diversity will be likely in future UV systems.

7. FUTURE WORK

The model in this paper has successfully replicated the results of experimental trials, but it has not been used to predict the performance of teams in hypothetical situations. Future work will look at developing the model to predict the performance of teams in new scenarios and then verify those results empirically. One such scenario could be if the teams had an additional member or decision support tool that aided in task allocation. While the mechanistic teams performed better than organic teams overall, the fact that the mechanistic teams were more sensitive to variations in the environment suggests that this team architecture may not be ideal for volatile environments such as those found in command and control settings. If an organic team had the benefit of a leader or decision support tool, then its coordination costs might drop significantly, whereas a leader or decision support tool would likely have little or no effect on a mechanistic team. Thus, the team model could be updated to see just how much of a performance difference one could expect by having a leader or decision support tool in both team structures.

8. ACKNOWLEDGMENTS

This research is sponsored by the Office for Naval Research and the Air Force Office of Scientific Research. We would also like to acknowledge M.I.T. undergraduates Tony MacDonald, Kyle Miller, Anrae Jamon Motes, Ben Sedat, Jieyun Fu and An V. Ho for their programming contributions to this project and post-doctorate associate Birsen Donmez for her assistance with statistical analysis.

9. REFERENCES

[1] A. Ollero, & Maza, I., *Multiple Heterogeneous Unmanned Aerial Vehicles*, Berlin: Springer, 2007.

[2] J. Franke, V. Zaychik, T. Spura *et al.*, "Inverting the Operator/Vehicle Ratio: Approaches to Next Generation UAV Command and Control."

[3] M. L. Cummings, S. Bruni, S. Mercier *et al.*, "Automation architecture for single operator, multiple UAV command and control," *The International Command and Control Journal*, vol. 1(2), 2007.

[4] M. L. Cummings, and P. J. Mitchell, "Predicting controller capacity in remote supervision of multiple unmanned vehicles," *IEEE Systems, Man, and Cybernetics, Part A Systems and Humans,* vol. 38, no. 2, 2008.

[5] S. R. Dixon, C. D. Wickens, and D. Chang, "Mission control of multiple unmanned aerial vehicles: A workload analysis," *Human factors,* vol. 47, no. 3, pp. 479, 2005.

[6] M. Cummings, C. Nehme, J. Crandall *et al.*, "Predicting operator capacity for supervisory control of multiple UAVs," *Innovations in Intelligent UAVs: Theory and Applications,* vol. 70, 2007.

[7] D. Olsen, and M. Goodrich, "Metrics for evaluating human-robot interactions."

[8] C. Nehme, "Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle Systems," Aeronautics & Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[9] M. Goodrich, M. Quigley, and K. Cosenzo, "Task switching and multi-robot teams."

[10] J. Banks, and J. Carson, "Discrete-event system simulation," Prentice-Hall, 1984.

[11] E. Salas, D. E. Sims, and C. S. Burke, "Is there a "Big Five" in Teamwork?," *Small Group Research,* vol. 36, no. 5, pp. 555-599, October 1, 2005, 2005.

[12] T. Burns, and G. Stalker, *The management of innovation*: Oxford University Press, USA, 1994.

[13] I. D. Steiner, *Group process and productivity*: Academic Press New York, 1972.

[14] "Common Warfighting Symbology," *MIL STD 2525B*, Defense, ed., 2005.

[15] B. Donmez, C. Nehme, M. Cummings *et al.*, "Modeling situational awareness in a multiple unmanned vehicle supervisory control discrete event simulation," *Journal of Cognitive Engineering and Decision Making, Special Issue on Computational Models of Macrocognition (in review)*, 2008.

[16] N. J. Cooke, P. A. Kiekel, and E. E. Helm, "Measuring team knowledge during skill acquisition of a complex task," *International Journal of Cognitive Ergonomics,* vol. 5, no. 3, pp. 297-315, 2001.

[17] M. Cummings, A. Kirschbaum, A. Sulmistras *et al.*, "STANAG 4586 Human Supervisory Control Implications," *Air and Weapon Systems Dept, Dstl Farnborough & the Office of Naval Research*, 2006.

Internal Model Generation for Evolutionary Acceleration of Automated Robotic Assembly Optimization

Jeremy A. Marvel
Case Western Reserve University
Cleveland, OH, USA
jeremy.marvel@case.edu

Wyatt S. Newman
Case Western Reserve University
Cleveland, OH, USA
wyatt.newman@case.edu

ABSTRACT

While machine learning algorithms have been successfully applied to a myriad of task configurations for parameter optimization, without the benefit of a virtual representation to permit offline training, the learning process can be costly in terms of time being spent and components being worn or broken. Parameter spaces for which the model is not known or are too complex to simulate stand to benefit from the generation of model approximations to reduce the evaluation overhead. In this paper, we describe a computational learning approach for dynamically generating internal models for Genetic Algorithms (GA) performance optimization. Through the process of exploring the parameter gene pool, a stochastic search method can effectively build a virtual model of the task space and improve the performance of the learning process. Experiments demonstrate that, in the presence of noise, neural network abstractions of the mappings of sequence parameters to their resulting performances can effectively enhance the performance of stochastic parameter optimization techniques. And results are presented that illustrate the benefits of internal model building as it pertains to simulated experiments of complex problems and to physical trials in robot assembly utilizing an industrial robotic arm to put together an aluminum puzzle.

Keywords: *Genetic algorithms, parameter optimization, model building, robotic assembly*

1. INTRODUCTION

Conceptualizing the transformation from knowing what needs to be accomplished in an optimizeable task to knowing how to actually go about accomplishing said undertaking is an expensive and time-consuming process. These costs are further compounded when the tasks being optimized are susceptible to complex, external influences such as the gross uncertainty of physical systems caused by friction, pressure and temperature. These tasks become problematic because the limitations of virtual models fail to fully capture the complexity of the operational environments and conditions, and thus necessitate the utilization of physical trials for learning.

In previous work at Case Western Reserve University [5] it was demonstrated that Genetic Algorithms can effectively and safely perform rigid-body assembly optimizations by using physical robot systems for parameter evaluation. Using a simple metric of success based on assembly trial speed and contact force, the system was capable of learning how to perform a variety of assemblies quickly and within the bounds of defined safety parameters. Though this implementation was highly successful, it was noted that, by the very nature of the learning method used, the process of optimization was often wasteful. Specifically, parameter sequences that were incapable of even completing the assemblies still had to be tested and allowed to time out before ultimately being discarded.

Numerous attempts to minimize this waste have been attempted, though their methods focus on experimenting with the learning rates [2], population sizes, mutation and crossover rates, child succession rates [4], and competition metrics [9]. In this paper, an augmentation to Genetic Algorithms implementations is described that utilizes the system's experiences in performing an optimizeable function to generate an internal model of the task space.

2. DEVELOPMENT OF INTERNAL MODELS

Genetic Algorithms, stochastic methods of parameter space exploration, follow the biological model of random gene mutations and Darwinian survival to evolve competitive gene vectors of parameters for optimization. While certain implementations may preserve information regarding the evolutionary genetic lineage, the competitive nature of the system does not maintain any history of the gene strains that are deemed unsuitable for survival. As a result, massive amounts of useful knowledge generated by the random search are discarded without actually benefiting the system. Many biological organisms maintain a memory of previous experiences—both positive and negative—and effectively learn from them by altering their future behaviors based on the results from the past. When applied to Genetic Algorithms, these memories could provide a basis for predicting the survivability of the progeny gene sequences, and may actually preempt the necessity of actually running trials doomed to fail.

Within the context of Genetic Algorithms, models are defined as functional mappings from the gene sequence parameters to their respective resulting performances. Explicitly, for the query parameter vector g, executing the gene sequence through the evaluation function f (such as physically performing an assembly task) produces a resulting performance r. By developing an enhanced filter function h to approximate the mapping of the full genetic parameter pool G to its respective output mapping R such that $h(G)\rightarrow R' \approx f(G)\rightarrow R$, where R' is an approximation of R, one should be able to effectively accelerate the convergence on an optimal solution by evaluating only those parameter sequences

PerMIS'09, September 21–23, 2009, Gaithersburg, MD, USA.

that are predicted to surpass the performance of their originating parents.

Assuming that the model is an effective predictive filter and that each trial run by the GA has a constant evaluation cost c, by evaluating only the K projected best-performing parameter sequences of the N total child genes produced by the Genetic Algorithms driver program per generation, one can expect an average convergence performance enhancement cost of cK/N. Of course, in the trivial case where $h = f$ (i.e. there is a perfect model that precisely maps all possible g to their respective r), K effectively becomes 0 since one can effectively eliminate the need to actually run the gene sequences since they can manifest and be evaluated *in silico* because the outcome is already known with certainty.

One of the chief underlying inspirations for this research was the desire to maintain a safe working environment. Manual tuning of assembly parameters is frequently employed, often using the manipulator, itself, as an input device for characterizing parameters and their subsequent performances. While methods such as Design of Experiments [3] have been developed, their implementations require the optimization experts and robot programmers. In an effort to minimize the expertise cost associated with these tuning methods, however, automated tools for the same processes are actively being developed [10].

For robot automation it is preferable to perform as much of the parameter optimization as possible offline due to the inherent risk of damaging the robots and their operational environments given suboptimal or dangerous inputs. For example, a similar approach was used in [6], which utilized existing simulators for reinforcement learning of helicopter flight prior to code deployment on the robot. Customizable simulators such as those used by [8] for clutch assembly modeling, and [7] for mobile robotics benchmarking and analysis, utilize real-world data to construct more realistic representations of the robots and their environments. These solutions, however, are useful only when all of the specified environmental constraints are known. In unknown conditions, [1] developed topological maps for peg-in-hole localization strategies. However, this approach required that the exploration be exhaustive, and that it occur entirely before the localization could begin. What we hope to gain from this work is inline knowledge acquisition and representation for optimization acceleration purposes.

To this end an inline helper function is proposed to selectively prune the child gene pool prior to being executed. This function would take the N children produced by the Genetic Algorithms software and rank-order them according to their predicted performances. The GA implementation would then select the top K projected child gene sequences for trial evaluation, and afterward report to the helper function the results of running those genes such that the helper might then adjust its mapping of the world to further improve the model's predictive abilities.

Because an analytical model is not always readily available, what were tested in this study are two simple numerical approximations of the data parameter-to-performance mapping trends. The first was a standard gradient descent approach to a least-squared linear fitting of the data. With this, we attempted to fit a high-dimensional plane to the surface plot of the observed system outputs for the known parameter sequences. Given an M-dimensional gene sequence, g_i, and a set $S \subseteq G$ of previously-executed gene sequences, selected from S are the M+1 closest (based on Euclidean distance) distinct sequences to g in the parameter space to form the $(M+1)\times M$ matrix X, illustrated as follows:

$$X = \begin{pmatrix} g_{1,1} & \cdots & g_{1,M} \\ \vdots & \ddots & \vdots \\ g_{M+1,1} & \cdots & g_{M+1,M} \end{pmatrix},\ y = \begin{pmatrix} r_1 \\ \vdots \\ r_{M+1} \end{pmatrix}$$

Taken from the set of known outputs $Y \subseteq R$ paired with their respective inputs S are the M+1 resulting performances y (see above). From the M+1 sampled sequences, a linear hyperplane is generated that best approximates the slope of the data trends. This hyperplane is equivalent to the estimated tangent of the model surface, and can thus be described by the normal vector definition given in Equation 1. In particular, the hyperplane solution is identified as the best-guess vector, b, that most accurately explains the observed model such that it minimizes the position and orientation error, ε. The value of b can be computed such that $y = Xb+\varepsilon$ by Equation 2.

$$\left(X^{\mathrm{T}}X\right)b = X^{\mathrm{T}}y \tag{1}$$

$$b = \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}y \tag{2}$$

The second approach employed yet another layer of abstraction by means of a standard feed-forward neural network trained via back propagation in order to generate a best-estimated fit to the explored parameter space. The neural network utilized followed a standard three-layer model (see Figure 1) that consisted of an input layer, I, composed of M nodes, an arbitrary number of "hidden" layer nodes, J, and an output layer, K, consisting of one or more nodes. Here, nodes are defined as equations that take as arguments a single scalar input, u, composed of the summed, weighted outputs from the layer before it, and produce a scalar output, $o = t(u)$, where t is a defined nonlinear "activation function." The links connecting nodes in Figure 1 are the weighted "synapses" that scale the output of the presynaptic node before feeding it into the postsynaptic node.

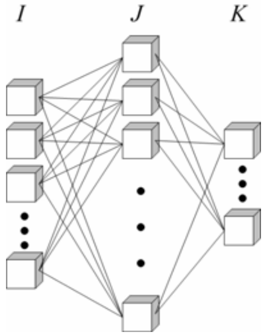


Figure 1: A standard three-layer neural network topology consisting of input-, hidden- and output-layer neurons

The full set S is used to train the network using the associated set of known outputs, Y, to compute the resulting error values, which are then utilized in adjusting the weights of the links connecting adjacent layers of hyperbolic tangent activation function nodes. The sensitivity factor (or how much effect a change in the current value will have in the total network error) for every synapse, w, is computed for each training sequence, p, and the weights are adjusted accordingly. For the synapses linking the hidden and output layer neurons ($w_{k,j}$) the sensitivity factor is computed using Equation 3, while the sensitivity factor for the synapses linking the input layer neurons and hidden layer neurons ($w_{j,i}$) are computed using Equation 4. Training the network adjusts the resulting outputs for a given input parameter sequence, and approximates the surface of the mapping from parameters to solutions.

$$\frac{\partial E}{\partial w_{k,j}} = -\sum_{p} \left(y_{p,k} - o_{p,k}\right) t'\left(u_{p,k}\right) o_{p,j} \tag{3}$$

$$\frac{\partial E}{\partial w_{j,i}} = -\sum_{k}\sum_{p} \left(y_{p,k} - o_{p,k}\right) t'\left(u_{p,k}\right) w_{k,j} t'\left(u_{p,j}\right) o_{p,i} \tag{4}$$

In the above equations, the sensitivity factors are dependent on the errors between the known performance outputs, y_p, and the outputs projected by the neural network, o_p. The function t' is computed as the first-order derivative of the activation function t discussed previously. For experimentation, this function was defined as the hyperbolic tangent, $t(u) = 1.7159\tanh(2u/3)$.

3. RESULTS: SIMULATED HARD PROBLEMS

Given the time constraints of training massive instances of assemblies, and the inherent uncertainties of the existence of a global optimum of physical configurations, initial trials consisted of a reconfigurable simulator of nonlinear problems. Preliminary experiments with mathematical simulators illustrated that simple problems with low-dimensional parameter spaces were too quickly solved by the GA, and that the dichotomy between assisted and unassisted implementations was difficult to distinguish visually. Because of this, it was decided to implement the simulation as a scalable M-dimensional Gaussian that computes a scalar output score, r_j, as defined by Equation 5.

$$r = f\left(g\right) = Ae^{\left(-\frac{(g_1-\mu_1)^2}{2\sigma_1^2} - \frac{(g_2-\mu_2)^2}{2\sigma_2^2} - \ldots - \frac{(g_M-\mu_M)^2}{2\sigma_M^2}\right)} + \beta \tag{5}$$

The value A is an arbitrary scaling constant to determine the maximum value of the Gaussian. For this research, it is assumed to be 1.0 in order to reduce the number of variables. The origin, μ, was centered at $\{\pi, \pi, \pi, \ldots, \pi\}$ and was configured for a variance $\sigma_i = i$ (specifically, $\sigma = \{1, 2, 3, \ldots, M\}$). The individual gene sequence elements g_i were all initialized to 0.5 for all simulator trials, but could be mutated by the GA to be any real value in the range $[-\infty, \infty]$. The addition of a noise parameter, β, allowed for the inclusion of varying levels of static or dynamic noise upon each trial evaluation if such noise is desired.

This model was chosen because it is demonstrably learnable, has a known global optimum, is sufficiently difficult in high dimensions, is repeatable and testable, and is easily augmented given the added noise value β. In the trials run with noise present within the simulator, β was assigned a random value with Gaussian distribution and variance 0.03. This value was dynamically generated upon each query to the simulator, such that for N different queries given a parameter sequence, g, N different random values would be assigned to β. Even in the presence of the random noise, however, the shape of the Gaussian was still clearly visible. Though, given the noisy nature of the problem, the surface became jagged and rife with local optima (as is seen in Figure 2).

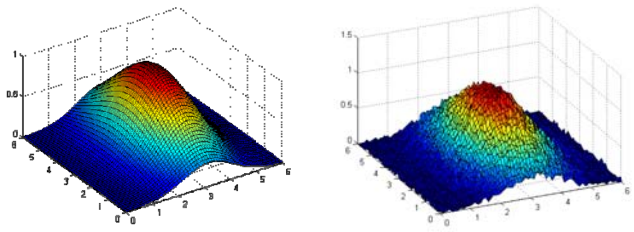


Figure 2: Low-dimension simulator surface plots in both noiseless (left) and noisy (right) operating environments

Because initial tests began with a known application model, there existed the benefit of also being able to demonstrate what advantage having omniscience would grant a Genetic Algorithms implementation for parameter optimization. By using the simulator itself as a predictor of its own outcome, a perfect model was gained which could thus establish a baseline for the best possible expected convergence on a solution with an assisted GA. Naturally, because perfect model of the problem domain existed, the number of performance queries to the simulator essentially dropped to zero due to the reasons discussed in the previous section. However, for the sake of argument, it was assumed the Genetic Algorithms driver program was unaware that the filter function was perfect, and thus it treated the function as it would any other assistant method.

Similarly, because perfect knowledge of the system existed, an analytical approach toward gradient descent for comparative purposes was provided. Using an analytical approximation of the slope of the known model system, performances considerably better than those of an unassisted GA could be expected. The results, however, would not be quite as effective as the perfect model. For the simulator problem, the analytical approximation consisted of an instantaneous tangent plane that passed through the best-performing child of the previous generation's stochastic search. This plane was defined by the solution of the gradient at the origin point $\hat{g}$ computed by the function described in Equation 6 (which can then effectively be reduced to Equation 7). For example, if the M-dimensional gene g_k performed such that $r_k \geq r_j \; \forall j \in N$ in the previous time step, the analytical tangent model would be that which passed through $\hat{g} = g_k$.

$$\frac{\partial}{\partial g_i} f\left(\hat{g}\right) = \frac{\partial}{\partial g_i}\left(e^{\left(-\frac{(g_1-\mu_1)^2}{2\sigma_1^2} - \frac{(g_2-\mu_2)^2}{2\sigma_2^2} - \ldots - \frac{(g_M-\mu_M)^2}{2\sigma_M^2}\right)} + \beta \right) \tag{6}$$

$$\frac{\partial}{\partial g_i} f\left(\hat{g}\right) = \frac{\partial}{\partial g_i}\left(-\frac{\left(g_i-\mu_i\right)^2}{2\sigma_i^2} \right) r = \frac{\mu_i - g_i}{\sigma_i^2} r \tag{7}$$

Associated with each M-element gene vector, g, is an equitable M-element mutation vector, h, that specifies the possible variance of random perturbation around the current set value. Upon each successive generation of training, the mutation variances for generation t+1 are multiplicatively reduced by constant, bimodal "learning rates" according to Equation 8. These learning rates are applied based on either gene succession (i.e. a child gene is selected as the next generation's parent), η^+, or genetic failure (no child performs better than the parent gene), η^-. These learning rates can either broaden the variance to allow for larger search spaces, or can narrow the variance to hone in on some optimum configuration. Gene elements that have an associated mutation variance of 0 are considered "locked," and can not be modified further.

$$h_i(t+1) = h_i(t)\eta \qquad (8)$$

For the simulator trials we set M = 15. The Genetic Algorithms implementation was executed with a single clan for over 30 iterations of learning with 10 children in the clan, for over 300 trial inquiries to the simulator interface. The assisted GA was allowed to generate 1,000 child gene sequences for the single clan, which were then rank-ordered by the filter methods and the 10 children with the best projected scores were selected for trials. The neural network topology consisted of three layers of hyperbolic tangent activation function nodes: an initial layer of 15 input nodes, a hidden layer of 20 nodes, and an output layer consisting of a single node.

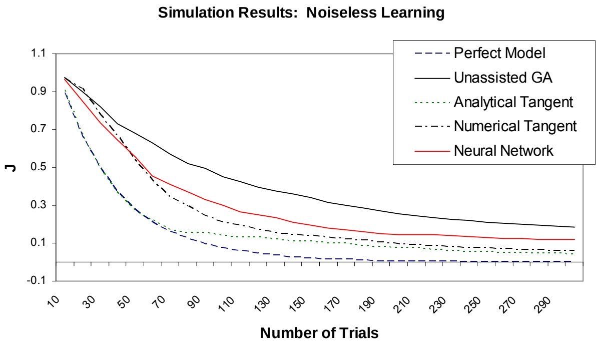


Figure 3: Results of internal modeling enhancement for a high-dimensional simulated Gaussian problem.

The subsequent expected performances of the five optimization techniques (unassisted Genetic Algorithms, perfect model, analytical tangent, numerical tangent, and neural network surface abstraction)—as computed by the averaged results over numerous trials—for the noiseless simulator model are illustrated in Figure 3. Each method is demonstrably monotonically decreasing toward the convergence of some optimal solution. In this instance, the optimum was defined as the apex of the Gaussian curve. Performance improvements are marked by a movement toward convergence (specifically, a smaller value of the error J, which is defined here as the distance from the known maximum value of 1.0) in the fewest number of inquiries to the simulator as possible.

The unassisted Genetic Algorithms implementation faired the worst of all five, while the perfect knowledge model moved toward convergence of the optimal solution the fastest as anticipated. The analytical tangent closely followed the rate of convergence of the perfect model initially, but diverged as it approached the optimal solution and slowed to a rate comparable with that of the GA search. This divergence is likely due to the tangent plane projecting beyond the optimal value, and thus the stochastic search took precedence as the internal model bounced back and forth over the zenith of the Gaussian curve. Both the numerical tangent and the neural network approximations faired considerably better than the unassisted Genetic Algorithms method, but came shy of the performance improvements granted by omniscience.

When noise was added to the simulator, however, the possibility of this omniscience from the GA's perspective was effectively lost. Because of this, the analytical model was thus omitted from this line of testing. The perfect model results from the noiseless experiment, however, are included in the test results as a basis for comparison. Each query to the simulator was performed 10 times, and the resulting noisy outputs of those queries were then averaged to produce an expected result for a given parameter sequence for a single trial. These results were thus used by the Genetic Algorithms program to select which gene sequences would be chosen for parental succession on the subsequent generation of training.

It should be noted, however, that the added noise β distinguishes what the Genetic Algorithms implementation observes and what are the actual outputs for a given set of parameter sequences. The values reported in Figure 4 are based on the performances of the observed best-performing gene sequences per generation as they would actually fare with the noise removed from the equation.

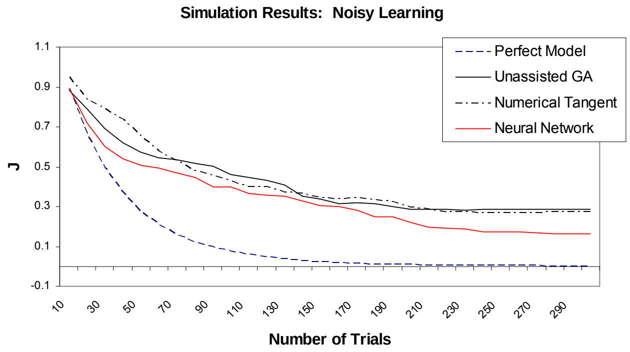


Figure 4: Results of internal modeling enhancement for a high-dimensional simulated Gaussian problem in the presence of noise

Running the simulator with noisy reported outputs demonstrated that the numerical tangent hyperplane performed no better than the unassisted Genetic Algorithms implementation for convergence improvements. One could legitimately argue that the performance of the numerical tangent actually had the potential for being inferior, because one might naturally expect that a stochastic search being run with bad additional information (such as that produced by line-fitting with erroneous data) would likely be worse off than one running with no extra information at all. The neural network approach, however, provided enough data abstraction to perform better than both the unassisted GA and the numerical tangent approximation.

When we increased the range of the noise value β such that the surface of the Gaussian curve became almost indistinguishable (see Figure 5), the dichotomy in performance became even more pronounced. By making the random noise uniformly distributed in the range of [-0.5, 0.5], the surface plot resembles a field of needles more than it does a gently-sloping hill. To the human eye, the curvature of the Gaussian is still barely visible, but to the stochastic search it is little more than a sea of noise. This is made evident by the noticeably worse performance of the unassisted Genetic Algorithms implementation. A similar performance of the gradient descent approach with the numerical tangent approximation is observed, as the resulting fitness was only marginally better, as is illustrated in Figure 6. The neural network model, while being far from converging on the optimum solution of the Gaussian, managed to guide the GA to a solution that was far superior to that which the Genetic Algorithms program would have found on its own. Without some additional insight into the nature of the model, it is possible that the results achieved by the neural network may even be the best possible practicable by a stochastic search method.

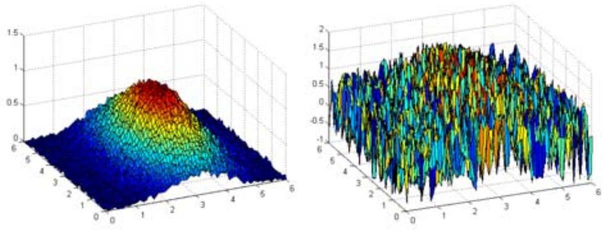


Figure 5: Comparison of the low-dimensional simulator surface plots in the presence of varying degrees of noise

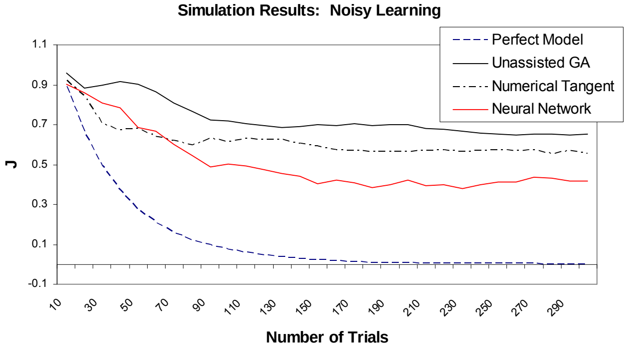


Figure 6: Results of internal modeling enhancement for a high-dimensional simulated Gaussian in the presence of massive noise

From these initial simulated trials, we can infer three things. First, that the presence of an internal model—whether developed analytically or by means of exploration of the search space—has the potential of greatly improving the performance of a stochastic search in its search for an optimal solution. Second, that in the presence of output noise, gradient descent approaches for traversing the parameter space may not perform any better than random walks. More intelligent models than simple hill-climbing are clearly needed. And third, in the early stages of training, the GAs with internal models typically see faster rates of performance improvement than those without internal models.

4. RESULTS: PHYSICAL ASSEMBLIES

While informative, the results of simulations run in Section 3 do not provide sufficiently definitive empirical proof that the internal modeling methods proposed are effective for the acceleration of convergence for stochastic searches. For the virtual problem, there was a known parameter sequence that resulted in a global optimum solution reachable by a simple hill-climbing algorithm. How does the system fare in an environment where the cumulative uncertainties rule out such an ascent, and where there may exist numerous global optima? To this end, a physical assembly trial was configured and implemented to test the proposed internal model generation.

For the assembly configuration, an aluminum pentagonal puzzle (see Figure 7) was set up to be put together using an ABB IRB-140 industrial robotic arm outfitted with an ATI GAMMA force/torque sensor for force feedback in order to facilitate compliant motion control. The puzzle consisted of two stages of assembly: a peg-in-hole search that locked the circular lip of puzzle piece in the inner circumference of the pentagonal hole, and a rotational search that aligned profile of the puzzle piece with the pentagon orientation such that it could be fully inserted. Each search was represented by a parameterized numeric vector of arguments that were generated and mutated by a host computer, and communicated to the ABB IRC5 robot controller using a 4ms fast Ethernet connection for interpretation and execution. Each vector was of fixed length, and the distinct searches are concatenated to form a single input vector to the internal model for training. The GA configuration is identical to the one introduced in previous work [5], with the exception of the addition of the internal model filter method.

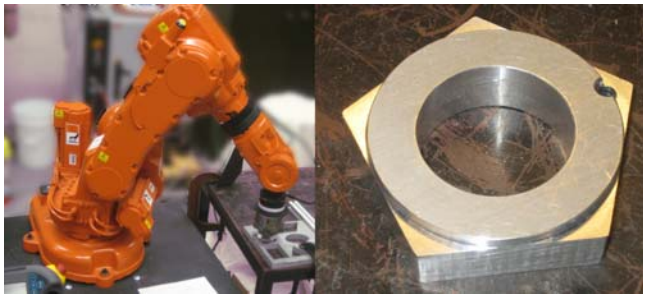


Figure 7: Aluminum pentagonal puzzle insert (right) to be assembled by the ABB IRB-140 open-chain manipulator (left).

The gene sequence fitness score for the physical assembly problem, Equation 9, was a function of the resulting assembly times, T, and incidental forces, F, encountered while performing the assembly task. Here the value of T is equal to the amount of time passed before either the assembly has been completed or the assembly attempt timed out, and F is equal to the average value of the maximum force recorded on the X, Y or Z axes of the torque sensor. Both time and force were bound by pre-defined maximum values ($T_{\max}$ and $F_{\max}$, respectively), and if the task exceeded either value the assembly process would be immediately aborted and the trial given a score of 0. Given that different assemblies may have different requirements regarding time and force, the scaling factor, $0 \le \alpha \le 1$, was used to shift the weight of the score accordingly with regard to where the process importance was focused.

$$r = f(g) = \max\left(\alpha\left(\frac{T_{\max} - T}{T_{\max}}\right) + (1-\alpha)\left(\frac{F_{\max} - F}{F_{\max}}\right),\quad 0\right) \qquad (9)$$

The assembly process for the pentagonal puzzle consisted of three phases of distinct search strategies. Each search strategy was defined by a vector gene consisting of 20 floating point numbers that identified the search strategy, the termination conditions, and the search parameters. For searches that required fewer than 20 numerical values to be fully defined, all unused vector elements were set to 0 in order to maintain a unit gene length.

The first phase was essentially little more than a localization offset that sought to minimize both 1) the search time necessary to engage the circular insert caused by position uncertainty, and 2) the profile orientation caused by rotational uncertainty. In short, the first phase was little more than lateral offsets and a rotation around the tool's Z axis to move the robot to what it believed was an optimal initial configuration for assembly. The second phase performed a spiral search to perform a peg-in-hole assembly of the circular insert. Parameters to be optimized included the spiral radius, search speed, and number of turns per spiral. And the third phase was a rotational search to engage the pentagonal profile of the puzzle piece. Optimizeable parameters for the rotational search included the rotational arc size, search speed, hopping frequency, and hopping amplitude. The hopping parameters controlled an oscillating vertical force profile and were included to minimize the likelihood of the edges of metal components seizing while being assembled. While the puzzle assembly had sub-millimeter tolerances, no great amount of force was necessary to join the insert and the puzzle housing. Because of this, the applied downward force for the assembly task was fixed at 5 N. Initial trials of the unassisted GA with the puzzle assembly demonstrated that the encountered forces never approached the 80 N value of $F_{\max}$. It was thus decided to set value of α in Equation 8 to 1.0, effectively eliminating the force term and restructuring the fitness function to take into account only the assembly time.

As mentioned previously, training for the Genetic Algorithms implementation for the puzzle assembly task was divided into two unique search stages: the spiral search, and the rotational search. The piece components being aligned for assembly are bolded in solid red in Figure 8 for each of the two searches. Stage 1 consisted of learning the optimal position offsets for the first stage of insertion, and then optimizing the spiral search for the circular insert (Figure 8-A). Phase 2 locked the values from the first phase in place, and optimized the orientation offset and rotational search parameters (Figure 8-B). For the assembly trials, Phase 1 training was performed without the assistant filter function and was evolved for twenty-five generations independently before being parametrically fixed. The internal model method was then applied to the second phase of training for performance gauging.

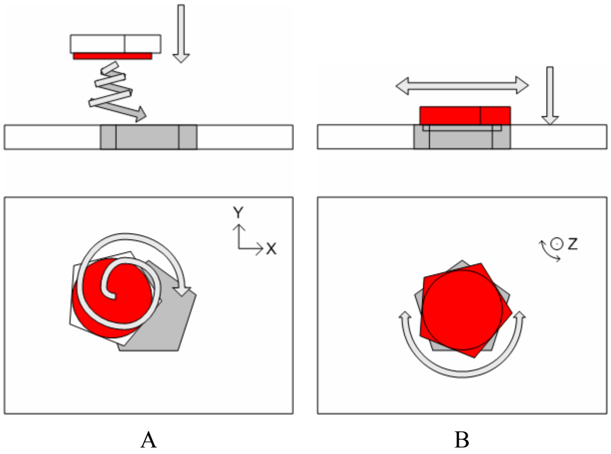


Figure 8: Pentagonal assembly representations of the peg-in-hole lateral search of the circular component insertion (A), and the Z-axis rotational piece profile meshing (B)

Given the results of the simulator with noise and the sub-par performance of the numerical tangent approximation, only the neural network filter method was used in physical testing as an assistive model. The neural network topology from Section 3 was augmented to accept the increased number of search parameters of all three stages, with the number of input-layer neurons being increased to 60. The numbers of hidden-layer and output-layer nodes were maintained at 20 and 1 respectively, however, with the single output representing the time for assembly completion.

The first three generations of parameter optimization were evaluated without the assistive model. The resulting inputs and outputs were used to train the neural network model offline. The assisted and unassisted Genetic Algorithms implementations were then started at the same point given the best-performing parameters after the third generation. The assisted GA model was again allowed to generate 1,000 children, but only the top 10 projected performers were actually evaluated. The results of adding an assistive function to the second sequence are discussed presently.

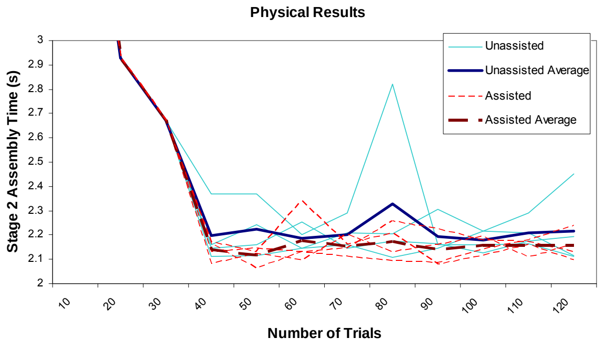


Figure 9: Sample results of internal modeling enhancement for a multidimensional physical assembly.

For each trial, the robot began searching for the assembly attempt at the same location and orientation in space. Plotted in Figure 9 are the results of running the Genetic Algorithms implementations with and without the assistive method. The solid blue lines represent the numerous test results of unassisted stochastic learning, while the dashed red lines show the results of stochastic learning with selective pruning of the genetic parameter pool. With only a few exceptions, the trials had quickly converged to some optimum performance around which subsequent trials oscillated. While the two fared quite well, the assisted model performed, on average, slightly better than the unassisted stochastic search. These results are reminiscent of those comparing the unassisted and assisted GA implementations of the high-dimensional simulated problem in the presence of noise discussed in Section 3.

As an additional test, random noise was introduced to the system in the form of random robot configurations. To simulate positional and rotational uncertainty, random perturbations were added to the robot's initial pose for each trial. Positional noise consisted of uniformly-distributed lateral offsets in the range of [-2.0 mm, 2.0 mm] for both the X and Y axes, and rotational noise took the form of a random rotation in the range of [-5.0°, 5.0°] along the Z axis. The results for these tests are illustrated in Figure 10.

The performance of the physical testing was comparable to the simulator results with massive added noise, which lends credence to the simulator model for performance testing. With assistance, the training performed better than when it was unassisted, with each training sequence performing better than or equal to the average performance of the unassisted model. Based on the simulated results, the expected performance of the unassisted Genetic Algorithms implementation was projected to improve at a slower rate than it would with assistance. Indeed, as was observed the assisted model's rate of improvement increased as the stochastic search explored more of the parameter space.

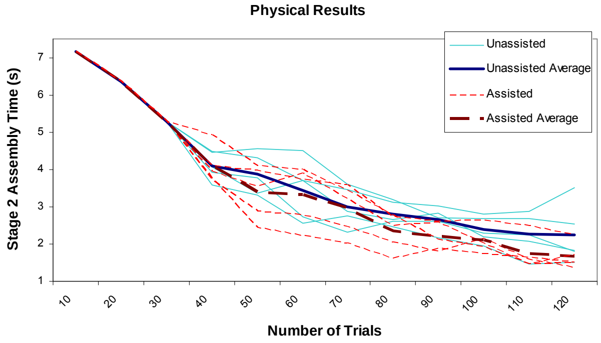


Figure 10: Sample results of internal modeling enhancement for a multidimensional physical assembly.

5. CONCLUSIONS

Discovering an optimal set of parameters for any given task is often a long, slow process that frequently requires repeated testing and adjustment. The research presented in this paper demonstrates that a stochastic approach for parameter optimization can be accelerated by means of internal model building. As the search explores the parameter space, an internal modeling method can extrapolate useful information about its operational environment, and essentially learn from the successes and failures of the system to aid in the guidance of parameter testing and effectively accelerate the rate of optimization.

The results have demonstrated that, even in noisy performance environments, intelligent systems have the ability to successfully extrapolate useful information and effectively improve the rate of improvement. The fact that both virtual and physical applications both benefit from the utilization of internal modeling has considerable implications for the automation of robotic optimization tasks. While these results are still largely preliminary, the potential of internal model generation implementations is clear. Further research into knowledge extraction and intelligent mutation, however, would seem to be warranted. For example, neural networks are only as effective as their input pattern distributions and architectural constructions. Clustering of randomly-generated input parameter sequences may adversely skew the surface model of certain applications away from a global representation and move toward a localized snapshot of the sampled regions with the greatest density, while other application solutions may be independent of such sample densities. In future work, will investigate the nature of automated problems and their parameter spaces that benefit the most from internal modeling.

6. ACKNOWLEDGMENTS

Support for this research was provided by the Robot and Automation division of the ABB Inc. U.S. Corporate Research Center.

7. REFERENCES

[1] Chhatpar, Siddharth R., and Michael S. Branicky. "Localization for Robotic Assemblies with Position Uncertainty." Proceedings of the 2003 IEEE RSJ International Conference on Intelligent Robots and Systems. October, 2003. Pp. 2543-2540.

[2] Gheorghies, O., Luchian, H., and Gheorghies, A. "A Study of Adaptation and Random Search in Genetic Algorithms." IEEE Congress on Evolutionary Computation. 16-21 July, 2006. Pp. 2103-2110.

[3] Gravel, David P. "Efficient Method for Optimization of Force Controlled Robotic Assembly Parameters." Control and Applications. 30 May – 1 June, 2007.

[4] Grefenstette, John J. "Optimization of Control Parameters for Genetic Algorithms." IEEE Transactions on Systems, Man, and Cybernetics. January/February 1986. Vol. SMC-16, No. 1. Pp. 122-128.

[5] Marvel, Jeremy A., *et al.* "Automated Learning for Parameter Optimization of Robotic Assembly Tasks Utilizing Genetic Algorithms." Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics. 21-26 February, 2009. Pp 179-184.

[6] Ng, Andrew Y., H. Jin Kim, Michael I. Jordan, and Shankar Sastry. "Autonomous Helicopter Flight via Reinforcement Learning." Advances in Neural Information Processing Systems. No. 16. 2004. Pp. 799-806.

[7] Taylor, Brian K., Stephen Balakirsky, Elena Messina, and Roger D. Quinn. "Analysis and Benchmarking of a Whegs^{TM} Robot in USARSim." Proceedings of the 2008 International Conference on Intelligent Robots and Systems. 22-26 September, 2008. Pp. 3896-3901.

[8] Yamanobe, Natsuki, *et al.* "Optimization of Damping Control Parameters for Cycle Time Reduction in Clutch Assembly." Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2-6 August, 2005. Pp. 3251-3256.

[9] Yip, Percy P. C., and Yoh-Han Pao. "A Guided Evolutionary Computation Technique as Function Optimizer." IEEE World Congress on Computational Intelligence. 1994. Vol. 2. Pp. 628-633.

[10] Zhang, George Q., et al. "On-Pendant Robotic Assembly Parameter Optimization." IEEE World Congress on Intelligent Control and Automation. 25-27 June, 2008. Pp. 547-552.

Development of Top-Down Analysis of Distributed Assembly Tasks

Anthony Cowley
GRASP Laboratory
University of Pennsylvania
acowley@seas.upenn.edu

M. Ani Hsieh
SAS Laboratory
Drexel University
mhsieh1@drexel.edu

C.J. Taylor GRASP
Laboratory
University of Pennsylvania
cjtaylor@seas.upenn.edu

ABSTRACT

Distributed assembly tasks, in which large numbers of agents collaborate to produce composite objects out of component parts, require careful algorithm design to ensure behavior that scales well with the numbers of agents and parts. Yet algorithm evaluation, through which design is guided, is complicated by the combinatorial nature of system states over the course of execution. This leads to a situation in which the algorithm design space is often severely cramped by the inefficiency of available analysis techniques. We review several available analysis strategies, and present two techniques for designing distributed algorithms that lend themselves to continuous differential analysis while avoiding catastrophic deviation between discrete and continuous system models. This methodology aims to allow optimization at the macro continuous level to inform parameter choice for discrete, real world systems.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

1. INTRODUCTION

In this work, we describe a flexible manufacturing system based on a robot swarm tasked with assembling composite products from distinct parts. The objective is to develop "top-down" design techniques for decentralized control policies that are invariant to changes in team size and part quantities while satisfying workspace and task constraints. To this end, we consider a distributed assembly task where heterogeneous parts are randomly placed within the environment. Assembly is achieved by tasking robots to wander the workspace, picking up parts as they encounter them, and assembling composite objects when they encounter other robots with complementary parts. The dynamics of the assembly task may be modeled as a chemical reaction network since robot-part and robot-robot interactions can be treated as chemical reactions between different molecules.

Our proposed approach is close in spirit to several previous works in which system dynamics are modeled as chemical reaction networks. Hosokawa *et al.* used such a model to predict the yield of full assemblies from a collection of vertically stirred modules [3]. Klavins *et al.* achieved distributed self-assembly from component parts through random collisions of parts that bind and detach from each other based on pre-programmed probabilistic rules [7]. Here, the chemical reaction based model was used to maximize assembly yield by optimizing the spontaneous detachment probabilities of the various components at equilibrium. However, the proposed optimization strategy required the enumeration of all reachable system configurations, which does not scale well with the number of parts. Similarly, Matthey *et al.* developed stochastic control policies from chemical reaction-based models that enabled a robot swarm to assemble distinct products from a collection of heterogeneous parts [10]. The control policies obtained here provided theoretical guarantees on overall system performance. The use of mobile robots to manipulate and assemble passive parts decentrally is similar to other work [11] where the objective was to derive a rule set to enable the construction of an entire structure out of simple building blocks.

Similar to earlier works by Hsieh *et al.* [4] and Matthey *et al.* [10], we propose to develop a "top-down" design methodology for generating stochastic agent-level control policies for a robot swarm based on the mathematical framework used to model chemical reaction networks. Other works have analyzed collective behavior in cooperative robotic tasks [6]. Macroscopic swarm models have been derived to study the performance of a distributed foraging strategy under varying conditions [9], while a similar approach has been used to analyze and study the effects of specialization within large robot teams [5]. In all these works, robots are treated as single molecules and assumed to be capable of simple atomic behaviors, with local interactions between robots governed by a set of *reaction rates*. Since individual robotic agents can only assume a finite set of basic behaviors, it is possible to model system dynamics solely by considering the population distribution across the set of behaviors. By describing the swarm dynamics via a macroscopic analytical model, these works have shown that it is possible to derive stochastic agent-level control policies to meet a particular desired group-level outcome [1, 4, 10], thus providing a "top-down" versus the traditional "bottom-up" approach to designing group behavior.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA
Copyright ©2009 ACM 978-1-60558-747-9/09/09 ...$10.00.

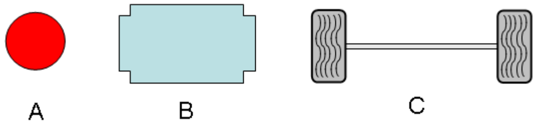


Figure 1: An example of heterogeneous primitive parts `A`, `B`, and `C` that can be assembled into a micro-robot.

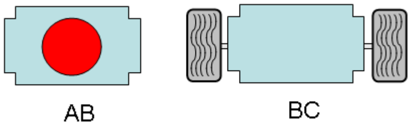


Figure 2: An example of allowable sub-assemblies obtained from the assembly of primitive parts `A`, `B`, and `C`.

We investigate methods to simultaneously adapt the development of these macroscopic analytical models alongside the discrete behavioral algorithms they are to be applied to in order to maximize the fidelity of the models. Improved compatibility between implementation and analysis creates a virtuous cycle where carefully designed algorithms lead to higher fidelity modeling which leads to improved algorithm refinement strategies. Specifically, we consider the execution of collaborative tasks by a swarm of robots whose goal is to assemble composite widgets made of several smaller parts. This is relevant to applications in areas such as flexible manufacturing where it may be desirable to have a system capable of assembling significantly different products on-demand. Other applications include automation of recycling plants and nanoscale assembly where stochasticity is often the norm rather than the exception.

2. PROBLEM FORMULATION

Consider the problem of deploying a swarm of N robots to assemble complex products from a set of heterogeneous parts. For example, consider the problem of assembling a micro-robot from a pair of wheels, chassis, and a sensor. As such, the set of possible part types is given {`A`, `B`, `C`} where `A` corresponds to the sensor, `B` corresponds to the chassis, and `C` corresponds to the pair of wheels as shown in Figure 1. The assembly of the micro-robot can be broken down into the assembly of intermediate products: either an `AB`, the attachment of the sensor to the chassis, or a `BC`, the attachment of the chassis to the wheels, sub-assembly as shown in Figure 2. The sub-assemblies may then be mated with the missing primitive part, either the set of wheels or the sensor, to complete the assembly of the micro-robot as shown in Figure 3. Rather than focus on the details of assembling micro-robots, this work will consider the analogue problem of assembling `ABC` widgets since it provides a nice abstraction for more general assembly tasks.

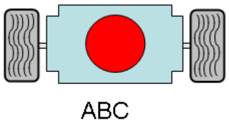


Figure 3: An example of an assembled micro-robot composed of primitive parts `A`, `B`, and `C`.

In particular, we assume uniform distributions of each part type, `A`, `B`, and `C`, within the workspace. Robots navigate the environment by following a trajectory chosen randomly at start-up and upon encountering an environment boundary. As robots wander the workspace, they are tasked to pick-up and assemble intermediate parts, *i.e.* `AB`'s and `BC`'s, and/or `ABC` widgets as they encounter parts and other robots. For simplicity, we assume that the primitive parts, `A`, `B`, and `C` are replaced in the environment as soon as they are picked up for any reason. Furthermore, the intermediate objects `AB` and `BC` are dropped in the environment upon production, while a successful assembly of an `ABC` widget is immediately removed from the workspace, returning each agent involved in its construction to a free state identical to that in which it started. Finally, it is important to note that agents performing these assembly operations have no *a priori* knowledge of their workspace: neither its geometry, the availability of parts, nor the disposition of other agents.

This abstract assembly task requires cooperation between at least two agents, without any high-level coordination. However, while agent-level behaviors that result in cooperative widget assembly are easy to express, and immediately suggest the opportunity for great parallelism through a simple scaling of the number of parts and agents in the environment, system-level performance is contingent on benign interactions between concurrent assembly operations. Toward this goal, we propose to develop robust concurrent assembly strategies that lend themselves to rigorous analysis for the purposes of tuning high-level algorithm parameters. These parameters may include such features as agent-level preference for certain parts in particular situations. Any such biases can have a dramatic effect on system performance, and thus represent important tuning parameters for the system as a whole. Yet, while the effect of such biases at the agent level may be clear from inspection, the effect of their interactions when embodied by hundreds of concurrently operating agents is less clear. In this way, the system tuning process relies on agent-level tuning, and may only be directed by considering multitudes of interacting agents.

To achieve this, we will first develop a baseline approach using a swarm of N non-communicating robots with limited sensing capabilities. In this baseline case, a free robot discovers and picks up a part by physically bumping into it. Robots encounter each other by a similar physical interaction, at which time they will produce a new composite part if such an assembly is possible given the parts held by each agent.

The second variation is similar to the first, but involves equipping each agent with a sensor that allows for the detection of parts, be they of type `A`, `B`, `C`, `AB`, or `BC`, within some fixed *sensing radius* of the agent. Additionally, each agent is capable of coordinating with any other agent within its *communication radius* in order to perform an assembly operation.

The third variation considered here is one in which agents do not speculatively pick up parts at all. In terms of the identification scheme of discrete agent states implied by the previous variations, *e.g.* an agent holding some part of type A, or some part of type C, this variation is as if agents are allowed to exist in several overlapping states simultaneously. An agent may be aware of multiple parts in the environment, *e.g.* an agent aware of both a part of type A and a part of type C, but does not commit to any subsequent operation until said operation is known to be terminating. That is, when the agent comes into contact with another agent such that the two may combine the parts they are aware of to produce a composite object. Put another way, the previous two problem formulations force an agent to commit to a course of action when it encounters a part: should an agent, upon discovering a part of type A, pick up said part, it has preordained its immediate future to consist of an assembly operation in which it contributes a part of type A.

3. ANALYSIS TECHNIQUES

Given an assembly algorithm defined over a set of parameters, a frequent objective is to determine the optimal set or subset of parameters that can satisfy specific performance metrics, *i.e.* maximize widget production. Perhaps the most intuitive approach is to search for these optimal parameters by simulating the assembly process. This is most commonly achieved via an agent-based simulation (ABS) where each robot agent is simulated individually, and time is a synchronous signal used to advance each agent's state of execution simultaneously. While this approach has the advantage of faithfully modeling both the finite individuality of swarm members and the constant advance of time, it ignores many opportunities for improved efficiency. First, if many agents are executing the same behavior, it is not always clear how much is gained by simulating N copies of the agents. Second, the regular sampling of time must be fine enough such that each robot is only expected to be involved in *one* interaction between simulation samples. Otherwise, the order of events during a time interval is unspecified, which can lead to undefined behavior. However, this fine-grained, regular sampling typically results in many samples when nothing interesting happens. Such intervals are identified by purely deterministic behavior that could be perfectly modeled in a more computationally efficient manner. For instance, the position of a particle moving with constant velocity under the influence of no external forces may be accurately predicted by simply integrating the known velocity, rather than simulating the movement by a sequence of identical discrete jumps through space.

To address the efficiency of time sampling while still explicitly representing the discrete agents that make up the system, one may employ a macro-discrete model [2]. In this model, one arranges to only sample the simulation when an interaction occurs. This is achieved by modeling the rate at which events happen with a stochastic process, typically a Poisson distribution, and advancing the simulation directly between the times at which events occur. A Poisson process with time constant k fires at random times with the firing probability per unit time given by k. The process is Markov since the firing probability is independent of past history. The distribution of intervals between two firings can be derived analytically and is given by $p(t) = ke^{-kt}$. Thus, one can simulate Poisson transitions in two mathematically equivalent ways. (1) Run iterations with a small time step $\Delta t << 1/k$; at each iteration, the probability of transition is $\Delta p = k\Delta t$. The transition is triggered in the current iteration if $r < \Delta p$, where $0 < r < 1$ is a uniformly distributed random number. This implementation is exact in the limit $k\Delta t \to 0$. (2) Generate a random number t_r distributed according to $p(t) = e^{-kt}$ and take the transition at time t_r. This second implementation has been shown to be mathematically equivalent to an agent-based simulation [2]. Since each agent is modeled individually, system dynamics dependent on small numbers of individuals may be faithfully captured, while overall simulation performance is greatly increased.

The final strategy considered here is a continuous model of system dynamics. For large enough numbers of robots and parts, it is possible to derive an analytical macroscopic description of the dynamics of the assembly process. Such a model stands in stark contrast to the previous two as it adopts a continuous model of the passage of time, but also takes the drastic step of abstracting discrete parameters into continuously varying values, such as mapping the number of agents engaged in a particular behavior to fractions of the total population. The distinct advantage of this approach is that it brings to bear all the long established analysis tools and techniques from the study of differential equations, and allows for the immediate numerical solution to parameter optimization for most systems. The performance of such methods vastly outpaces the alternate simulation techniques, and thus allows for significantly more rapid iteration of algorithm design since algorithm performance can be easily approximated.

4. ALGORITHM DESIGN

While continuous models are incapable of reflecting the discrete dynamics of a distributed assembly task, they are often good enough, and represent such a compelling performance advantage over other simulation techniques that investigating their applicability is a worthy endeavour. While continuous models treat inherently discrete quantities as continuously varying, their behavior may closely mimic that of the discrete system when large numbers of particles are considered. For example, a continuous model may indicate that, at some point in time, 50.02% of agents are engaged in a particular behavior. If there are one thousand agents in the actual system, then such a configuration is impossible: the assignment of agents to behaviors is entirely discrete. However, if the true performance of the system would have exactly five hundred agents engaged in the behavior, then the deviation of the continuous model from the discrete system may not introduce significant error.

For the specific application of assembly tasks, one may identify the key areas where continuous models break down. One such area is the possibility of deadlock. A deadlocked configuration is one in which no agent is able to change its own state with respect to an assembly task. That is, agents may move about the environment, but none makes any productive contribution to an assembly operation. Such a configuration may result in the example problem described above if every agent should pick up a part of type A, for example, and not release it until it is able to contribute that part to an assembly operation. In this case, no agent is able to find another agent with a complementary part, thus no progress is made in the assembly task. In comparison, a continuous model may suggest that some fraction, say 10%, of a single agent is still free to take productive action, thus allowing some non-zero probability that the system should recover. Yet the discrete, real system does not allow this configuration: if zero agents are able to make progress, then the system is in a stable state of non-production.

Crucially, such a configuration never occurs in a continuous model of the system. This is because each agent exists in all possible states at once according to some probability distribution; no commitment need be made. This deviation of the continuous model from the real system is catastrophic when it occurs, and arguably invalidates any proposed utility of the entire approach. However, this deviation of the continuous model from real system performance may be precluded by designing the algorithm such that deadlock is not a reachable configuration. An example of such a design technique is a transactional approach to resource-consuming operations.

One may view a productive assembly as a sequence of robot resource acquisitions followed by an assembly operation. Note that the action of a robot picking up a part consumes a robot resource. While it is unknown precisely what assembly operation will ultimately exploit that resource, the robot may retro-actively be classified as locked by the abstract assembly operation that eventually uses it. Viewed against time, the interval during which a robot resource is held begins when a free robot encounters a part, and ends when that robot contributes the part to an assembly operation. But this time is unbounded! If instead one is able to bound the time any resource is exclusively held by some operation, then deadlock is avoided. This may be achieved in distributed assembly tasks by taking advantage of the fact that robots are not molecules, and are typically imbued with remote sensing and communication capabilities. Thus, a robot may simply transition between various states of awareness of components in its vicinity without physically acquiring *exclusive* domain over any one part. Note that the exclusive domain in this case is mutual: the part may be viewed as having exclusive ownership of the robot that has picked it up. The robot resource is consumed by the part until it is able to integrate the part into a composite object.

When a communicating group of robots decides that, collectively, they know how to obtain the components necessary to assemble a composite part, then, and only then, do they move to physically pick up the necessary pieces before engaging in a cooperative assembly operation. Part acquisition may fail, but such an eventuality may now be reasonably detected by a simple time limit on the action of a robot picking up a sensed part. That is, one may assume that a robot actively sensing a part may acquire, or fail to acquire, that part in bounded time. In this way, the entire assembly operation occurs in bounded time, and may be viewed as transactional – the full assembly occurs instantaneously or not at all – by an external observer.

A shortcoming of this approach is that it may be too conservative. In fact, some speculative execution of the assembly task may suffice to overcome an environmental condition such as an excessive sparsity of parts. Specifically, if parts are spaced farther apart than twice the robots' communication radii, then no two robots will ever be able to share their concurrent knowledge of part locations. This represents a type of live lock, wherein individual robot state, *vis-à-vis* the set of parts sensed by the robot, changes as time progresses, but no productive work is done. However, if a robot should optimistically acquire a part, thus locking itself to an assembly operation, it may move within communication range of another robot either sensing or holding a complementary part.

Algorithm 1 A simple probabilistic behavior causing robots to switch between driving to the left or to the right for τ time.

```
while true do
  if random() < k then
    driveLeft($\tau$)
  else
    driveRight($\tau$)
  end if
end while
```

In order to avoid deadlock, while maintaining the benefits of optimistic execution, one may specify that some robots act optimistically while other act pessimistically. This amounts to a hedging strategy against unforeseen environmental conditions: the pessimistic strategy is advisable when parts are densely packed as it maximizes potential parallelism, while the optimistic strategy is necessary to make any progress when parts are few and far between. Such a heterogeneous behavior population may be arrived at by programming agents to probabilistically assign themselves one behavior or the other. This stochasticity may be added without affecting the execution of either behavior by wrapping the two deterministic behaviors in a probabilistic conditional expression.

5. EXAMPLE ANALYSIS

One may consider a single probabilistic deterministic algorithm that induces a specific population distribution over discrete classes. This is useful because the single algorithm can be duplicated an arbitrary number of times while always maintaining the desired population statistics. However, for analysis, one can strip off a layer of randomization by representing the algorithm as two distinct sub-populations, each of whose relative prevalence is defined by the statistics of the probabilistic element of the original algorithm. This may be demonstrated by a simple example.

Consider Algorithm 1, which, when executed by a population of N agents equipped with a suitable random number generator approximating a uniform distribution, can be expected to yield kN agents driving to the left, and $(1-k)N$ agents driving to the right. This same algorithm can be deconstructed by lifting the impact of the **if**...**then**...**else** construct into the top-down population specification. Alternatively, this algorithm may be viewed as an implementation of a top-down design directive. Concretely, the chemical kinetics specification,

$$R_{right} \xrightarrow{\frac{k}{\tau}} R_{left}$$
$$R_{left} \xrightarrow{\frac{1-k}{\tau}} R_{right}$$

may be used to model Algorithm 1, or, from the other direction, the above reaction equations may be implemented at the agent level by Algorithm 1. Here, τ can be interpreted as a simple scale factor in the time domain. These reaction equations result in a differential model of the population distribution,

$$\dot{R}_{left} = \frac{k}{\tau} R_{right} - \frac{1-k}{\tau} R_{left},$$
$$\dot{R}_{right} = \frac{1-k}{\tau} R_{left} - \frac{k}{\tau} R_{right}.$$

In order to evaluate steady state production levels of an assembly algorithm, one would typically be interested in equilibrium conditions of this system,

$$\frac{1}{\tau} \begin{bmatrix} k & 1-k \\ 1-k & k \end{bmatrix} \begin{bmatrix} R_{right} \\ R_{left} \end{bmatrix} = 0$$

The equilibrium condition of the linear system induced by Algorithm 1 indicates the relative proportion of agents driving left, and those that are driving to the right. Given a value for k, R_{right} and R_{left} can be solved for by imposing a *conservation* condition that $R_{right} + R_{left} = 1$. That is, the sum of the population fractions must be one. Thus one arrives at the expected result that $R_{right} = R_{left} = 0.5$ if $k = 0.5$, for example. Assembly tasks do not often lend themselves directly to description by linear system, however the resulting system of differential equations may still be solved numerically when no analytic solution is available.

5.1 Results

While the continuous model of the example widget assembly task is free to consider nonsensical concepts such as a fraction of a robot, its performance still matches the agent-based simulation at large population sizes, Figure 4. These graphs show that solutions of the differential equations modeling system dynamics, number of robots holding a part of a particular type and total ABC widget production rate, closely match the behavior of the discrete agent based simulation.

The assumption of the existence of partial robots does, however, deviate strongly from reality in the presence of deadlocked behaviors. This is best seen by considering smaller populations executing locking behaviors where deadlocked configurations are statistically probable events. Figure 5 shows the agent-based simulation's production rate of ABC widgets dropping to zero due to the population working itself into a deadlocked configuration. In this scenario, the most common cause for deadlock proved to be too many agents holding onto intermediate parts of type AB or BC while the remaining agents held parts of type B. If all the agents in the system assume one of those roles, then overall production ceases. The cessation of production is stable and unrecoverable: the controller makes no allowance for detecting and escaping a system-wide deadlocked behavior.

This type of deadlock, where all robot resources are consumed, may be avoided by a transactional assembly approach that does not speculatively lock a robot to a particular part. The steady state production rate for the transactional assembly technique, as seen in Figure 6, dwarfs the steady state of the locking approach shown in Figure 4, even when just considering the continuous model. This is due to the greater extent to which the transactional assembly behavior exploits potential concurrency in the system: the duration for which a robot resource is exclusively locked is strictly bounded. Turning to the agent-based simulation results shown in Figures 4 and 6, the danger of deadlock in the speculative locking strategy is starkly apparent. While the continuous model is not susceptible to falling into a stable "stuck" configuration, the discrete system is, and may do so at any time.

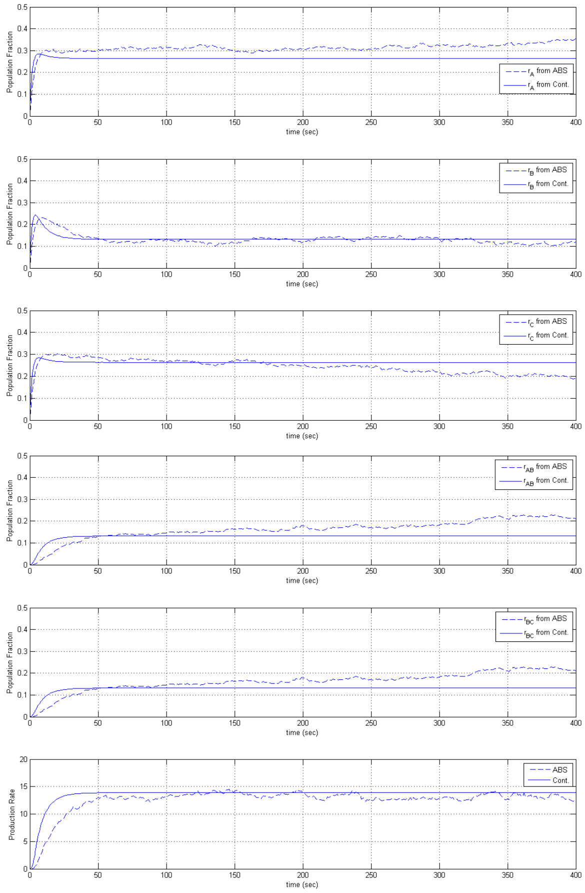


Figure 4: Comparison of system states over time between an agent-based simulation (ABS) and a continuous differential model (Cont) for a locking assembly strategy. Agent population size of 1000; 900 of each type of primitive part.

However, the conservative approach taken by the transactional assembly behavior has its own drawbacks. While it avoids prematurely locking a robot resource, it is far more susceptible to production shortfalls due to unfortunate environmental conditions. Namely, if part density is low, there is a chance that individual parts may be so widely spaced that two robots can *never* simultaneously sense two different, compatible parts and be in communication range of each other. This occurrence is, once again, not seen in the continuous model since, in such a model, a low part density means that a productive configuration, in which two robots sense two parts and can communicate, is unlikely, but never impossible. In a discrete system, however, the transactional assembly behavior may yield a production rate of zero if the parts are distributed with too much distance between them. This second form of deadlock, in which no positive progress may be made by the system, can be avoided by a mixed population.

Without complicating agent-level behaviors, one may avoid both forms of deadlock by hedging against either a sparsity of robots or a sparsity of parts. The performance advantage

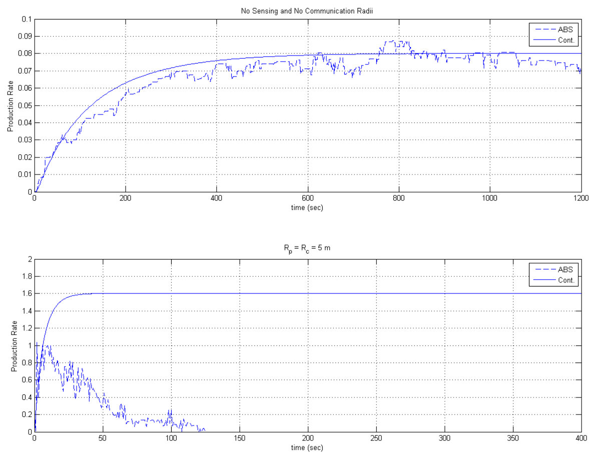


Figure 5: Composite widget productions rates for a team whose agents have effectively zero sensing and communication radii and a team whose agents have finite, non-zero sensing and communication radii, R_p and R_c respectively. Results for the agent-based simulation (ABS) are compared with those from the continuous differential model (Cont). Sensing and communication increase theoretical performance, but quickly deadlock in the agent-based simulation. Agent population size of 100; 900 of each type of primitive part.

of a mixed population is shown in Figure 7, in which 100 robots are operating in an environment with an uneven part distribution. Namely, there are regions of the environment densely packed with parts, and there are regions with very low part density. The production rate of the transactional assembly behavior, shown with a solid red line, plummets when a significant fraction of agents find themselves in low density regions, while the heterogeneous team has a much higher minimum production rate.

6. DISCUSSION

One can see, by comparing Figure 6 with Figure 5, the difference in absolute widget production rates for a given set of system parameters. Even if one ignores absolute productivity measures, the deadlock-freedom of the transactional assembly behavior trumps any other performance considerations of different agent-level behaviors.

Modifications to agent-level behavior that avoid deadlocked configurations have typically been implemented as spontaneous decay reactions in other robot swarm simulations based on chemical reaction networks [8, 7, 5, 10]. However, such spontaneous reactions, in which a robot has some non-zero probability of simply dropping a part it is carrying, may not be necessary in order to avoid deadlock configurations. Instead, deadlock freedom may be attained by more deterministic behavior specification, as in the transactional assembly scheme that avoids robot starvation.

By working with agent-level behaviors that preclude deadlocked scenarios, one is able to leverage computationally efficient differential models of system dynamics. This ability means that one can quickly predict system performance for a given set of parameters, *e.g.* part density, and take some action to adjust these parameters if system performance is insufficient. Hedging strategies, as demonstrated in the mixed transactional-speculative population above, may be efficiently implemented by a probabilistic role assignment mechanism in which agent's adopt a particular behavior based on a desired distribution. This assignment technique is robust to changes in population size, and requires no per-agent customization, thus making it suitable for swarm deployment scenarios. The probabilistic combination of deterministic behaviors represents a sweet spot of easily understood agent-level behaviors, scalability to large, varying

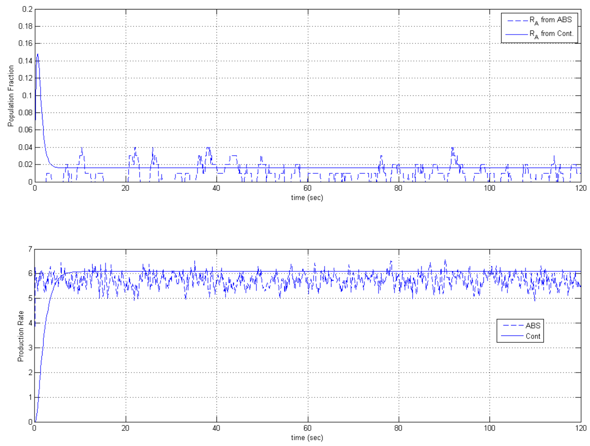


Figure 6: Composite ABC widget production rates for the transactional assembly behaviors. The fraction of agents aware only of a part of type A is also shown for reference. Agent population size of 100; 900 of each type of primitive part.

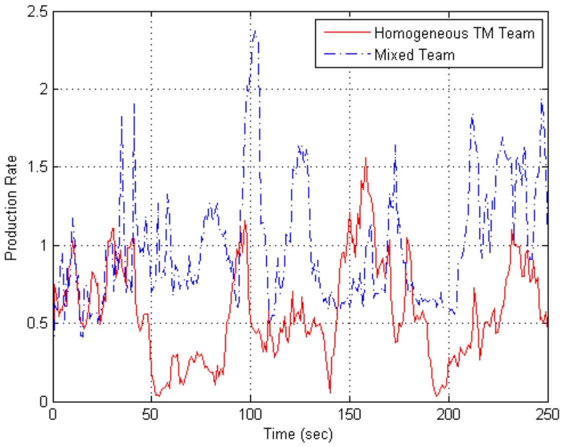


Figure 7: A purely conservative approach can be stymied by a sparsity of resources. The homogeneous team is adversely affected by regions of the environment with a low part-density. To counter this, an even mix of eager and conservative agents (100 of each) avoids the production fall-offs of each in an agent-based simulation.

population sizes, and amenability to differential modeling techniques.

7. REFERENCES

[1] S. Berman, A. Halasz, V. Kumar, and S. Pratt. Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. In *Proceedings of the 2007 International Conference on Robotics and Automation (ICRA07)*, pages 2318–2323, Rome, Italy, April 2007.

[2] D. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58:35–55, 2007.

[3] K. Hosokawa, I. Shimoyama, and H. Miura. Dynamics of self assembling systems: Analogy with chemical kinetics. *Artificial Life*, 1(4):413–427, 1994.

[4] M. A. Hsieh, A. Halasz, S. Berman, and V. Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2):121–141, 2008.

[5] M. A. Hsieh, A. Halasz, E. D. Cubuk, S. Schoenholz, and A. Martinoli. Specialization as an optimal strategy under varying external conditions. In *Accepted the 2007 International Conference on Robotics and Automation (ICRA07)*, Kobe-Japan, May 2009.

[6] E. Klavins. Programmable self-assembly. *Control Systems Magazine*, 24(4):43–56, 2007.

[7] E. Klavins, S. Burden, and N. Napp. Optimal rules for programmed stochastic self-assembly. In *Proc. Robotics: Science and Systems II*, pages 9–16, Atlanta, GA, 2007.

[8] K. Lerman, A. Galstyan, A. Martinoli, and A. J. Ijspeert. A Macroscopic Analytical Model of Collaboration in Distributed Robotic Systems. *Artificial Life*, 7(4):375–393, 2001.

[9] K. Lerman, C. V. Jones, A. Galstyan, and M. J. Mataric. Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, 25(4):225–242, 2006.

[10] L. Matthey, S. Berman, , and V. Kumar. Stochastic strategies for a swarm robotic assembly system. In *2009 IEEE International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009.

[11] J. Werfel and R. Nagpal. Three-dimensional construction with mobile robots and modular blocks. *International Journal on Robotics Research*, 27(3-4):463–479, 2008.

Context-Based Object Recognition

Shaun Edwards
Southwest Research Institute
6220 Culebra
San Antonio, TX
1-210-522-3277
shaun.edwards@swri.org

Meredith Wright
Southwest Research Institute
6220 Culebra
San Antonio, TX
1-210-522-6247
meredith.wright@swri.org

Ben A. Abbott, Ph.D.
Southwest Research Institute
6220 Culebra
San Antonio, TX
1-210-522- 2802
ben.abbott@swri.org

ABSTRACT

Several methods have been developed for context-based object recognition within aerial imagery. These methods were inspired by human object recognition, which has been shown to rely on contextual information as opposed to classical appearance based methods. While this concept may not be new, this research sought to develop generic methods that leveraged recent developments in cognitive systems research, and more specifically large scale ontologies or knowledge bases. The results of the research have shown that context-based methods, supported by an ontology, can increase recognition rates versus classical appearance based methods. These methods have the potential to automate many complex object recognition tasks, aerial imagery analysis being one of them, that currently require human analysis.

Categories and Subject Descriptors

J.2 [**Physical Sciences and Engineering**]: Engineering

General Terms

Algorithms, Performance.

Keywords

Computer vision, image processing, object recognition, ontology, context.

1. INTRODUCTION

The process of object recognition, within the context of computer vision, is the method by which an object is identified within visual images. The applications of object recognition include robotics, image storage and retrieval, automated surveillance, and aerial imagery analysis. In each of these applications areas, accurate object recognition is critical. Unfortunately, the performance of current techniques is not sufficient for the bare minimum functionality required for those applications.

This lack of performance is not due to a lack of research. To the contrary, object recognition has been well researched, with several texts dedicated to the topic [1]. A popular approach is to isolate objects within an image, calculate a set of features for that object, and then compare those features to known object types using pattern recognition techniques. While generic and powerful, this pattern recognition approach is inherently appearance based, and as such, tends to fail when object types look very similar. This weakness is exacerbated when the number of object types becomes large, thus increasing the likelihood that two object types will look very similar. Unfortunately, future applications of object recognition are moving in this direction. Without a significant leap in technology, current methods will not be able to address future challenges.

The current pattern-based approaches to date have shown limited success, but are still grossly inadequate, especially when compared to the object recognition capabilities of people. It stands to reason, that the performance of current techniques could be improved by borrowing from those methods that have been shown to be used by people. A key aspect used by people, and all but ignored by pattern matching techniques, is the image context [2]. The use of context for object recognition is not new [1]. It could also be argued that context is implicitly "hard coded" into an application. For example, a method for an outdoor scene might assume the sky is higher in an image than the ground. This "hard coded" approach is inflexible. A generic context-based object recognition approach will require flexible and efficient methods for storing and reasoning over contextual information. Such methods, while not developed specifically for computer vision and object recognition, have been developed for cognitive systems.

In this paper, we present several methods for using contextual information stored within an ontology to aid in object recognition in aerial imagery. The ability to capture aerial images is far outpacing the ability to analyze them, a task that is typically performed by people. The data resulting from this image analysis is used for anything from city planning to intelligence gathering. In order to be useful, this data must be up to date and accurate. The developed methods include two heuristics and one mathematical approach. The results of these methods are compared to classical generic object recognition algorithms to determine the relative effectiveness of context when combined with traditional pattern recognition techniques.

2. METHODS

2.1 Technical Approach

The software developed for this research contains three core components, an ontology, a classifier and a simulator. The ontology software component captures the system knowledge base. It stores the object features and relationships. The classifier software component contains all the classifiers developed or used

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.

for this project. This includes classical feature only classifiers and the newly developed context-based classifiers. The final software component, the simulation environment, generates objects to be classified and computes the resulting classifier statistics. The use of a simulated environment removes the need for image processing and segmentation, both complex problems. The simulation instead focuses this research on pattern recognition.

2.2 Ontology Structure

The Cyc (pronounced "psych") ontology was used for this project[5]. Cyc contains a general purpose framework for representing common sense knowledge. An initial survey of the topics stored within Cyc, found that it contained many of the concepts relevant to aerial imagery. Many of the object types found in an aerial image were already stored in Cyc. In fact, with only one or two exceptions, no new object types had to be entered into the ontology. In total, 49 object types, many of which looked very similar, were identified in the Cyc ontology for classification of overhead imagery.

Cyc also contained relevant spatial relations such as, near, parallel, spatially contains and others. Six relations were identified for use. The six base relations required mathematical and geometrical grounding. Given two objects, a mathematical formula was required to determine if the two objects were indeed related. For example, the Cyc relation *spatiallyContains* requires that one object contain another. Given location and size information for each object, determining if one object contains another is straight forward. Other relations, such as *near*, required more complicated formulations given the ambiguity in the relation itself.

Unfortunately, Cyc was missing the necessary assertions about the object types and their relations to be useful for aerial imagery analysis without modification. Over 100 assertions were made to the ontology. All assertions were limited to relating two objects. Positive assertions were made if, in general, a relation was found to hold. For example, in general, "Road vehicles are found near other road vehicles". Negative assertions were made if a relation was found not to hold. For example, "Modern-Houses are not found near highways". By way of Cyc's inheritance rules the 100 assertions resulted in almost 500 defined spatial relations between individual object types.

2.3 Classifiers

One of the goals of this research was to evaluate classical pattern matching methods against the newly developed contextual matching methods. For the purposes of this research two classical methods were chosen, Naïve Bayes[3] and Nearest Neighbor[4]. Both methods were taken from the Weka Data Mining library[6]. These methods are very similar in that both use training data to perform classifications. The training data was obtained from extracting images (a maximum of 10 images per object) of the desired objects from overhead images.

2.3.1 Generic Classifier

Figure 1 shows the generalized classifier. This implementation is similar to the Weka implementation; except that this implementation requires an entire object set for classification, whereas the Weka implementation works only on single objects. This is because, in order to make use of context, the object to be classified as well as the related objects must all be classified at once.

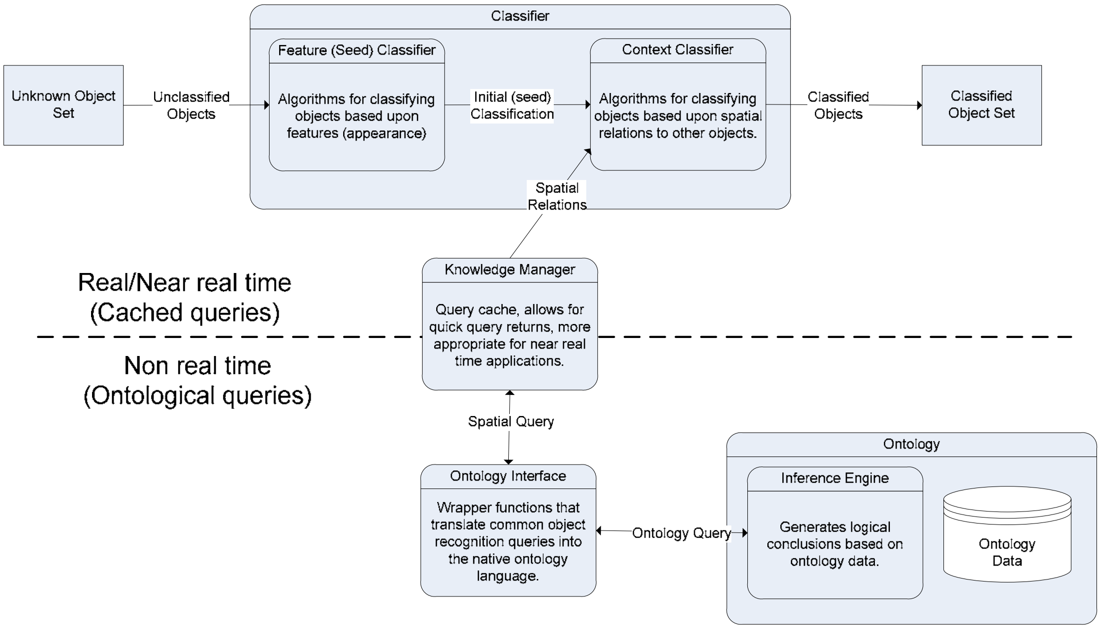

Classifier
Feature (Seed) Classifier
Algorithms for classifying objects based upon features (appearance)
Context Classifier
Algorithms for classifying objects based upon spatial relations to other objects.
Unknown Object Set
Unclassified Objects
Initial (seed) Classification
Classified Objects
Classified Object Set
Spatial Relations
Real/Near real time (Cached queries)
Non real time (Ontological queries)
Knowledge Manager
Query cache, allows for quick query returns, more appropriate for near real time applications.
Spatial Query
Ontology Interface
Wrapper functions that translate common object recognition queries into the native ontology language.
Ontology Query
Ontology
Inference Engine
Generates logical conclusions based on ontology data.
Ontology Data

Figure 1: Generalized classifier

Each classifier takes a set of unknown objects as an input. An initial seed classification is performed. For the purposes of this research, the seed classification is a feature classification, but nothing within the generic structure requires this. The seed classification could just as easily be another generic classifier. The seed classification is then refined by applying contextual information provided by the ontology. In order for contextual information to be applied, it must both be logical true and physically true. For example, given two objects, an airplane and an airport terminal building, the two are only contextually significant if logically the two objects would be found “near” each other (which is true) and if physically the objects are “near” each other (which may or may not be true). A classified object set is produced by the classifier. The classified set consists of the list of possible classes along with some measure of the match. Typically, this list is normalized so that the measure represents the probability of a match, but this isn’t required.

2.3.2 Combined Context Classifier

The combined classifier was the first and most simplistic context classifier developed. It is a heuristic for modifying the initial seed classifier given the number of arguments for and against the classification (as determined by the ontology).

In the first step, the results of the seed classifier are compared to a threshold constant. Those classifications that exceed the threshold are considered complete. Those that do not, are updated based on contextual information provided by the previously determined complete classifications. Two types of contextual information are used, supporting (represented by positive assertions within the ontology) and opposing (represented by negative assertions within the ontology). A counter is kept of arguments. Every piece of true (logical and grounded) supporting evidence increments the counter, and every piece of true opposing evidence decrements the counter. The seed classification is then scaled by the following factor shown in Figure 2.

$$
\begin{aligned}
&\mathit{where:}\\
&\qquad n = \mathit{evidence\ counter}\\
&\mathit{if}\ n \geq 1\\
&\qquad f = n\\
&\mathit{if}\ n < 1\\
&\qquad f = 1/n
\end{aligned}
$$

Figure 2: Combined classifier scaling factor definition

The scaling factor calculation multiplies the seed classification, if there are more arguments for a classification, and divides it if there are more arguments against the classification. The resulting values are then normalized in order to return probabilities of a match.

The pseudo code implementation for the combined context classifier is shown below.

```
for classification in seedClassification
    if classification >= threshold
        classifiedObjects.add(object)
    else
        unclassifiedObjects.add(object)

for object in unclassifiedObjects
    for relatedObject in classifiedObjects
        for relation in contextRelations
            for possibleType in objectClasses
                if positive relation holds for (object,
                possibleType, relatedObject)

                    argCount++
                if negative relation holds for (object,
                possibleType, relatedObject)

                    argCount- -

    updateSeedClassification(type, scaleFactor(argCount))

return updatedSeedClassification
```

2.3.3 Probability Combined Context Classifier

The probability combined classifier is very similar to the combined classifier (see section 2.3.2), except that classifications are not considered complete after the seed classification. Instead, all possible classifications and their corresponding probabilities are considered when tallying the number of supporting and opposing arguments. The amount the argument counter is incremented/decremented is the probability of a match (instead of +/- 1 as with the combined classifier).

The resulting classifier improves upon the combined classifier by including the probabilities of classification of other objects. In other words, a strong classification of a related object is more relevant than a weak classification. In addition, because a classification is not considered complete after the initial seed classification, this method can reject a false positive classification, even if it is a strong classification.

The pseudo code implementation for the probability combined context classifier is shown below.

```
for object in unclassifiedObjects
  for relatedObject in unclassifiedObjects
      for relation in contextRelations
          for possibleType in objectClasses
              for relatedType in
              seedClassification(relatedObject)

              if positive relation holds for (
                  object, possibleType, relatedObject,
                  relatedType)

                  argCount =
                      argCount + relatedType.measure

              if negative relation holds for (
                  object, possibleType, relatedObject,
                  relatedType)

                  argCount =
                      argCount - relatedType.measure

      updateSeedClassification(possibleType,
      scaleFactor(argCount))
```

```
return updatedSeedClassification
```

2.3.4 Bayesian Context Classifier

The Bayesian context classifier represents a more mathematical approach to using context for classification whereas the two previous methods were heuristics. In the simplest of terms, an object classification is either supported or opposed by the relations to the objects around it. In terms of probability, if a particular object classification is reaffirmed by the surrounding objects, then the probability of the classification should be increased. Bayes theorem (shown in Figure 3) describes mathematically how an existing probability is updated; given the observation of new evidence.

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)}$$

where :

$P(H \mid E)$ is the updated probability of the
hypothesis H given the evidence given E

$P(H)$ is the prior probability of the hypothesis
H before the evidence

$P(E \mid H)$ is the probability of the evidence
occuring given the hypothesis

$P(E)$ is the marginal probability of the evidence
or the probability of the evidence occurring
given all other possible hypothesies

Figure 3: Bayes thereom

Bayes theorem, in its standard form, does not apply to the specific application of using context for object recognition. Bayes theorem requires that the evidence must be observed, implying a probability equal to one. In object recognition, the probability of the evidence (or a particular classification of the surrounding object) is uncertain. This problem is common to other applications, and methods for adapting Bayes theorem have been developed[7]. The equations for updating the probability of a classification have been adapted to our application and are shown in Figure 4.

$$P(H \mid E') = \begin{cases} 0 \le P(E \mid E') \le P(E): \\ \qquad P(H) \\ \\ P(E) \le P(E \mid E') \le 1: \\ \qquad \dfrac{P(H) - P(H \mid E)P(E)}{1 - P(E)} + \\ \qquad P(E \mid E')\dfrac{P(H \mid E) - P(H)}{1 - P(E)} \end{cases}$$

where :

$P(H \mid E')$ is the updated $P(H)$ given the features
E' of the related object

$P(H)$ is the prior probability that the unknown
object is of a particular type (given
by seed classification)

$P(E \mid E')$ is the probability of the related object
classification given the features
E' of the related object (given by the seed
classification)

$P(H \mid E)$ is the probability of that the object type is of
a particular type given evidence E, given
by Bayes thereom with the following :

$P(E \mid H)$ is the probability that the
related object is observed given
that the unknown object is a
particular type. This is given by the
ontology where the $P(E \mid H)$ is 0.75
if a positive assertion between
the related object classification and
unknown object classification
exists. If a negative assertion exists
the $P(E \mid H)$ is 0.25. If no assertion
exists then existence of the evidence
results even odds or $P(E \mid H)$ is 0.5.

$P(E)$ is the probability of the evidence occurring given
all other possible hypothesies or $\sum P(E \mid H_i)P(H_i)$.

Figure 4: Probability of object classification given the features of a related object.

The Bayesian context classifier works very similarly to the probability combined context classifier in that all possible classifications of the unknown object and the related object are taken into account. However, the Bayesian approach allows the probability of an unknown object classification to be updated directly and recursively, removing the need for counting the number of positive and negative arguments.

The pseudo code implementation for the Bayesian context classifier is shown below.

```
for object in unclassifiedObjects
   for relatedObject in unclassifiedObjects
      for relation in contextRelations
         for objType in objectClasses
            for relatedType in
seedClassification(relatedObject)
         if positive relation holds for (
            object, objType, relatedObject, relatedType)
```

```
                updateSeedClassification(object, objType,
                relatedObject, relatedType)
        if negative relation holds for (
            object, objType, relatedObject, relatedType)
                updateSeedClassification(object,
                objType,relatedObject, relatedType)

return updatedSeedClassification
```

2.4 Simulation

The simulation environment (shown in Figure 5) generates the objects within the scene and provides a user interface in which to test and debug classification algorithms. Each scene is described by an xml file. The xml file is generated by hand classifying overhead images and contains the type (for evaluation of the classifiers), location (x, y) and orientation of each object. The simulator generates the object features for each object in the xml file; using information about the distribution of each feature obtained from the training data. In addition, the simulation can add uniform noise to each feature. The uniform noise is represented as a percentage of the feature value. It is important to note that although specific objects are simulated from statistics of an object sample set, the contextual relations are not. This is because context is determined by the location and orientation of each object, which is determined from actual overhead imagery.

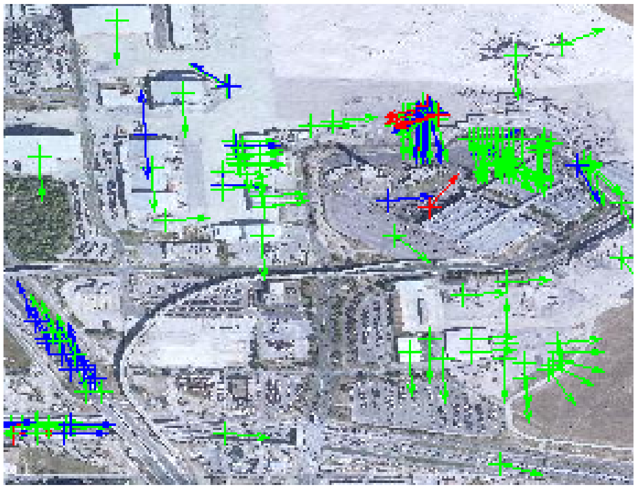


Figure 5: Simulation screen shot (San Antonio Airport test case). Markers indicate objects for classification. Blue markers indicate unknown objects, green indicate correctly classified, and red indicated incorrectly classified.

3. RESULTS

The three context-based classification methods were each compared to the classification results of the seed classifier alone. This comparison shows the increase or decrease in recognition that context provides. The classifiers were tested on aerial imagery of different types of areas, including an airport, industrial area, military base, downtown, and an agricultural area.

3.1 Classifier Optimization

The classifiers were optimized at three different random noise levels in the simulation. By optimizing the classifiers, the best classifier performance is achieved, allowing meaningful comparisons to be made between classifiers. The classifiers were optimized by finding the optimum points along the receive-operator characteristic (ROC) curve[8]. The ROC curve shows the tradeoff between correct classifications and incorrect classifications as some classifier parameter is varied (in this case, the threshold probability for a classification).

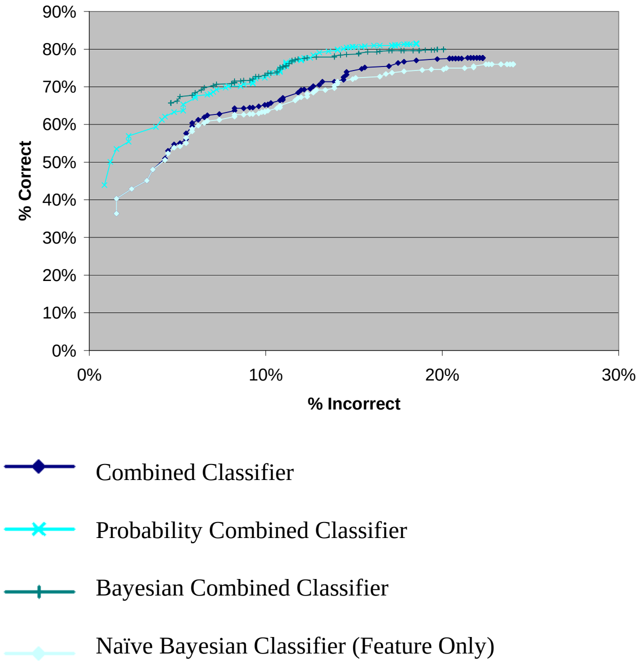


Figure 6: Typical ROC curve showing the three context-based recognition algorithms vs the seed classifier (Naïve Bayesian) alone

3.2 Classifier comparison

Table 1 and Table 2 show the optimized classifier comparisons, relative to their seed classifiers (Naïve Bayes and Nearest Neighbor, respectively) at different random noise levels. Two intermediate metrics are used to evaluate the difference between classifiers, the percent difference correct and percent difference incorrect. As can be observed from the ROC curves (Figure 6), there is typically a tradeoff between these two metrics. Assuming equal weight to these two metrics, taking the difference between them (%difference correct - %difference incorrect) indicates the value of the tradeoff. For instance, a reduction in both percent correct and percent incorrect would still have superior recognition as long as the %difference incorrect was greater than the %difference correct.

Table 1: Results of classifier simulations at varying noise levels versus a naïve Bayesian classifier. In almost all cases the context-based classifiers show better recognition (increased percent correct, decreased percent incorrect, or both).

Noise	Classifier	Sum of Number Correct	Sum of Number Incorrect	% Difference From Number Correct	% Difference From Number Incorrect	**Total Percent Difference**
0%	Naïve Bayes	2938	1930			
	Bayesian Context Classifier	2496	1221	-15.04%	-36.74%	**21.69%**
	Combined Context	3165	1543	7.73%	-20.05%	**27.78%**
	Probability Combined Context	3909	1100	33.05%	-43.01%	**76.05%**
10%	Naïve Bayes	1976	1523			
	Bayesian Context Classifier	2349	1932	18.88%	26.85%	**-7.98%**
	Combined Context	2534	1928	28.24%	26.59%	**1.65%**
	Probability Combined Context	2582	1775	30.67%	16.55%	**14.12%**
25%	Naïve Bayes	1010	1357			
	Bayesian Context Classifier	1138	1445	12.67%	6.48%	**6.19%**
	Combined Context	1047	1367	3.66%	0.74%	**2.93%**
	Probability Combined Context	1115	1062	10.40%	-21.74%	**32.14%**

Table 2: Results of classifier simulations at varying noise levels versus nearest neighbor classifier. In almost all cases the context-based classifiers show better recognition (increased percent correct, decreased percent incorrect, or both).

Noise	Classifier	Sum of Number Correct	Sum of Number Incorrect	% Difference From Number Correct	% Difference From Number Incorrect	**Total Percent Difference**
0%	Nearest Neighbor	3829	1020			
	Bayesian Context Classifier	3798	900	-0.81%	-11.76%	**10.96%**
	Combined Context	3870	771	1.07%	-24.41%	**25.48%**
	Probability Combined Context	3832	741	0.08%	-27.35%	**27.43%**
10%	Nearest Neighbor	3618	975			
	Bayesian Context Classifier	3658	915	1.11%	-6.15%	**7.26%**
	Combined Context	3695	847	2.13%	-13.13%	**15.26%**
	Probability Combined Context	3617	790	-0.03%	-18.97%	**18.95%**
25%	Nearest Neighbor	2313	1163			
	Bayesian Context Classifier	2248	813	-2.81%	-30.09%	**27.28%**
	Combined Context	2400	1350	3.76%	16.08%	**-12.32%**
	Probability Combined Context	2514	1210	8.69%	4.04%	**4.65%**

4. Conclusion and Discussion

Table 1 and Table 2 show that, in most cases, recognition rates of the context-based classifiers are superior to the feature only classifiers. It was hoped that the context classifiers would show robustness under increasing amounts of noise. However, recognition actually decreased with increasing noise. The application of noise causes a large proportion of the seed classifications to be weak classifications. That is, instead of a single strong match, several types are matched to a single object. Strong matches are required in order to establish context and in turn reinforce weaker matches. It stands to reason that fewer strong matches will result in weaker reinforcement of weaker matches, and thus fewer correct classifications.

The data does not support a single best implementation of the context classifier. This is not unexpected given that is the nature of most classifiers, with each performing differently under different applications. The popularity of the Bayesian approaches to classification could be attributed to the strict mathematical approach. While the Bayesian context classifier has a similar mathematical basis, the probabilities associated with ontological assertions were arbitrary. Better results may have been achieved if more accurate probabilities were used. Accurate probabilities for relations could be developed over time by adding a learning component to the ontology.

The focus of this work was to evaluate methods for integrating contextual knowledge into object recognition techniques. The Cyc ontology was chosen for this task due to its breadth and depth of knowledge. While some knowledge was added for the purposes of this research, the criticality of that knowledge was not evaluated. For the application of object recognition in aerial imagery, real-time performance was not a constraint. However, for most object recognition applications real-time performance will be required. For these applications, the size and structure of the knowledgebase will be critical. More research is required to answer the questions: What is the optimum type and amount of information required for context-based recognition?

5. ACKNOWLEDGMENTS

The authors would like to acknowledge Southwest Research Institute for funding this research. In addition, they would like to recognize the contributions of Ian Mienzen for developing much of the software that enabled this research.

6. REFERENCES

[1] D. H. Ballard and C. M. Brown, Computer Vision. Englewood Cliffs, NJ. PrenticeHall, 1982

[2] Bar, M. (2004). Visual objects in context. Nature Neuroscience Reviews, 5, 617-629.

[3] George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.

[4] D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.

[5] Lenat, D. B. 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. The Communications of the ACM 38(11):33-38

[6] Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[7] R. Duda, P. Hart, and N. Nilsson, ªSubjective Bayesian Method for Rule-Based Inference System,º AFIPS, vol. 45, pp. 1,075-1,082, 1976.

[8] T. Fawcett, Roc graphs: Notes and practical considerations for researchers," http://www.hpl.hp.com/personal/Tom Fawcett/papers/ROC101.pdf

Manufacturing Unit Process Life Cycle Inventories (uplci)

Michael Overcash
Wichita State University
Wichita, Ks 67260
mrovercash@earthlink.net

Janet Twomey
Wichita State University
Wichita, KS 67260
janet.twomey@wichita.edu

Jackie Isaacs
Northeastern University
Boston, MA
jaisaacs@coe.neu.edu

ABSTRACT

Tools to make environmentally informed product decisions at the design stage have long been an identified need of the manufacturing community. Potential solutions to address this need have been one of the topics of a series of National Science Foundation funded international workshops in the area of environmentally benign design and manufacture (EBDM). This paper reports one outcome of those workshops and the progress toward the development of a new approach to use manufacturing unit processes as the basis for evaluating environmental impacts at the manufacturing phase of a product's life cycle. The research presented here is funded through a Department of Energy award DOE DE-FG36-08GO88149.

General Terms

Measurement, Documentation, Design.

Keywords

Manufacturing, environmental, unit process, life cycle inventory, sustainability.

Permis'09, September 21–23, 2009, Gaithersburg, MD, USA.
Copyright 2009 ACM 978-1-60558-747-9/09/09…$10.00.

1. INTRODUCTION

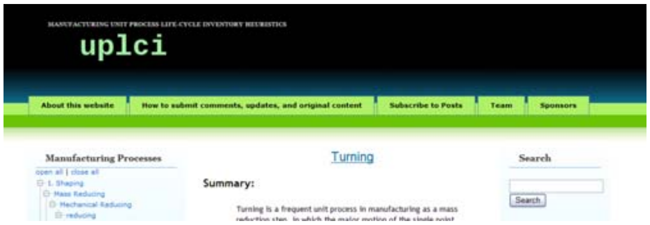
Tools to make environmentally informed product decisions at the design stage have long been an identified need of the manufacturing community. Potential solutions to address this need have been one of the topics of a series of National Science Foundation (NSF) funded international workshops in the area of environmentally benign design and manufacture (EBDM). Dr. Delcie Durham, Program Officer at the NSF (now at University of South Florida) created the workshops as a means for the international community to collaborate in the creation of new knowledge in EBDM. This paper reports on a collaboration formed as an outcome of those workshops and the progress toward the development of a new approach to use manufacturing unit processes as the basis for evaluating environmental impacts of a product's life cycle at the manufacturing phase.

2. uplci PROJECT

The new approach is developed to use the manufacturing unit process, commonly outlined in manufacturing process taxonomy systems, as the basis for life cycle inventory. This will initially involve 50-70 unit processes from the taxonomy and will generate energy and mass profiles for each unit process life cycle (uplci). These uplci can be adjusted for each case to include the major variables affecting such operations as related to any specific product. The sum of the performance of a sequence of uplci thus provides the life cycle of the specific product from a defined set of plant process inputs [1].

The research supporting the development of the uplci is located at Wichita State University, Wichita, KS. A website (www.wichita.edu/sustainability) has been established to allow access to the uplci (see Figure 1). The website is also intended for the community to supply comments and submit it additional unit processes.

The Wichita State University project is currently funded through a U.S. Department of Energy award (DOE DE-FG36-08GO88149) in wind energy. The DOE has interest in this area because the primary environmental impacts of wind energy are at the manufacturing phase. uplci will be used as means for making comparisons of wind energy with sources other than coal and throughout the turbine life cycle.



Turning is a frequent unit process in manufacturing as a mass reduction step. In which the major motion of the single point

Figure 1. Snap shot of uplci website

3. REFERENCES

[1] Overcash, M., Twomey, J. and Kalla, D. (2009). ASME International Manufacturing Science and Engineering Conference October 4-7, 2009, West Lafayette, IN, USA

Conceptual Foundations of Energy Aware Manufacturing

Soundar Kumara
The Pennsylvania State University
310 Leonhard Building
University Park, PA 16802
(814)863-2359

skumara@psu.edu

ABSTRACT

In this paper we define "Energy Aware Manufacturing (EAM)," as the "*Manufacturing paradigm that concentrates on integrated optimal usage of energy in manufacturing right from material procurement to part disposal.*" The main emphasis of EAM is on an integrated optimal energy usage. The focus is to model energy related issues in the interactions between suppliers, production machines, part manufacturing and plant physical facilities. There is a need to develop quantitative models and information models and implementing a simulation test bed to accomplish the objective of EAM. In this paper, we develop a conceptual roadmap for accomplishing the objective of energy efficiency. We discuss the conceptual foundations of our approach.

Categories and Subject Descriptors

Performance, Management

General Terms

Management, Measurement, Performance

Keywords

Sustainability, Energy Aware Manufacturing, Energy efficiency

1. INTRODUCTION

If manufacturing sector in the USA were to be a country by itself, it would be the 8th largest economy in the world. The manufacturing sector in the USA in 2005 accounts for 16% of the national GDP, and is still the world's largest manufacturer. In 2004, it accounted for a quarter of the global production. Though the current figures are slightly dipping (by about 1.1%) still USA holds its superiority in the manufacturing sector. China and South Korea are increasing their share in manufacturing which is mainly due to the increase in non-direct material related costs such as health care, labor, and energy in the USA. ***Given the increase in global demand for energy any attempt to improve energy efficiency will lead to considerable value addition to not only to the US economy but to the global welfare*** (World Bank and Economy.com, 2009; US EPA, 2007).

Energy efficiency is a global issue and cannot be solved by unilateral and myopic approaches. Energy efficiency can be achieved by proper energy management, and is a multidisciplinary challenge involving science, technology, ecology, information technology and common sense (ICT, 2009). Energy management is critical to the sustainability of the future of US and Global economy. Optimizing energy usage and minimizing energy loss are central to energy management. It is important that manufacturing integrate this concept into its operations from procurement to maintenance or from cradle to grave. Energy Aware Manufacturing (EAM), which focuses on the energy management in manufacturing, is central to reducing the carbon footprint of manufacturing industries. We define EAM as the ***"Manufacturing paradigm that concentrates on optimal usage of energy in manufacturing right from material procurement to part disposal."*** We believe that the state-of-the-art research in commercial as well as academic domains though focuses on sustainability, has not addressed an integrated methodology development for efficient use of energy from conception to grave of a product. Such an integrated development must deal with information models, quantitative models, metrics for energy usage, and development of standards. Efficient Tools and methods for estimation of the total energy consumption of a complete manufacturing system are missing today. To be able to calculate the most optimal use of energy, new tools, and methods must be developed (Asnafi et al., 2008). This forms the central theme of our conceptual paper. Energy labeling is already in progress in Europe. In this paper we propose the concepts for developing manufacturing part label which will incorporate manufacturing energy as a part of it so that we can track the carbon footprint of each part.

2. THE NEED

We argue that energy is an important consideration in manufacturing, both from process as well as plant operational view. Though energy efficiency has become important in the last decade, it is necessary to establish a scientific basis for including reduction in energy consumption as one of the focuses of manufacturing. We need to define, develop, and establish standards to evaluate the total energy consumption by the entire manufacturing operation. With out consideration to the legal and political issues, it may be beneficial for the society to have "Carbon Labeling" for every manufactured part, akin to food labeling. Due to the complexity of this problem, there is a clear

Permis'09, September 21–23, 2009, Gaithersburg, MD, USA.
Copyright 2009 ACM 978-1-60558-747-9/09/09…$10.00.

and urgent need to develop quantitative and information based models for energy management.

In order to realize EAM, we postulate three important research streams. We do not address the process efficiency aspects. Though they are critical we consider EAM from a systems view point.

Research Stream1: Define and implement "*Manufacturing Energy Computations*" that will help in generating quantitative models to realize EAM.
Research Stream 2: Define and implement "*Manufacturing Energy Information Modeling*" that will help in representing the models from stream 1.
Research Stream 3: Design, develop and implement "*Manufacturing Energy Simulation Modeling*" to establish a simulation framework and a platform that will use the information and the quantitative models developed to study the dynamics of manufacturing.

3. THE NEED

Environmental regulations, such as European Union's Integrated Product Policy (IPP) and the EU's directive on the Ecodesign of Energy-Using Products (EUPs), will directly regulate the negative contribution to the environment across the entire lifecycle of the product, not just the use phase. This implies that the environmental impact of any given product will be addressed by taking all aspects of its supply chain and life cycle into account: raw materials, components/part sub-assembly/final product manufacturing, transportation, distribution, marketing, sales, delivery and waste treatment at the end of life. Therefore, the carbon footprint and Green House Gas (GHG) emission accounting for a product will be directly affected by:

1. The manufacturing processes as well as the equipment used for producing the product. For example, in the vehicle body shop in an automobile manufacturing company, the energy consumed and GHG emitted by spot welding process will be different compared to the ones by laser welding process. It should be noted that a product can be a complete vehicle or a component (e.g., a wheel) or a sub-assembly (e.g., an instrument panel).
2. The supplier footprint and logistics (transportation) of components/parts for the final product are integral to the carbon foot print of the part. Take engine assembly as an example, if the engine block and cylinder head are manufactured in an engine assembly plant located in US, whereas the other components such as pistons, crankshaft, and camshaft, could be purchased and shipped from supplier(s) located in US, India, or China. Therefore, the final carbon footprint and GHG emission accounting for the engine assembly will depend on where each of all the engine components is made and how it is shipped from its supplier to the final engine assembly plant.

Another analysis and implementation exercise is closely tied to the second point stated above, i.e., the carbon footprint/GHG emission accounting based on the components/parts' supplier footprint. With the calculation/measurement of the carbon footprint/GHG emission of each component/part/sub-assembly, product manufacturer can implement its "green procurement" strategy to set up and maintain its supply chain to be the one having lowest energy consumption and safest impact on the environment.

4. RESEARCH STREAM 1: ENERGY COMPUTATION

Increasing raw material prices, necessary investments in the environmental technologies, potential penalties for not complying with regulations as well as ability to attract incentives and public image is forcing manufacturing companies to rethink about energy related issues (Hesselbach et al., 2008). Manufacturing plants from an energy perspective are integrated units of building and production machines. Energy efficiency mainly relates to optimizing the ratio of production output to the energy input for the technical building services (heating and cooling) and production machines. Energy efficiency is a generic term and there is no one uniform quantitative measure of "energy efficiency." instead one must rely on a series of indicators. Patterson (Patterson, 1996) discussed four indicators (Thermodynamic, Physical-Thermodynamic, Economic-thermodynamic, and Economic) of energy efficiency. These cannot be used directly in manufacturing; however, we can use some variants of the concepts. We extend the traditional definition of energy efficiency by including the supply chain also into its computation. Figure 1 reproduced from Hesselbach (Hesselbach et al., 2008) shows a holistic view of the facility-production machines without supply chain interface from an energy awareness viewpoint.

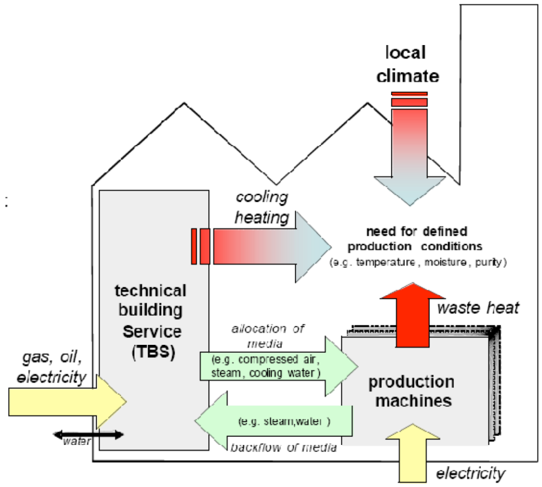

:
local climate
cooling
heating
need for defined production conditions
(e.g. temperature, moisture, purity)
technical building Service (TBS)
waste heat
allocation of media
(e.g. compressed air, steam, cooling water)
gas, oil, electricity
production machines
(e.g. steam,water)
water
backflow of media
electricity

Figure 1: Facility-Production Machines Holistic View (Reproduced from (Hesselbach et al., 2008))

We propose a part energy profile comprising of three components (see Figure 2), namely:

1. Procurement component
2. Production component
3. Delivery component

These energy component values can be calculated from either top down or bottom up fashion. In the first method, the aggregate energy for the procurement, production, and delivery of a batch of products and assemblies is obtained from energy metering and apportioned to individual components. In the second approach individual component, energy usage values are obtained through individual unit procurement, production, and delivery and aggregated for the assembly. In both the cases due to the inherent uncertainties in the operations, stochastic estimation methods are needed. Bottom up approach though extremely cumbersome may yield results that are more accurate.

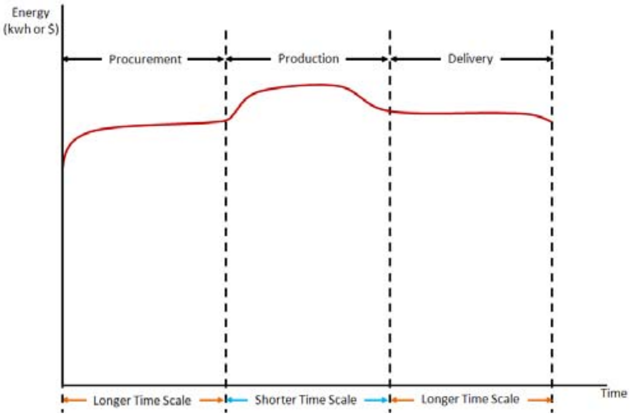


Figure 2: Part Energy Profile

For an identified critical part we generate the part energy profile based on the following steps:

1. Generate alternate process plans
2. Identify the alternate processes
3. From material removal rate computations calculate energy requirements using alternate processes (and hence alternate production machines)
4. Identify alternate suppliers and if possible next tier suppliers
5. Identify alternate transportation facilities(modes and equipment) and routes
6. Compute energy expenditures (Kwh)

The energy parameters to consider are:

1. Process plans
2. Manufacturing Processes
3. Production Machines
4. Suppliers
5. Transportation Modes
6. Transportation Facilities (equipment)
7. Packaging modes

Metrics for EAM: The metrics needed to evaluate EAM are: Supply Chain Related, Physical Facility Related, Production Machine Related, and Production Cell Related. In all these categories, we need to formulate metrics for: 1. Energy Wastage due to not utilizing set-up properly on a machine (a batch of 100 parts would have for example made the optimal use of a set up as opposed to producing 50 units), 2. Energy Wastage due to not utilizing set-up properly in a production cell (a batch of 100 parts would have for example made the optimal use of a set up as opposed to producing 50 units), 3. Energy wastage due to improper re-ordering, and 4. Cost of non-compliance to regulations (now most of these are voluntary), and 5. Return on Investment.

In general, in the industrial sector the total energy used per unit time (month or a day) is expressed as a combined component of the facility energy expenditure (E_F) and production machines energy expenditure (E_P).

$$E_F + E_P = E_T$$

This is basically the power consumed in Kilo-watt-hours (Kwh). For example, E_F can be 0.4 of E_T; and E_P therefore can be 0.6 of E_T. This when related to the total production in unit time

$$\frac{0.6E_T}{Total\ \Pr oduction} = E_U$$

Where E_U is Energy consumed per unit of the product. This can be weighted by individual processes and machine rating. This of course is a gross measure. Detailed methodological aspects need to be developed.

5. RESEARCH STREAM 2:INFORMATION MODELING

We view manufacturing systems as comprising of five levels (we do not define them as hierarchically related):1. Supply chain level, 2. Company level, 3. Production system level, 4. Production cell level, and 5. Machine level

Thought there are several commercial systems available to represent and analyze each of these systems from different viewpoints there is very little attention paid to the energy related aspects. Similarly there are commercial (Sahlin et al., 2004; Chen et al., 2001) building and HVAC simulators, but each one with a specific objective of computing total energy consumed vis-a-vis wasted. Very little work exists in relating production machines to the facility. To the best of the author's knowledge, no work exists in relating manufacturing with supply chain with respect to energy issues. We need to define a robust information model before we can build a tool and standards for analyses. We suggest an integrated information model based on Service Oriented Computing (SOC), which is an increasingly popular and an efficient paradigm for collaborative planning and execution.

Services are self-describing, open components that support low cost composition of distributed applications. Services are offered by service providers who are distributed over the web (web services). Service descriptions are used to advertise the service capabilities, interface, behavior, and quality. Service description provides the conceptual purpose and expected results of the service. The service interface description publishes the service signature (input, output, error parameters, and message types). The expected behavior of the service during its execution is described by service behavior description (for example aggregate energy consumption trace). Quality refers to service cost, performance metrics and security attributes (Papazoglou & Georgakopoulos, 2003). When these are offered through the web, they become web services.

Our vision is to utilize Service Oriented Architecture (SOA), which is the most advanced Web-based service system architecture (Erl, 2004), to formalize an IT framework to support EAM. The main aspects to consider are:

1. Develop an interface-oriented machine-readable representation scheme for components;
2. Formalize an accessible cyberinfrastructure-based framework that enables global users to describe, publish, and discover component information in a standardized way; and

3. Adapt Artificial Intelligence (AI) planning algorithms to support energy computations.

The National Institute for Standards and Technology (NIST) has proposed the use of eXtensible Markup Language (XML) for the description of functions and associated flows in computer-based design (Szykman, et al., 1999; 2002; Bohm, et al., 2008) introduced an extensive data schema to capture fundamental elements of design information. These representation schemes for products or parts are designed mostly from the viewpoint of the materials or energy flow, which is sequential or flow-oriented; however, research efforts towards EAM, demands a different viewpoint for representation. We assert that the new viewpoint should be *interface-oriented* because physical interfaces help bridge modularized components (sub-assemblies). Figure 3 shows a SOA-based Web Services for Energy Aware Manufacturing. Conceptually, in our proposed framework, a Universal Description Discovery and Integration (UDDI) would correspond to the digital Design Repository implemented by the Missouri University of Science & Technology and NIST (Bohm & Stone, 2004; Szykman & Sriram, 2006) while a Web service would correspond to a production process.

Component designers describe components in a standard language such as XML (eXtensible Markup Language) version of STEP (the Standard for the Exchange of Product). The description of each component includes input/output interfaces, dimensions, features, and required energy efficiency of it, etc. On the other hand, manufactures describe their manufacturing processes also in a standard language. The description of each manufacturing process in Web Service Description Language (WSDL) includes machines, tools, coolants, and energy rating, etc. Those standardized descriptions are stored in the SOA-based infrastructure for energy-aware manufacturing (Figure 3). SOA provides subscribe/publish mechanism for users, component designers and manufacturers. They can access those descriptions in an automated way. Three types of services can be provided by the proposed architecture, atomic, composite, and managerial services. Atomic services can respond to simple queries, such as getComponent, getMfgProcess, and getEnergyEfficiency; composite services can satisfy sophisticated queries, such as checkEnergyEfficiency (of a product) and designProduct; finally, managerial services provide administrative transactions for users. Those services must be open to public(or a networked manufacturing system) through a secured authentication/authorization protocol, and discoverable in a standardized way.

As we are interested in studying the interaction between the physical facility and the production machines, it is important to represent the building related entities. EAM needs to consider the following services related to the physical facility (building):

1. Lighting service: related to lighting aspects of the facility containing total lighting units, area, and energy requirement with respect to the season and time of the day
2. Heating service: related to heating aspects of the facility containing total area, heating units, relationship to production machines, and energy requirement with respect to the season, time of the day and personnel on duty.
3. Cooling service: related to cooling aspects of the facility containing total area, and energy requirement with respect to the season, time of the day and personnel on duty.

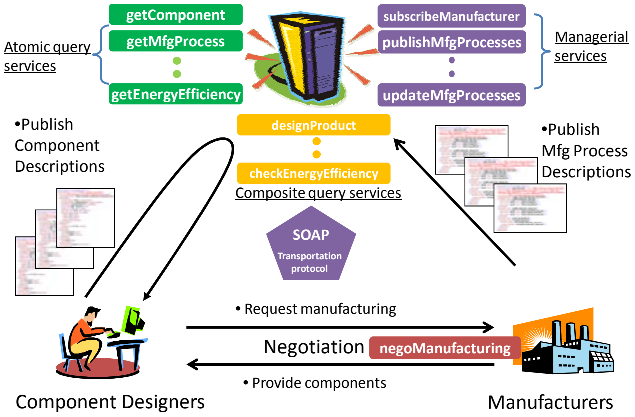


Figure 3: SOA for Energy-Aware Manufacturing

4. Airflow pumping service: Airflow requirements based on production, time of the day, season and personnel.
5. Energy computation service: Aggregation of energy requirements calculations based on the component manufacturing plan (one can use AI planning algorithm and integrating with energy constraints of the physical facility to accomplish this).

6. RESEARCH STREAM 3:SIMULATION MODELING

The third research stream relevant to EAM is the estimation of energy consumption that minimizes the overall carbon foot print of the manufactured part. We would also like to ask what-if questions (study the dynamics) considering the stochastic nature of suppliers, machines, physical facilities and environment. We need to integrate the five levels from supply chain to component (discussed earlier). Though the information model we suggest is powerful to compose the processes and hence machines needed, and the suppliers to be selected, at this stage the SOA technology is not mature enough for building a simulator. In this section, we develop a conceptual framework for a multi agent based simulator for integrated energy analyses in manufacturing.

Agent-based computation is a new paradigm of information and communication technology that largely shapes and, at the same time, provides supporting technology to the above trends (Luck et al., 2005; Wooldridge, 2000; Weiss, 1999). Agent theories and applications have appeared in many scientific and engineering disciplines. Agents address autonomy and complexity; they are adaptive to changes and disruptions, exhibit intelligence and are distributed in nature. In this setting computation is a kind of social activity. Agents can help in self-recovery, and react to real-time perturbations.

The Core Manufacturing Simulation Data (CMSD) Information Model (CMSD, 2006) captures the essence of manufacturing simulation information. The UML based description can be used as a foundation to build the M-EAMS(Multi Agent Based Energy Aware Manufacturing Simulator).

We identify the relevant agents from these information categories. Some of these are: Supplier Agents, Transportation Agents representing different transportation resources, Calendar Agent for storing all the relevant calendar information and pushing it as and when needed, Resource Agents each representing one of the resources, Employer Agent interfaced with skill database, Process Planning Agent, Component(Part) Agent, Sub-assembly Agent (containing all the components needed for Sub-assembly), Final Assembly Agent (containing all the components needed for Final Assembly), Inventory Agent, Maintenance Agent, Scheduling Agent, Energy Computation Agent (having decomposition and aggregation algorithms as a part of agent behaviors), Visualization agent and Metrics Calculator Agent. We also identify certain agents related to the physical facility representing Lighting, Heating, Cooling, Air-Flow and Building Energy Computation Agents.

In certain cases, for example as in the case of Process Planning Agent, Maintenance Agent, and Scheduling Agent COTS is available that can be a part of the simulator. We can pass a message from the Part Agent to the Process Planner by wrapping the COTS Process Planner. In this case we need to focus on the ontological equivalences in communication primitives between the Part Agent and the Process Planner. Once we have a Process Plan, it can be communicated to a scheduling agent in a similar fashion. We undertake an explanation of the operational architecture of the EAM simulator (see Figure 4). We define four communities: Supplier Community: Composing of suppliers, Transportation Units-interfaced with carrier networks, and GIS Database

Physical Facility Community: Consisting of heating, lighting, cooling other amenities represented as agents interfaced with energy providers and cost databases.

Part Community: This contains components/parts/sub-assembly/Assemblies. All the geometric and material, process related information is stored in these agents.

Production System Community: Composing of machines, other resources such as materials handling equipment, personnel, and monitoring devices and sensors.

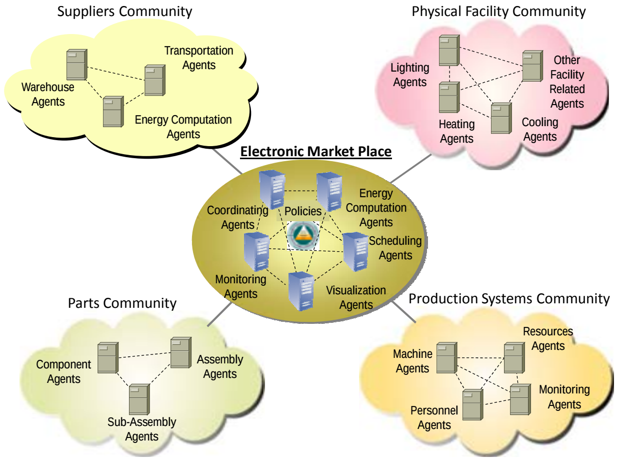


Figure 4: Operational Architecture for Multi Agent Based EAM Simulator

This agent organization captures the five levels of a manufacturing system. Here we deliberately break away from the hierarchical relationship traditionally used to model manufacturing systems. We view the integrated manufacturing systems as a collection of loosely coupled communities. In each of these communities, we will have Monitoring (sensors and RFID devices) agents, Energy Computation agents and scheduling agents. We need to build the simulation platform using these four communities that capture the static information through agents. However, each of the communities will have their own monitoring agents that monitor the community and take control (adaptive) actions. These are local, guided by global metrics communicated from the electronic market place.

The stochasticity of each of the communities and the relevant individual agents (for example machine breakdown, cost variation, Energy fluctuation etc) is injected externally and sensed by the monitoring agents. The electronic market (EM) place offers the means to carry out the overall dynamic optimization of EAM. The operation of the simulator can along the following lines. 1. Components/Sub-Assembly/Assembly request initiated by part community. 2. EM- Monitoring agent senses the information and invokes the process planning process through a Process Planning agent. 3. EM- Market Place Communicates information to Production Facility and supplier community (Bid Announcement). The utility function should be a composite of cost and energy requirements. EM will interface with the BOM or MRP to compute material requirement. 4. Machines and other resources bid to carry out the operations. When we have alternate process plans, we have different machines bidding with different energy budgets based upon their own ratings. 5. Similar process for materials will be initiated by the supplier community. 6. Physical facility agents send their bids on energy based upon their criteria of the time of the day, environmental conditions etc. 7. EM-Coordinating agent will have all these information and will initiate through a market based control mechanism to arrive at a Nash equilibrium that will corresponds to a solution that can no longer be improved by individual communities. 8. Given the current solution, the EM- Coordinating agent will invoke different regulations (through the policies) to see their effect on the solution. 9. After performing policy impact analysis, EM-coordinating agent will communicate the detailed plans to individual communities for final acceptance. 10. The visualization agents can be invoked to see dynamic changes in each community and each agent.

7. CONCLUSIONS

In this paper, we propose a conceptual framework to undertake the development of the paradigm of EAM. We elaborated on three research streams that are critical: Energy Computation, Information Modeling and Simulation Development. The next step is to develop the details related to each of these streams. Our research so far on EAM shows that a well-designed simulator is essential to define and enumerate performance metrics for EAM.

8. ACKNOWLEDGMENTS

The author wishes to thank Dr. Arun Ayyagari, of Boeing; Dr. Xiang Zhao, Dr. Ninja Huang and Dr. Jeffrey Tew for their discussions on the concept of EAM. However, the writing in this paper expresses the author's views only.

9. REFERENCES

(Asnafi, 2008). Asnafi, N., et al., 2015 Sustainable Manufacturing Systems Capable of Protecting Environmentally Friendly and Safe Products, R&D Proposal, Dec. 2008

(Bohm & Stone, 2004). Bohm, M. and Stone, R., Representing Product Functionality to Support Reuse: Conceptual and Supporting Functions, DETC2004-57693, Proceedings of DETC2004, Salt Lake City, UT, 2004

(Bohm, et al., 2005). Bohm, M. R., Stone, R. B. and Szykman, S., Enhancing Virtual Product Representations for Advanced Design Repository Systems, Vol. 5, pp. 360-372, 2005

(Chen et al., 2001). Chen, T. Y., Burnett, J., and Chau, C. K., Analysis of Embodied Energy Use in the Residential Building of Hong Kong, Energy, Vol. 26, pp. 323-340, 2001

(CMSD, 2006). Core Manufacturing Simulation Data information model part 1: UML model, Draft Product Development Group, Simulation Interoperability Standards Organization, Available via, http://discussions.sisostds.org/default.asp?action=9&boardid=2&read=39532&fid=24

(Erl, 2004). Erl, T. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services. Prentice Hall, Upper Saddle River, NJ, 2004

(Hesselbach et al., 2005). Hesselbach, J., Junge, M., Herrmann, C., and Dettmer, T., Production focused life cycle simulation, In: Proceedings of the 38th CIRP International Seminar on Manufacturing Systems, Florianópolis, Brasilien, 2005.

(ICT, 2009). ICT and Energy Efficiency-The case for Manufacturing, Recommendations of the Consultation Group, European Commission, Information Society and Media, Europea Communities, February 2009

(Luck et al., 2005). Luck, M., McBurney, P., Shehory, O., et al., Agent Technology: Computing as Interaction. A Roadmap for Agent-Based Computing, AgentLink III, http://www.agentlink.org/roadmap/al3rm.pdf, 2005

(Papazoglou & Georgakopoulos, 2003). Papazoglou, M.P. and Georgakapoulos, G., Service-Oriented Computing, Commun. ACM, Vol. 46(10), pp. 24–28, 2003

(Patterson, 1996). Patterson, M.G., What is energy efficiency? Concepts, indicators and methodological issues, Energy Policy, Vol. 24(5), pp. 377-390, 1996

(Sahlin et al., 2004). Sahlin, P., Eriksson, L., Grozman, P., Johnsson, H., Shapovalov, A. and Vuolle, M., Whole-building simulation with symbolic DAE equations and general purpose solvers, Building and Environment, Vol. 39(8): pp. 949-958, 2004

(Szykman, et al., 1999). Szykman, S., Racz, J., and Sriram, R., The Representation of Function in Computer-Based Design, Proceeding of DETC99/DTM-8742, Las Vegas, NV, 1999

(Szykman & Sriram, 2006). Szykman, S. and Sriram, R.D.,Design and implementation of the Web-enabled NIST design repository, ACM Transactions on Internet Technology (TOIT), Col. 6(1), pp. 85-116, 2006

(US EPA, 2007). Energy Trends in Selected Manufacturing Sectors: Opportunities and Challenges for Environmentally preferable energy Outcomes, Final Report, Prepared by ICF International, Fairfax, VA, March 2007

(Weiss, 1999). Weiss, G., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, 1999

(Wold Bank and Economy.com, 2009). World Wide Web, http://www.worldbank.org

(Wooldridge, 2000) Wooldridge, P., An Introduction to Multiagent Systems, Addison-Wesley, Reading, MA, 2000

Discrete Event Simulation to generate Requirements Specification for Sustainable Manufacturing Systems Design

Björn Johansson
Product and Production Development
Chalmers University of Technology
SE-412 96 Gothenburg, SWEDEN
bjorn.johansson@chalmers.se

Anders Skoogh
Product and Production Development
Chalmers University of Technology
SE-412 96 Gothenburg, SWEDEN
anders.Skoogh@chalmers.se

Mahesh Mani
National Institute of Standards and Technology
100 Bureau Drive
20899 Gaithersburg, MD, USA
mahesh.Mani@nist.gov

Swee Leong
National Institute of Standards and Technology
100 Bureau Drive
20899 Gaithersburg, MD, USA
swee.Leong@nist.gov

ABSTRACT

A sustainable manufacturing systems design using processes, methodologies, and technologies that are energy efficient and environmental friendly is desirable and essential for sustainable development of products and services. Efforts must be made to create and maintain such sustainable manufacturing systems. Discrete Event Simulation (DES) in combination with Life Cycle Assessment (LCA) system can be utilized to evaluate a manufacturing system performance taking into account environmental measures before actual construction or use of the manufacturing system. In this paper, we present a case study to show how DES can be utilized to generate requirements specification for manufacturing systems in the early stages of the design phase. Requirement specification denotes the description of the behavior of the system to be developed. The case study incorporates use of LCA data in combination with DES. Data for the model in the case study is partly provided through the format supported by the Core Manufacturing Simulation Data (CMSD) standardization effort. The case study develops a prototype paint shop model, and incorporates alternate decisions on energy use, choice of machines, and environmental bottleneck detection. The study results indicate the potential use of utilizing DES in combination with LCA data to generate requirements specification for designing sustainable manufacturing systems.

Categories and Subject Descriptors

J.6 [**Computer Applications**]: COMPUTER-AIDED ENGINEERING – *Computer-aided manufacturing (CAM)*

General Terms

Management, Measurement, Performance, Design, Economics, Experimentation, Standardization.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
Copyright 2009 ACM 978-1-60558-747-9/09/09…$10.00.

Keywords

Discrete Event Simulation, Life Cycle Assessment, Design for Sustainability, Manufacturing System Design, Standardization

1. INTRODUCTION

Requirements specification plays a vital part during design reviews when designing sustainable manufacturing systems. DES can be potentially used to generate requirements specification after considering what-if scenarios and analyzing alternative models to reflect how a system performs in implementation. This paper discusses how sustainability factors can be incorporated in defining requirements specification using DES to provide decision support for a more sustainable environment and society.

The paper is organized as follows: Section 2 presents a state of the art on DES. LCA as a measurement tool in the context of DES is described in Section 3. Section 4 presents a case study using an automotive paint shop facility example to demonstrate how DES in combination with LCA can be used. Section 5 provides discussions and conclusions as to how the presented case study can be generalized and used for decision support and requirements specification for a sustainable manufacturing systems design.

2. DISCRETE EVENT SIMULATION

Simulation has been demonstrated to be a very effective approach for problem solving and optimizing manufacturing systems design. One of the primary application areas for modeling and simulation is manufacturing system, according to Law and McComas [1]. However, analysis and optimization of multiple objectives is not very common in manufacturing simulation. Detailed discussion of modeling and simulation can be found in numerous books, among the best known are Banks et al. [2], and Law and Kelton [3]. The technology of utilizing DES has been rapidly evolving, hundreds of academic publications and new software features are released every year. DES software and languages have been used for numerous purposes, such as patient flows in healthcare, military strategies, logistics, call centers, restaurants, etc. One of the most frequently stated objectives in DES is profit optimization, i.e., analyzing which of the alternative solutions is the most profitable over time. There are many other criteria, which one could measure with DES. In the past, the emphasis has been mainly on profitability. However, environmental considerations are becoming more relevant and require greater attention as long as humans continue to utilize natural resources. DES and LCA is one possible combination for analyzing the cause and effect of various scenarios where time, resources, place, and randomness of input variables affect the outcome in sustainable manufacturing design. This analysis is an unexplored area; only a few research publications exist. The few examples include: Solding and Petku [4] and Solding and Thollander [5] both describe how DES could be utilized to reduce electricity consumption for foundries. Östergren et al. [6] and Johansson et al. [7] describe how DES could be utilized in combination with LCA for decreasing environmental impacts during food production.

3. LIFE CYCLE ASSESSMENT FOR DISCRETE EVENT SIMULATION

LCA is a methodology for evaluating the environmental impact associated with a product during its life cycle. LCA can be accomplished by identifying and quantitatively describing a product's requirements for energy and materials, and the emissions and waste released to the environment. A product under study is followed from the initial extraction and processing of raw materials through manufacturing, distribution, and use, to final disposal, including the transports involved, i.e., its entire lifecycle. LCA is an ISO standardized tool [8-10].

Using LCA data in a DES model is a novel multidisciplinary technique, which enables environmental impact evaluations of the manufacturing system performance. To the best of our knowledge, only three models of real world systems have been built so far, which utilizes LCA data in a DES model. We discuss one such system in the paper. The other systems were developed for simulating a factory which produces sausages [6, 11, and a dairy, which produces cultured dairy products [12].

4. CASE STUDY

To demonstrate a manufacturing planning scenario with an emphasis on sustainability a simulation model has been built based on the work flow schematic as shown in Figure 1. This scenario presents a paint shop with six painting steps to set the scene for requirements specification in an automotive paint shop.

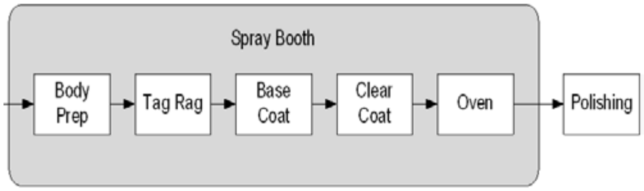


Figure 1. Example of paint shop processes [13]

Figure 1 shows six steps (Body Preparation, Tag Rag, Base Coat, Clear Coat, Oven and Polishing) incorporated in the simulation model. The model was created based on some earlier work [14-17] as seen in Figure 2.

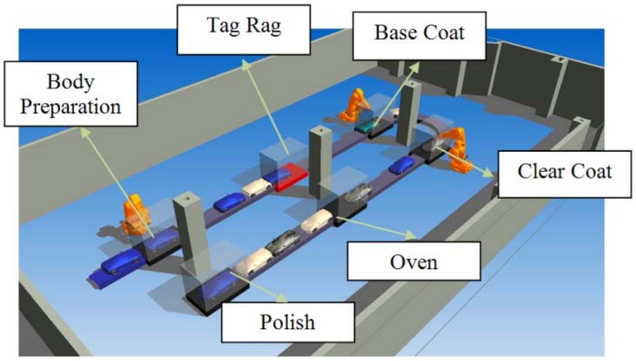


Figure 2. 3D-representation of the paint shop test model

Table 1. Default settings for resources in the paint shop

Resource	Body Prep	Tag Rag	Base Coat	Clear Coat	Oven	Polish
Processing Times						
Cycle time (Normal distribution)	n	n	n	n	n	n
mean (Seconds)	120	130	140	130	240	125
Standard deviation	2	4	1	3	2	1
Energy (kW)						
Down	1	1	1	1	1	1
Idle	5	4	50	50	1800	50
Busy	20	18	500	500	1800	200
Failures						
MTTF (Uniform distribution)	u	u	u	u	u	u
Min (Seconds)	1000	1200	1000	900	1000	900
Max (Seconds)	5000	5200	11000	10900	15000	4900
MTTR (Normal distribution)	n	n	n	n	n	n
Mean (Seconds)	240	260	600	590	1000	240
Standard deviation	2	3	2	3	2	3

4.1 Input data

Each production step has a setting for the resource to be down, idle, or busy. Down means disconnected from the power provider, i.e., no electricity is used. Idle means that the resource is on standby, i.e., some electricity is used. Busy means doing the work cycle as such, i.e., electricity is used. Table 1 shows the input data specifying the energy use from the default settings in the paint shop model, as well as other data needed for setting parameters at the resources of the model such as cycle times, MTTF (Mean Time To Failure), MTTR (Mean Time To Repair), etc.

The data herein presented are for the purposes of demonstration of our scenario and do not necessarily imply an actual paint shop data.

4.2 Problem description

When designing a new manufacturing system certain production goals and economic measures need to be fulfilled. For example the production capacity is specified to be at least a certain level, the cost of the manufacturing system needs to be within the budget, and the environmental impact is expected to be below a certain guideline value.

4.3 Goal

In this case study, the goals of the sustainable manufacturing system are assumed as follows: 1. to reach a production capacity of at least 50000 cars per year, 2. there will be no more than 500 metric tons of CO_2 emission per year, and 3. no new investment in equipment for the existing factory. The current factory is represented by the input data in Table 1, as well as the output data from Trial run 1 in Table 2.

4.4 Experiments

In this case study, the number of input variables are simplified to only a few choices as shown in Table 2. In a real world application however, a variety of designed operating parameters are considered based on the required system throughput. In the experiments, the number of input data parameters can be varied more extensively and practically anything feasible for a real world change could be varied if necessary to bring forth sound requirements specification for the considered manufacturing system.

From the initial settings (Trial run 1 in Table 2), the oven had been identified to be the bottleneck in terms of utilization as well as energy consumption. Some trial runs were performed based on different parameter settings. The settings included the energy source, oven cycle time, and energy consumption as well as a single or two ovens in parallel. The energy sources in the parameter setting included wind, water, or a mix of energy sources depending on the country where the factory is located.

The primary purpose of this simulation is to provide requirements specification support data, and hence also provide support towards designing a sustainable paint shop. In line with this effort, some examples of measures are provided in terms of energy, throughput and CO_2 based on the simulation runs. In Table 2, from the twelve trial runs one can identify the bottlenecks, energy consumption and CO_2 emissions due to energy type used in the paint shop. The results presented in Table 2 are calculated by running the simulation model. The model incorporates lifecycle assessment data from an European Union LCA database as described in Heilala et al. [14].

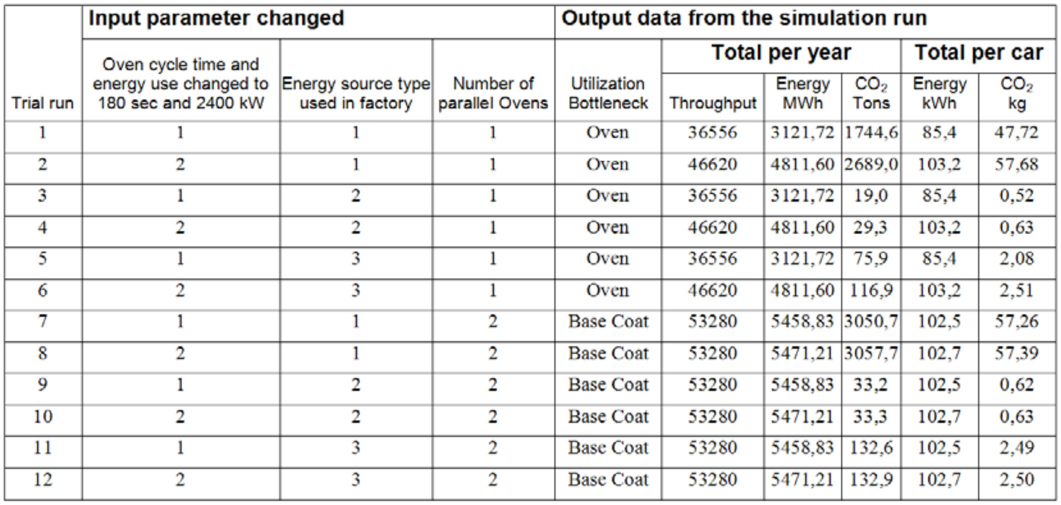
4.5 Results

Following are examples of conclusions arrived from looking at Table 2:

- The initial setting gives the lowest energy consumption per produced car, as well as trial 3 and 5
- The Oven is the throughput bottleneck initially (trial 1)
- Decreasing cycle time for the oven with 60 seconds does increase output of cars; however Oven is still the bottleneck.
- By adding another parallel oven, the Base Coat will be the bottleneck.
- Wind powered paint shop gives the lowest CO_2 emissions (from energy) per car produced.

Note that these conclusions are not the only items to consider, however they give more information needed and provide for a better decision space that a normal non-discrete event simulation analysis does.

Table 2. An example result of twelve simulation runs



Trial run	Input parameter changed			Output data from the simulation run					
					Total per year			Total per car	
	Oven cycle time and energy use changed to 180 sec and 2400 kW	Energy source type used in factory	Number of parallel Ovens	Utilization Bottleneck	Throughput	Energy MWh	CO_2 Tons	Energy kWh	CO_2 kg
1	1	1	1	Oven	36556	3121,72	1744,6	85,4	47,72
2	2	1	1	Oven	46620	4811,60	2689,0	103,2	57,68
3	1	2	1	Oven	36556	3121,72	19,0	85,4	0,52
4	2	2	1	Oven	46620	4811,60	29,3	103,2	0,63
5	1	3	1	Oven	36556	3121,72	75,9	85,4	2,08
6	2	3	1	Oven	46620	4811,60	116,9	103,2	2,51
7	1	1	2	Base Coat	53280	5458,83	3050,7	102,5	57,26
8	2	1	2	Base Coat	53280	5471,21	3057,7	102,7	57,39
9	1	2	2	Base Coat	53280	5458,83	33,2	102,5	0,62
10	2	2	2	Base Coat	53280	5471,21	33,3	102,7	0,63
11	1	3	2	Base Coat	53280	5458,83	132,6	102,5	2,49
12	2	3	2	Base Coat	53280	5471,21	132,9	102,7	2,50

The left side of Table 2 shows the input data which is varied for the twelve runs. Column one on "Input parameter changed" can be set to either 1 for normal conditions or 2 for 180 sec cycle time and 2400kWh. Column two shows which type of energy is used, 1 for an average country energy (i.e. mixed sources), 2 for wind power, 3 for water power. Column three shows the number of parallel ovens used in the model.

4.6 Discussions

The study results and output data are shown in Table 2. Constraints from the stated goals of the study have to be considered while analyzing the study results. To satisfy the goal to produce at least 50000 cars per year, Table 2 output data shows that trial runs 7-12 are feasible, however an investment in another oven will need to be added to the process. The next goal is to decrease the CO_2 emissions to less than 500 metric tons per year. To reach this goal, standard fossil fuel energy cannot be used. Alternatively wind or water powered energy will need to be used. Table 2 shows trial runs 9-12 as feasible solutions with the use of "green" energy alternatives. In order to minimize the investment goal, the cycle time and energy consumption of the oven does not need to be changed. This means trial run 9 or 11 will be the preferred choice, depending on the energy cost from the power provider. It may be worthwhile to notice that the wind power could be a better choice than the water powered energy alternative in terms of CO_2 emissions.

5 CONCLUSION

The study demonstrated that using the environmental measures from a LCA database and traditional input data with cycle time, disturbance data, etc. for discrete event simulation, new output measures from the model can be used to identify and analyze sustainable manufacturing system design and measures such as energy consumption at the aggregated shop floor level, resource level, and production throughput. Such analysis can also be useful in identifying the bottlenecks on any environmental measure; in this case the energy consumption and carbon footprint in relation to energy source used.

The software used for building and evaluating this model was developed under the SIMTER project as described in Heilala et al. [14], Lind et al. [15], Lind et al. [16] and Johansson et al. [17]. To our knowledge, this software solution is the first effort on combining lifecycle assessment data directly into the discrete event simulation engine.

6 FUTURE RESEARCH

Based on the described case study it would be desirable to be able to represent sustainability related data in a neutral format. One possible solution is to store and use sustainability and other related data for discrete event simulation through the CMSD (Core Manufacturing Simulation Data) specification [19], developed under Simulation Interoperability Standards Organization (SISO) [18, 19]. This will allow us to maintain neutral and accessible measures for sustainability data.

7 ACKNOWLEDGMENTS

We acknowledge the funding from VINNOVA (Swedish Agency for Innovation Systems, integrates research and development in technology, transport and working life.).

8 DISCLAIMER

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied.

9 REFERENCES

[1] Law, A. M., McComas, M. G. 1999. Simulation of Manufacturing Systems. In Proceedings of the 1999 Winter Simulation Conference, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, Pheonix, AZ. pp. 56–59.

[2] Banks, J., Carson, J. S., Nelson, B. L., Nicol, D. M. 2000. Discrete-event system simulation. 4th ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

[3] Law, A. M., Kelton. W. D. 2000. Simulation modeling & analysis. 3rd ed. New York: McGraw-Hill, Inc.

[4] Solding, P., Petku. D. 2005. Applying Energy Aspects on Simulation of Energy-Intensive Production Systems. In Proceedings of the 2005 Winter Simulation Conference, ed. Kuhl, M. E., Steiger, N.M., Armstrong, F. B., Joines, J. A. Orlando, FL, USA

[5] Solding, P., Thollander. P. 2006. Increased Energy Efficiency in a Swedish Iron Foundry Through Use of Discrete Event Simulation. In Proceedings of the 2006 Winter Simulation Conference, ed. Perrone, L. F., Wieland, F. P., Liu J., Lawson, B. G., Nicol, D. M., Fujimoto, R. M. Monterey, CA, USA.

[6] Östergren, K., Berlin, J., Johansson, B., Stahre, J., Tillman, A.M. 2007. A tool for productive and environmentally efficient food production management. European conference of Chemical Engineering, Copenhagen, 16-20 September 2007.

[7] Johansson, B., Stahre, J., Berlin, J., Östergren, K., Sundström, B., Tillman, A.M. 2008. Discrete Event Simulation with Lifecycle Assessment data at a Juice Manufacturing System. In proceedings of the 5th FOODSIM Conference, University College Dublin, Ireland.

[8] ISO. 1997. Environmental Management – Life Cycle Assessment – Principles and Framework. ISO 14040:1997. European Committee for Standardization CEN, Brussels, Belgium.

[9] ISO. 1998. Environmental Management – Life Cycle Assessment – Goal and Scope Definition and Inventory Analysis. ISO 14041:1998. European Committee for Standardization CEN, Brussels, Belgium.

[10] ISO. 2000. Environmental Management – Life Cycle Assessment – Life Cycle Interpretation. ISO 14043:2000. European Committee for Standardization CEN, Brussels, Belgium.

[11] Johansson, C., Ingvarsson, A., 2006. Flow simulation of food production; Ingemar Johansson i Sverige AB, MSc Thesis (In Swedish), Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

[12] Alvemark, O., Persson, F. 2007. Flow simulation of food production; cultured dairy products, MSc Thesis (In Swedish), Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

[13] Leng, C. F., Yingchao G. 2005. Production Performance Improvement Using DES of Low Volume Production in a Paint Shop, MSc Thesis, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

[14] Heilala, J., Saija, V., Tonteri, H., Montonen, J., Johansson, B., Stahre, J., Lind, S. 2008. Simulation-Based Sustainable Manufacturing System Design, In Proceedings of the 2008 Winter Simulation Conference, eds. Mason, S. J., Hill, R. R., Mönch, L., Rose, O., Jefferson, T., Fowler, J. W. 1922–1930, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[15] Lind, S., Krassi, B., Viitaniemi, J., Kiviranta, S., Heilala, J., Berlin, C. 2008. Linking Ergonomics Simulation to Production Process Development, In Proceedings of the 2008 Winter Simulation Conference, eds. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1968–1973, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[16] Lind, S., Krassi, B., Johansson, J., Viitaniemi, J., Heilala, J., Stahre, S., Vatanen, S., Fasth, Å., Berlin C., 2008. SIMTER: A Production Simulation Tool for Joint Assessment of Ergonomics, Level of Automation and Environmental Impacts. The 18th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2008), June 30 – July 2, 2008.

[17] Johansson, B., Fasth, Å., Stahre, J., Heilala, J., Leong, S., Lee, Y. T., Riddick, F. 2009. Enabling Flexible Manufacturing Systems by Using Level of Automation as Design Parameter. In Proceedings of the 2009 Winter Simulation Conference, eds. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[18] SISO 2009. Standard for Core Manufacturing Simulation Data Information Model: UML model (Draft) CMSD Product Development Group, Simulation Interoperability Standards Organization. http://www.sisostds.org/index.php?tg=fileman&idx=get&id=49&gr=Y&path=Papers+and+Presentations&file=06F-SIW-028_final.pdf [Accessed April 13, 2009].

[19] Simulation Interoperability Standards Organization (SISO) Executive Committee. 2008. SISO Polices and Procedures, SISO-ADM-002-2008.

Towards a New Geometric Metric for Sustainability Assessment

Gaurav Ameta
School of Mechanical and Materials
Engineering
Washington State University
Pullman, WA, 99164
+1-509-335-8218

gameta@wsu.edu

ABSTRACT

This paper puts forth a novel geometric metric for assessing sustainability of a product. The purpose of the novel geometric metric is to present sustainability aspects of a product, for its entire life-cycle (material production, manufacturing, supply-chain, use and disposal), to design engineers in a readily comprehensible way. Achieving sustainability is critical for minimizing the detrimental effects caused by global warming due to modern equipment manufacturing, use and disposal. Impact of various products on the environment, society and economy are primarily locked in the design stage of the product. Therefore engineers need a comprehensive metric to be used at the design stage for designing a sustainable product. The paper first reviews sustainability metrics and then focuses on the geometric metrics for evaluating sustainability aspects of a product. Based on the concept of areal coordinates, this paper constructs a preliminary geometric metric for sustainability assessment.

Keywords

Sustainability, Geometric metric, Product life-cycle.

1. INTRODUCTION

Due to the current climate change scenario, the notion of sustainability has recently gained wide interest. According to the United Nations Environment Program, climate change is affected by various human activities such as land use changes and fossil fuel burning [1]. Although Sustainability is a common objective of all entities over the world, its realization is difficult as it is engulfed in myriad of political, societal, regional, technological, economical, legal and geological issues. It is also quite evident that sustainable development is a dynamic process by nature [2, 3], as the biosphere and conditions around the world are ever changing and still quite unpredictable. Despite this unpredictability, scientist, governments, industry, consumers etc., have realized that increase in global temperatures is very likely due to the increase in anthropogenic (human) greenhouse gas concentrations. This increase in global temperatures, if not curbed, will have a debilitating effect on the viability of the biosphere to sustain life [4]. To impede and hopefully reverse the debilitating climate changes that have occurred, sustainable products should be designed.

A sustainable product can be defined by understanding the meaning of sustainability or sustainable. The word "sustainable" was first used with respect to its current usage as sustainable development. Sustainable development is the development that "meets the needs of the present without compromising the ability of future generations to meet their own needs." [5]. A definition of sustainability according to the US National Research Council is "the level of human consumption and activity, which can continue into the foreseeable future, so that the systems that provides goods and services to the humans, persists indefinitely" [6]. Other authors (e.g., Stavins *et al.* [7]) have argued that any definition of sustainability should include dynamic efficiency, should consist of total welfare (accounting for intergenerational equity) and should represent consumption of market and non-market goods and services.

In this paper, sustainable product would imply that the product is sustainable in all aspects (society, economy and environment) throughout its entire life-cycle from material production to manufacturing, supply chain, use and disposal of the product.

This paper puts forth preliminary work for a novel geometric metric for evaluating sustainability of a product. The next section discusses product life-cycle and interaction of sustainability aspects. Section 3 summarizes life cycle assessment method. Then, section 4 discusses various metrics and related efforts for sustainability assessment. Section 5 puts forth the notion of the novel geometric metric.

2. LIFE CYCLE ASSESSMENT

Life cycle assessment (LCA), have been developed by International Organization for Standardization (ISO) [8], for assessing the environmental impacts of products. By including the impacts throughout the product life cycle, LCA provides a comprehensive view of the environmental aspects of the product or process. LCA has been widely popular for identifying environmental impact of a product or process. LCA methodology has been incorporated into several commercial (SimaPro [9] and GaBi [10]), governmental (TRACI [11], BEES [12]) and academic environmental assessment tools (EioLCA [13], and

Permis'09, September 21–23, 2009, Gaithersburg, MD, USA.

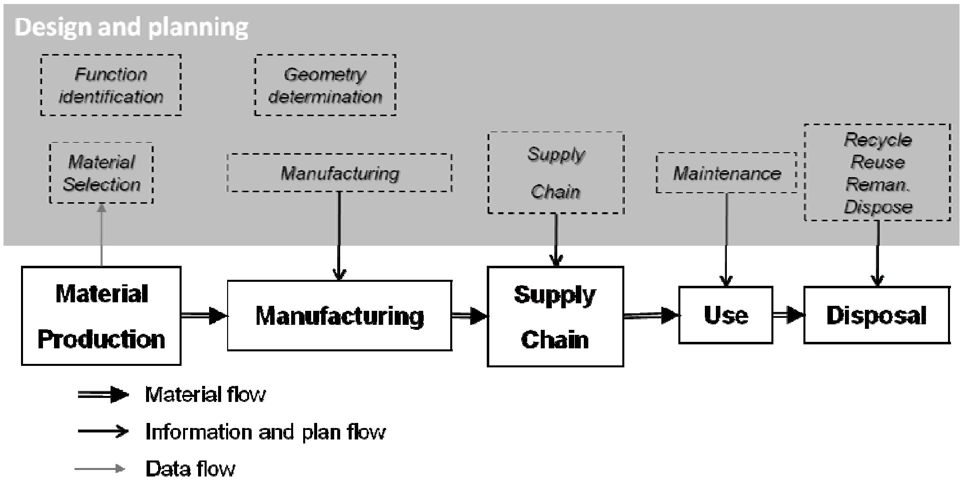


Figure 1: Different stages of a product life-cycle, including the material handling stages and the planning stage (design).

EcologiCAD [14]) One of the important aspects of LCA is that it is able to present environmental and economic impacts in an aggregated manner.

Despite the large application of LCA, it has been attributed with some drawbacks related to (a) System Boundaries (b) Data Issues and (c) Methodology Issues such as (Weighing methods, Aggregation methods and Comparison across indices)[2, 15-20]. The methodology issues are related to the selection of appropriate metric for comparing products.

3. PRODUCT LIFE-CYCLE AND SUSTAINABILITY

As shown in Figure 1, there are several stages in a product life-cycle. At the design and planning stage, function identification, geometry and material optimization, overall manufacturing, supply chain, and disposal planning is conducted. Then the design documentations are passed to the manufacturing stage.

Manufacturing, supply chain, and disposal of the product are further planned (refined) before actual handling of material. Therefore, there is a need of sustainability related decision support at different levels; at the design stage and at the individual manufacturing, supply chain and disposal stages. The sustainability impact of the entire product life-cycle can be estimated with the following equation

$$S = S_{matprod} + S_{manu} + S_{supply} + S_{use} + S_{disp} \qquad (1)$$

where, $S_{matprod}$ represents sustainability impacts from the material production, S_{manu} represents sustainability impacts for manufacturing, etc.

Each of the sustainability terms in equation (1) can be further sub-divided based on its influence one or a combination of aspects from society, economy or environment.

4. METRICS FOR SUSTAINABILITY ASSESSMENT

In this section we will discuss various issues related to sustainability and environmental metrics and some examples of metrics studied in the literature.

4.1 Metrics Classification

Scoping sustainability and defining clear system boundaries are critical for properly defining metrics for sustainability assessment [21]. Various metrics developed so far to measure the progress towards sustainability have been classified by Mayer [2] and Jain [22] into: a) indicators, b) indices and c) frameworks. In a recent article by Sikdar, indicators were identified as 1-D metric as they would quantify changes in only one of the bottom lines of sustainability [23]. Indices could be a 2-D metric or 3-D metric, in a sense that they could quantify changes in either two or three of the bottom lines of sustainability.

4.1.1 Indicators

Indicators basically measure a single parameter of a system, e.g., CO_2 emission or energy use. A detailed survey of indicators has been presented in Patlitzianas *et al.* [24]. Keffer *et al.* propose a framework for developing a classification of indicators [25]. In the framework, indicators are classified based on aspects and categories. Categories are broad areas of influence related to environment, economy and society, referred to as the triple bottom line of sustainability. Aspects are defined as general type of data that is related to a specific category.

4.1.2 Indices

Indices are basically aggregates of several indicators, e.g., Ecological Footprint [26] (a ratio of the amount of land and water required to sustain a population to the available land and water for the population) or Environmental Vulnerability Index (consists of indicators of hazards, resistance and damage). Indices represent a single score by combining various indicators of different aspects of a system.

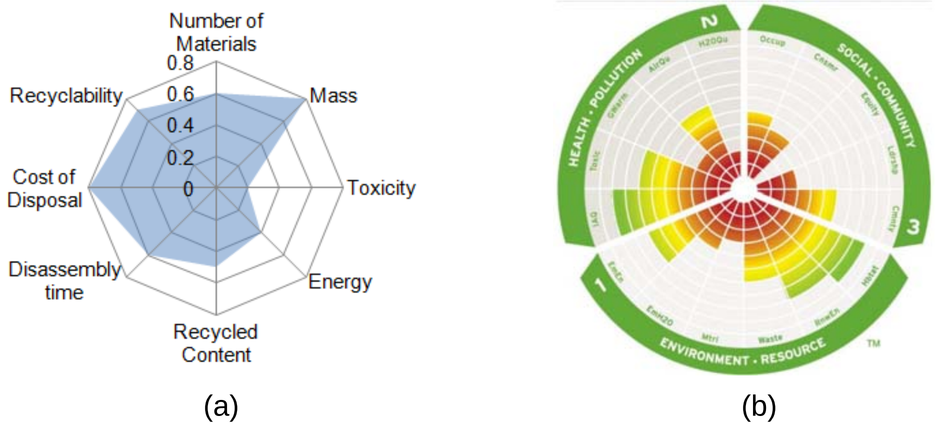


Figure 2: (a) Spider Chart or Radar Graph, (b) Modified Spider Chart as demonstrated by the pharos project [39].

Key requirements and rigorous mathematical requirements for sustainability indices are proposed in Bohringer and Jochem [27] and Ebert and Welsch [28], respectively. The strengths and weakness of several sustainability indices are compared by Mayer [2].

4.1.3 Frameworks

Frameworks present large numbers of indicators in qualitative ways, e.g., the vulnerability framework [29] or the CRITINC Framework [30]. Frameworks do not aggregate data in any manner. An advantage of frameworks is that the values of all indicators can be easily observed and are not hidden behind an aggregated index. The disadvantage of using frameworks is that they are hard to compare over time although this is possible by using Hasse diagrams [31]. A brief review of sustainability frameworks is provided by Mayer [2].

4.2 Examples

In this section we will survey metrics and their categories that have been used to evaluate products from manufacturing enterprises.

4.2.1 Sustainability Metric

Sustainability metric should include impacts from not just environmental aspects but economic and societal aspects of the entire product life-cycle. In this regard, Datschefski proposed that the sustainability of a product should be measured using recyclability, safety, efficiency, use of renewable energy and social effects [32].

4.2.2 Environmental impacts Metric

A review of eco-indicators used in product development is provided by [33]. Environmental impacts metric have been classified into quantitative (Material Intensity per Service Unit (MIPS), Cumulative Energy Demand (CED), and EcoIndicator 95 (EI95)) and qualitative metrics in [34, 35]. As quantitative methods, were analyzed.

4.2.3 Geometric Representations

The sustainability metric can be represented as a single aggregated number or represented geometrically. The geometric representations can be in the form of (a) multidimensional geometric representation (b) spider chart [36] also known as radar graph [37], ternary plot, ternary graph, triangle plot, simplex plot, or de Finetti diagram (c) simplified graphs and tables and (d) annotated /colored CAD or PLM product design [38]. Figure 2 demonstrates two different forms of spider chart.

5. GEOMETRIC METRIC

The purpose of creating the geometric metric is to provide the sustainability related information to engineers in a graphical manner. The geometric metric will be constructed from a basis–simplex and described with areal coordinates, a method from affine geometry [40].

5.1 Areal Coordinates

For a geometric metric of three-dimensions, four (3+1) basis-points ($\sigma_1..\sigma_4$) positioned arbitrarily will be utilized to create the three-dimensional space. (It is only necessary for affine geometry that the positions for basis-points $\sigma_1..\sigma_4$ be independent.) A geometric metric for sustainability of three-dimensions would consider three sustainability aspects that can be converted / normalized to have a single unit.

At the four basis-points place four masses λ_1, λ_2, λ_3, and λ_4 that may be positive or negative. So long as $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 \neq 0$, the position of σ , the centroid of these masses, is uniquely determined by the linear combination

$$(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)\sigma = \lambda_1\sigma_1 + \lambda_2\sigma_2 + \lambda_3\sigma_3 + \lambda_4\sigma_4 \qquad (2)$$

and σ can assume any position in the space of $\sigma_1..\sigma_4$ by varying $\lambda_1..\lambda_4$; e.g. for $\lambda_1..\lambda_4$ all positive, σ identifies any point inside tetrahedron $\sigma_1\sigma_2\sigma_3\sigma_4$. The four masses $\lambda_1..\lambda_4$ are the barycentric coordinates of σ, yet, note that the position of σ depends only on three independent ratios of these magnitudes. Consequently, the four λ_i can be normalized by setting $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$. Then, they are areal coordinates and

$$\sigma = \lambda_1\sigma_1 + \lambda_2\sigma_2 + \lambda_3\sigma_3 + \lambda_4\sigma_4 \qquad (3)$$

In three-dimensional space, the basis-simplex is a tetrahedron. In n-space the tetrahedron for dimension 3 generalizes to the n-simplex. The simplex occurs in spaces of all dimensions: in dimension zero, it is a point; in dimension 1, a line-segment; in 2, a triangle; in 3, a tetrahedron; in 4, a four-simplex; etc. Non-regular simplexes occur in all spaces of dimension 2 and higher;

they are formed simply by spacing some or all of the vertices at different distances from each other.

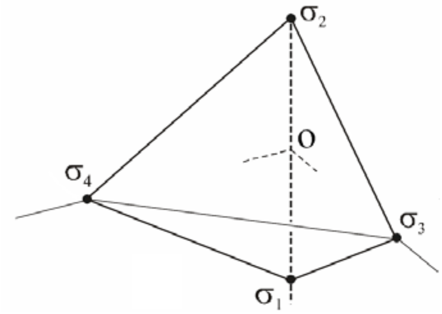


Figure 3: Basis-simplex used to create a three-dimensional geometric metric.

5.2 Metric Use

A sample basis-simplex and axes mapped on the simplex is shown in Figure 3 and Figure 4, respectively. The metric also shows emissions, carbon weight [41] and global warming potential [42] as axes mapped on the simplex. All of these impacts can be converted into the units of GWP (Global Warming Potential).

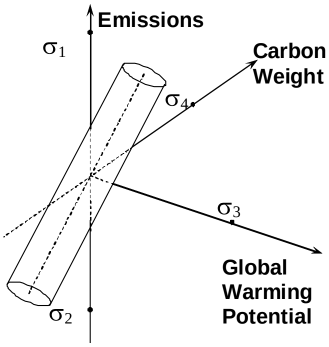


Figure 4: An example of a geometric metric constructed using the basis-simplex in Fig 3. The metric also shows Emissions, Carbon Weight and Global Warming Potential as axes mapped on the simplex.

Based on the inherent inter-relations between emission, carbon weight and GWP and the effect of a particular product life-cycle, a designer may want to specify an allowable limit for the particular stages of the life-cycle of the product. These limits will take some shape geometrically in the hypothetical space of the geometric metric. One such sample shape (the shape is shown just as an example and does not represent the real inter-relation between these three axes) is shown in Figure 4 as a cylinder. Similar shapes from different stages of the product may be combined together (equation (1)) to obtain the possible total impact for the entire life-cycle of the product.

The combination of these geometric shapes can be obtained through a process known as Minkowski Sum, which is defined as the vector sum of points on geometric shapes. Minkowski Sum has applications in image processing, robotics [44,45], CAD [46, 47], spatial planning, graphic arts, animation [48,49] and tolerance analysis [50, 51].

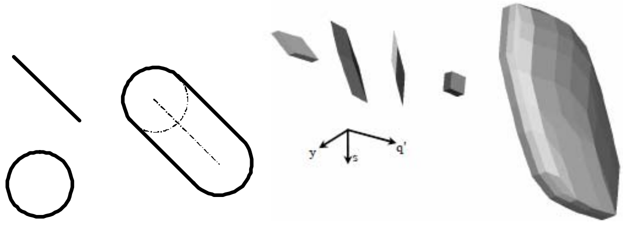


Figure 5 (a) Minkowski sum of two shapes; 1-dimensional line and 2-dimensional circle and (b) Minkowski sum of four 3-dimensional convex polyhedral (figure from [50]).

5.3 Example Metric

Let us consider a simplified case for constructing a simple metric. Consider the electricity consumption and carbon weight at the manufacturing stage. Data is available in [43] relating electricity consumption and carbon emission in different states across United States of America. If energy (electricity) and carbon weight are considered as two impacts from the manufacturing stage of a product, then the set of lines shown in Figure 6 can be obtained.

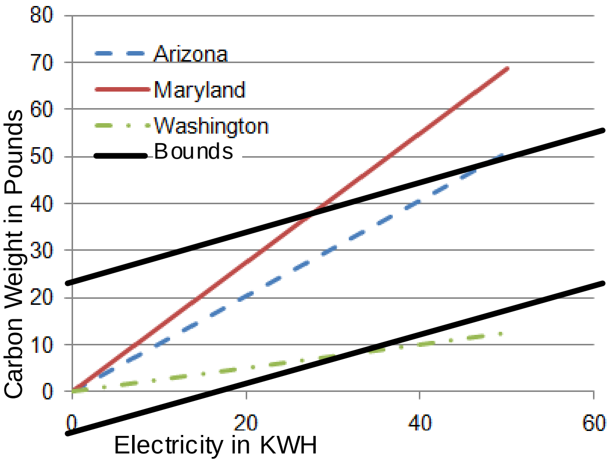


Figure 6: Set of lines as a geometric metric for representing electricity use and carbon weight.

As is quite evident from the allowable bounds set by the designer that for the same amount of electricity consumption there will be higher amount of carbon weight associated. Currently, the shape/slope of the bounding lines is selected based on the average slope of the lines available for all the states.

6. FUTURE WORK

Although a very preliminary study is presented in this paper for developing the geometric metric for assessing sustainability, the geometric metric can be easily extended to include many more aspects of sustainability. In future the author would build different manifold geometries in n-dimensional space, will be obtained such as curves, open surfaces, closed surfaces, hyper-curves and hyper-surfaces based on n-impacts selected from a particular stage of the life-cycle. Since, not all sustainability aspects can be converted to have same units; a set of these geometric metrics would be created. More over each stage of the product life-cycle will have a complete set of these metrics. The aggregation of these sustainability impacts from different stages of the life-cycle would be accomplished using minkowski sum of the respective geometric metric.

7. REFERENCES

[1] Rekacewicz, P. 2005. Climate change: processes, characteristics and threats. UNEP/Grid-Arendal Maps and Graphics Library.

[2] Mayer A.L. 2008. Strengths and weakness of common sustainability indices for multidimensional systems. Environment International. 34(2), 277-291.

[3] Mog J. M. 2008. Struggling With Sustainability – A Comparative Framework For Evaluating Sustainable Development Programs. World Development. 32(12), 2139-2160.

[4] http://ipcc-wg1.ucar.edu/wg1/wg1_home.html

[5] Our Common Future, 1987 (Oxford University Press).

[6] Our Common Journey: A Transition Toward Sustainability, 1999 (National Academy Press).

[7] Stavins R.N., Wagner A.F., and Wagner G. 2003. Interpreting sustainability in Economic Terms: Dynamic Efficiency Plus Intergenerational Equity. Economic Letters. 79(3), 339-343.

[8] ISO 14001:2004, Environmental management systems -- Requirements with guidance for use, ISO, International Organization for Standardization (ISO), Geneva, Switzerland, 2004.

[9] http://www.pre.nl/simapro/default.htm

[10] http://www.gabi-software.com/

[11] http://www.epa.gov/nrmrl/std/sab/traci/

[12] http://www.bfrl.nist.gov/oae/software/bees/

[13] http://www.eiolca.net/

[14] http://www.leibrecht.org/

[15] Udo de Haes HA. 1993. Applications of life cycle assessment: expectations, drawbacks and perspectives. Journal of Cleaner Production. 1(3–4), 131–37

[16] Coatanéa E., Kuuva M, Makkonen P. E. and Saarelainen T. 2007. Environmental analysis of the Product Life Cycle by using an aggregated metric based on exergy. International Journal of Product Lifecycle Management. 2(4), 337-355.

[17] Reap J., Roman F., Duncan S. and Bras B. 2008. A survey of unresolved problems in life cycle assessment; Part 1: goal and scope and inventory analysis. International Journal of Life Cycle Assessment. 13, 290-300.

[18] Reap J., Roman F., Duncan S. and Bras B. 2008. A survey of unresolved problems in life cycle assessment; Part 2: impact assessment and interpretation. International Journal of Life Cycle Assessment. 13, 374-388.

[19] Lenzen, M. 2001. Errors in Conventional and Input-Output-based Life-Cycle Inventories. Journal of Industrial Ecology. 4(4), 127-148.

[20] Strømman, A. H., Peters, G. P., and Hertwich, E. G. 2009. Approaches to correct for double counting in tiered hybrid life cycle inventories. Journal of Cleaner Production. 17(2), 248-254.

[21] Sustainability risks being about everything and therefore, in the end, about nothing, The Economist, 2002.

[22] Jain, R. 2005. Sustainability: metrics, specific indicators and preference index. Clean Technologies and Environmental Policy. 7, 71-72.

[23] Sikdar S.K. 2008. Sustainable Development and Sustainability Metrics. AIChE Journal. 49(8), 1928-1932.

[24] Patlitzianas, K. D., Doukas, H., Kagiannas, A. G., and Psarras, J. 2008. Sustainable energy policy indicators: Review and recommendations. Renewable Energy. 33(5), 966-973.

[25] Keffer, C., Shimp, R., and Lehni, M. 1999. Eco-efficiency Indicators & Reporting, Report on the Status of the Project's Work in Progress and Guideline for Pilot Application, WBCSD, Working Group on Eco-efficiency Metrics and Reporting, Geneva, Switzerland.

[26] Wackernagel M, Rees W.E. 1998. Our Ecological footprint: reducing human impact on earth. New Society Publishers. Gabriola Island, BC, Canada.

[27] Bohringer C.and Jochem P.E.P. 2007. Measuring the immeasurable - A Survey of sustainability indices. Ecological Economics. 63, 1-8.

[28] Ebert U. and Welsch H. 2004. Meaningful Environmental Indices: A Social Choice Approach. Journal of Environmental Economics and Management. 47, 270-283.

[29] Turner, B. I., Kasperson, R. E., Matson, P. A., McCarthy, J. J., Corell, R. W., and Christensen, L. 2003. A framework for vulnerability analysis in sustainability science. Proceedings of the National Academy of Sciences of the United States of America. 100(14), 8074-8083.

[30] Ekins, P., Simon, S., Deutsch, L., Folke, C., and De Groot, R. 2003. A framework for the practical application of the concepts of critical natural capital and strong sustainability, Ecological Economics. 44(2-3), 165-185.

[31] Patil G.P. and Taillie C. 2008. Multiple indicators, partially ordered sets, and linear extensions: multi-criterion ranking and prioritization. Environmental and Ecological Statistics. 11, 199-228.

[32] Datschefski E. 2001. The Total Beauty Of Sustainable Products. Rotovision SA, Switzerland.

[33] Perrson J.G., 2001. Eco-indicators in product development. Proc Instn Mech Engrs. 215(Part B), 627-635.

[34] Ernzer M. and Wimmer W. 2002. From environmental assessment results to Design for Environment product changes: an evaluation of quantitative and qualitative methods. Journal of Engineering Design. 13(3), 233–242

[35] Lenau T. and Bey N. 2001. Design of environmentally friendly products using indicators. Proc Instn Mech Engrs. 215(B), 637-645.

[36] Rogers C. B. 1995. The Spider Chart: A Unique Tool for Performance Appraisal, Annual Quality Congress Proceedings-American Society For Quality Control, pp. 16-22.

[37] Bothe D. R., QICID: 18844 Title: Column: One Good Idea: The Circular Radar Graph.

[38] Lifecyclewiki - Case Study. - http://www.lifecyclewiki.net/CaseStudy

[39] Pharos Project Home. - http://www.pharoslens.net/

[40] Coxeter H. S. M. 1961. Introduction to geometry. New York.

[41] Wiedmann T., and Minx J. 2008. A definition of carbon footprint. In Ecological Economics Research Trends. Editor CC Pertsova, 1, pp. 1-11.

[42] Global Warming Potentials - http://unfccc.int/ghg_data/items/3825.php.

[43] Updated State-level Greenhouse Gas Emission Coefficients for Electricity Generation 1998-2000. 2002. Office of Integrated Analysis and Forecasting Energy Information Administration U.S. Department of Energy.

[44] Lengyel J., Reichert M., Donald B. R., and Greenberg D. P. 1990. Real-time robot motion planning using rasterizing computer graphics hardware. Proceedings of the 17th annual conference on Computer graphics and interactive techniques. pp. 327–335.

[45] Borst C., Fischer M., and Hirzinger G., 1999. A fast and robust grasp planner for arbitrary 3D objects. IEEE International Conference on Robotics and Automation.

[46] Peternell M., and Steiner T. 2007. Minkowski sum boundary surfaces of 3D-objects. Graphical Models. 69(3-4), pp. 180–190.

[47] Ghosh P. K. 1991. Vision, geometry, and Minkowski operators. Vision geometry: proceedings of an AMS special session held October 20-21, p. 63.

[48] Rossignac J. and Kaul A. 1994. AGRELs and BIPs: Metamorphosis as a Bezier curve in the space of polyhedra. In Proc Eurographics. Oslo, Norway Computer Graphics forum. 13(3), pp C179-C184.

[49] Kaul A. and Rossignac J. 1991. Solid-interpolating deformations: construction and animation of PIPs. In Proc. Eurographics. pp. 493-505.

[50] Ameta G., Davidson J. K., and Shah J. J. 2007. Tolerance-Maps Applied to a Point-Line Cluster of Features. Journal of Mechanical Design. 129. 782-793.

[51] Ameta G. 2006. Statistical tolerance analysis and allocation for assemblies using Tolerance-Maps. Doctoral Dissertation. Arizona State University.

A Mission Taxonomy-Based Approach to Planetary Rover Cost-Reliability Tradeoffs

David Asikin
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA, USA
dasikin@cs.cmu.edu

John M. Dolan
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA, USA
jmd@cs.cmu.edu

ABSTRACT

Our earlier work on robot mission reliability provides tradeoff analysis between input parameters such as mission success rate, robot team size, and robot component reliability, but only for specific tasks. Here we take a more comprehensive approach in order to draw more general conclusions about robot mission reliability. The approach is based on a mission taxonomy coupled with detailed reliability analysis of each of the resultant mission classes. This paper describes initial work towards that goal.

In this paper we present the above-mentioned taxonomy, which divides planetary robotic missions into subgroups with common characteristics with respect to the time proportion of tasks involved in the missions. For a given mission class, we show how a mission designer can obtain the optimum robot configuration in terms of robot team size and component reliability that maximize mission success rate under a budget constraint.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids;
G.3 [**Probability and Statistics**]: *reliability and life testing, stochastic processes, survival analysis*

General Terms

Performance, Design, Reliability.

Keywords

Mission design, planetary robot, mission taxonomy, reliability, mission cost, failure, robot configuration optimization.

1. INTRODUCTION

Planetary robots built for NASA by the Jet Propulsion Laboratory are notable for their extremely high reliability. To achieve this magnitude of reliability, the robots make use of some of the most reliable components available and provide high redundancy in the design. This design paradigm comes at high financial cost, however, both in the development cost of the robots and in the ongoing operational costs.

One good example of the high cost of NASA robots is the Mars Science Laboratory (MSL). The mission was given the highest scientific priority in NASA's Mars Program of 2002, but then delayed in the 2006 plan as a result of cost constraints [1]. MSL was initially approved at a budget of approximately $1.5 billion [2], but the budget for the mission kept rising until it reached $2.3 billion [3], of which $1.6 billion was development costs for the rover, its instruments and the spacecraft. The fiscal problem has led to the thought of continuing the over-budget MSL mission at the expense of delaying or even cancelling other projects [4].

MSL is one of many in-situ planetary missions NASA plans to launch in the future. If these future robots follow legacy designs, then the increasingly demanding future missions will require robots to be built using components of order-of-magnitude higher quality, rendering the mission eventually infeasible from a cost and availability standpoint.

Reduction of robot development costs can be achieved if overly reliable components are exchanged for ones more in line with the mission requirement. For this, it is necessary to consider the impact of reduced component reliability on the overall mission risk. It is also necessary to regard risk not simply as something to be minimized to the greatest extent possible, but instead as a quantitative design factor to be traded off against other design factors in order to seek an optimal mission configuration. Therefore, tradeoff analysis between mission risk, component reliability and cost is crucial.

A quantitative methodology for doing so has been proposed in our previous work [5] and [6], but has only been used on a limited number of specific examples. As a result, a mission taxonomy is desirable in order to characterize and examine the full range of planetary robotic missions.

A number of taxonomies for robots, robot tasks, and robot teams have been proposed. For instance, [7] provides a taxonomy that classifies multirobot teams in terms of team size and composition, communications, and processing capability. Reference [8] provides a taxonomy which classifies multirobot tasks in terms of criteria such as time, energy and robot capabilities, and [9] & [10] present a taxonomy which categorizes robot tasks in terms of the amount and type of human-robot interaction involved.

In order to have fundamentally different reliability characteristics and therefore tradeoff relationships among mission classes, we

PerMIS'09, September 21–23, 2009, Gaithersburg, MD, USA.

hypothesize that it may be sufficient for the amount of time spent using various modules to differ significantly.

Our taxonomy differs from the taxonomy mentioned above in that it classifies robot missions with respect to the time proportion of the tasks involved in the missions. The breakdown of the time proportion of the tasks in a mission is important for analyzing the nature/emphasis of a mission.

Drawing from the NASA Roadmap for the exploration of the Solar System over the next 30 years [11] and Mars Exploration Program [1] & [12], we propose three mission classes formed from a set of tasks, which we call "Basic Activities", defined in the next section.

Using the method we introduced in a previous paper [6], we characterize each of the mission classes via state transition graphs and investigate the time proportions of basic activities for the various mission classes. Extending our work in [5], we explore the abovementioned tradeoffs by finding the global optimum robot configuration with respect to the cost and reliability for a specific mission class and setting.

We expect significant differences in time allocation of the basic activities to be reflected in significant differences in the character of the tradeoffs.

2. ROBOT MISSION TAXONOMY

2.1 Taxonomy Criterion

In generating a mission taxonomy, it is crucial to create a standard feature with respect to which different missions are compared and classified. It is also important that the feature be quantifiable so that the classification generated can be studied analytically. Identification of this key feature will allow its systematic variation in the step of determining mission reliability. In this light, we attempted to answer the following three questions:

1. Can different types of missions be identified?
2. If missions can be identified, can their features be isolated?
3. If features of a mission can be isolated, can they also be tailored to form another type of mission?

We comprehensively surveyed future in-situ planetary robotic missions from the NASA Solar System Exploration Roadmap (SSER) and Mars Exploration Program (MEP) and identified several fundamental tasks, independent of each other, that are present in all of the missions but exist in different proportions. The mix of these fundamental tasks, which we term "Basic Activities", is the feature with which we measure and compare different missions. We quantify the proportion of a basic activity in a mission in terms of percentage by comparing the time spent in that particular task to the total mission time.

Analyzing every mission instance we encountered in the roadmap, we concluded that a mission can be characterized using the following nine basic activities:

1. Traverse (e.g., driving, flying)
2. Subsurface Access (e.g., drilling, grinding, digging)
3. Instrument Deployment (e.g., manipulator, camera)
4. Sampling (e.g., image, soil)
5. Assembly
6. Communication
7. Sample Analysis
8. Recharging
9. Idling

We do not expect the percentage proportions of basic activities that constitute a specific mission class to be absolutely fixed. Rather, they will fall into a range such that the character of a mission significantly changes only when the proportions exceed that range.

Initially, we performed qualitative separation between mission classes, thus determining the range mentioned above, based on our analysis of SSER and MEP. Afterwards, we will ground-truth the qualitative boundaries we made for each mission class quantitatively using the methodology outlined in subsection 4.1 and use the resulting boundary for differentiating between missions.

2.2 Mission Classes

For the purpose of generating the taxonomy, we did a comprehensive study of the following NASA-proposed in-situ robotic operations:

- Europa Explorer, Europa Astrobiology Lander,
- Titan Explorer,
- Venus In-Situ Explorer, Venus Mobile Explorer, Venus Sample Return,
- Neptune-Triton Explorer,
- Lunar South Pole-Aitken Basin Sample Return,
- Mars Pathfinder-Sojourner, Mars Scout Phoenix, MER, MSL, Astrobiology Field Laboratory, and
- Comet Cryogenic Nucleus Sample Return

Based on the above study, we propose categorizing missions into three classes followed by their examples:

1. Search and Exploration Mission:
    a. Search for biomarker signatures
    b. Search for water resources
    c. Surface mapping

2. Sample Acquisition and Composition Analysis Mission:
    a. Surface rock sampling
    b. Organic materials sampling
    c. Analysis of chemical and isotopic composition of surface

3. Construction Mission:
    a. Radiation shielded habitat construction (Lunar)
    b. Lunar outpost construction (Lunar)

In order to validate the above classification scheme, we use the methodology introduced in our previous work [6] by stochastically simulating a mission class using a state transition diagram. Each state in the diagram corresponds to a basic activity. The process flow and the resulting time proportion of the basic activities are explained in subsections 3.1 and 4.1, respectively.

3. CONSTRUCTION MISSION SCENARIO

3.1 Mission and Environment

Due to NASA's strong interest in building permanent bases in planetary environments, we use the taxonomy above as a framework to consider a Construction Mission class in a planetary environment to install modules at several sites using a team of robots. The installation task consists of carrying modules from a module depot to the designated sites and then assembling them. We extend our previous work of simply carrying a module to a site and repeating the task [5] into a more mature scenario that better resembles a general planetary in-situ construction mission. This allows us to consider energy limitations on the robots and further elaborate the robot model and the working environment.

Based on [13] and [14], we envision that in a construction mission at least three types of location would exist: a battery recharging station (i.e., solar panel plant, robot base), a module depository and the construction sites. As also stated in the literature, we expect that in a planetary environment there exist areas that receive steadier sunlight; thus, solar power generation on a stationary site would be more efficient than on the peripatetic robots, which would potentially work in a shadowed and dusty environment. However, our methodology also works in the case where a site co-locates with another or in the case where all of them co-locate in one place.

This environment model is shown in Figure 1. The locations are represented as nodes with the distance between locations written as weights (in meters) on the edges. Connected nodes show the possible paths the robots can take. We selected the weights in the figure as our baseline model and varied them in the simulation.

For the purpose of the reliability analysis, the mission is broken down into seven basic activities:

1. Transit to the module depot
2. Fetch modules
3. Transit to the construction site
4. Stack modules
5. Assemble modules
   Repeat 1 – 5 until all sites are completed.
6. Communicate with other robots after every subtask.
7. When needed, return to the recharging site (via depot as checkpoint) and replenish battery.

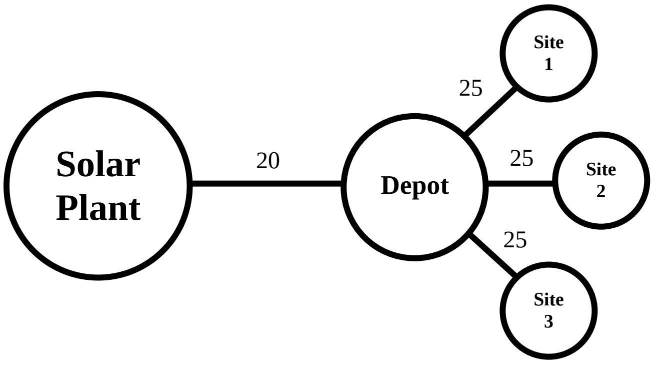


Figure 1. Environment model in graph diagram.

Each robot works independently of the others and coordination to avoid overlapping tasks (e.g., carrying more modules than needed, assigning more robots than needed to install a module, etc.) is done by communication among the robots after the completion of each subtask. In the case of a failure of a robot, a new spare robot will be deployed from the headquarters to continue the mission. The mission is considered a success when all the necessary modules are installed at the sites.

3.2 Robots and Components

For this analysis we assume that the robots are identical. Making appropriate inference from various sources [14], [16], we assume each robot weighs 174 kg and uses two 7.15-kg lithium ion batteries (150 W-h/kg) for energy storage. The power consumption model used in the simulation is listed in Table 1. The robot velocity is assumed to be a constant 0.1 m/s throughout the mission.

The robot subsystem reliabilities are listed in Table 2. The subsystem reliability data were derived from component reliability data provided by the Jet Propulsion Laboratory and are representative of components used in NASA's planetary robots. The usage times of each subsystem for each basic activity are shown in Table 3. These usage times were assigned using reasonable assumptions about the relative durations of different activities and the relative usage of different modules.

We used the above-described model and the methodology in our previous work [6] to calculate the probability of subsystem survival for a given mission activity. In this calculation, we assumed that the battery recharging task is always successful and hence excluded it from the reliability consideration. We also assumed that the recharging station is always capable of generating maximum energy and fully recharging batteries regardless of climate disturbances, e.g., dust storms, dust build-up, or poor sun exposure.

Table 1. Power consumption model

Basic Activity	Power Consumption
Traverse	150 W
Instrument Deployment	52 W
Communication	74 W
Assembly	52 W
Idling	15 W

Table 2. Robot subsystems and reliabilities

Subsystem	MTTF (h)
Power	4202
Computation & Sensing	4769
Mobility	19724
Communications	11876
Manipulator	13793

Table 3. Subsystem usage by task in minutes using baseline constant in Table 1

Subsystem	Transit (Solar Plant -Depot)	Fetch / Stack Modules	Transit (Depot-Site)
Power	33.33	60	4.17
Computation & Sensing	33.33	60	4.17
Mobility	33.33	30	4.17
Communications	0	0	0
Manipulator	0	60	0

Subsystem	Module Assembling	Communication
Power	300	15
Computation & Sensing	300	15
Mobility	120	0
Communications	0	15
Manipulator	300	0

4. APPROACH

We generated a state-transition diagram for the Construction class mission based on the mission flow described in subsection 3.1. The state machines represented by the diagram are then implemented in simulation software. The simulation is repeated many times, with the average score of all trials giving the overall probability of mission completion (PoMC).

For this mission scenario, once the mission flow, the basic activity durations and the baseline module reliabilities are fixed, then the input variables are:

- Number of installation sites
- Number of modules to be installed per site
- Number of robots
- Number of spare robots
- Reliability of the components used
- Maximum number of modules a robot can carry
- Distance between recharging site and module depot
- Distance between module depot and installation sites

Thus, PoMC and the time proportion of the basic activities are functions of these input variables and varying these variables results in a change in the output PoMC and the time proportion.

4.1 Time Proportion of Construction Class

Given the hyper-dimensionality of the model, we simplify the analysis by varying only one variable at a time and fixing the rest, and looking at the relationship between the varied variable and the time proportion of the basic activities in the Construction mission class as well as PoMC.

Graphically, this means that we are reducing the dimensionality of the model by only analyzing a slice of the hyper-plane at a time. Intuitively, the function of PoMC with respect to the variable being varied and the time proportion would only be valid at that particular slice and might not hold at different set of values of the remaining variables. In that light, here we would like to see how much the time proportion of the basic activities will vary for different slices of the hyper-plane.

In every simulation with mission success rate greater than zero, we record the time spent on the basic activities. We set the baseline variables as shown in Table 4 and then increment the variable to be varied along the x-axis from the minimum to the maximum expected value.

Our result shows that the time proportion of the basic activities in the Construction mission class does not vary greatly between different slices of the hyper-plane (see Table 5).

Our sensitivity analysis shows that the number of installation sites and the number of modules to be installed per site have small influence on the time proportions. Intuitively, an increase in the amount of work (i.e., number of sites and/or modules) leads to an increase in the time share of module assembling. However, due to the limitation on the number of modules that a robot can carry, the robots are forced to return to Depot to fetch more modules, hence increasing the time proportion of other activities, as well (see Figure 2).

The variables that have the most influence on the time proportions are the number of robots deployed and the distance (i.e., Solar plant to depot, depot to construction sites) to be travelled by them. For the latter, the reason is straightforward: an increase in either

Table 4. Baseline constants used in the simulation

Variable	Value
#Sites	10
#Modules/site	5
#Robots	2
#Spare robots	0
%MTTF	100%
#Module capacity/robot	3
d(Solar Plant - Depot)	200
d(Depot – Site)	25

Table 5. Time proportion in the Construction class mission

Basic Activities	% of Mission Time (±2%)
Traverse	4
Instrument Deployment	21
Module Assembling	47
Communication	17
Recharging	11
Idling	< 1

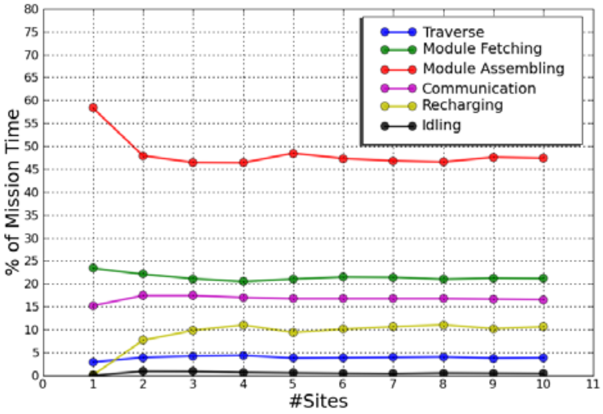


Figure 2. Varying number of installation sites.

of these distances directly increases the time spent in Traverse, thus increasing its relative proportion to the remaining basic activities. For the former, increasing the number of robots proportionally increases the amount of coordination (hence, communication) among the robots. By fixing the amount of work (i.e., number of sites and/or modules), the coordination to avoid overlapping tasks between the robots causes deploying more robots to result in some of the robots' being idle (see Figure 3). However, if the amount of work increases proportionally with the number of robots deployed, then the time proportion of the basic activities will follow the general proportion described in Table 5.

Indeed, the resulting time proportion significantly depends on the robot work rate in performing an activity (see Table 3). However, we also have confirmed that the time proportion for each activity does not fluctuate drastically (still falls within a small range) and observed the same sensitivity pattern in different models.

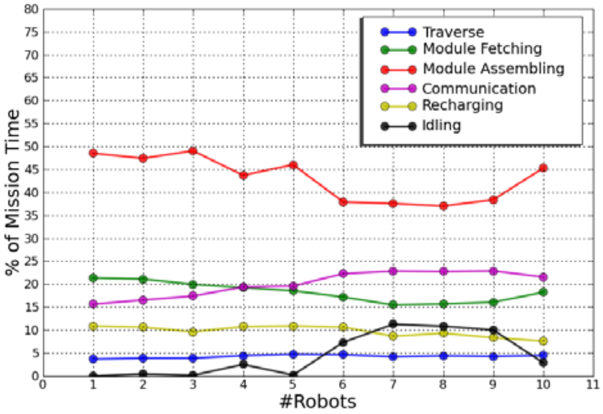


Figure 3. Varying number of robots.

4.2 Designer Questions

In the conceptual mission designing phase, a mission designer might ask questions such as:

Given a fixed budget and a fixed number of sites to build,

1. Which configurations of robots in terms of team size and robot component reliability (%MTTF) give the highest mission success rate?
2. Which is better, a smaller number of robots with highly reliable components or a larger number of robots with less reliable components?

4.3 Cost Function

We adopt the general relationship of reliability and cost, where cost is an exponential function of component reliability. The exponential relation means that, the higher the reliability of a component, the smaller increase in reliability per unit expenditure. We assume that the robot component cost is a linear combination of two types of cost function: component material cost and production cost where each is represented as an increasing exponential function. The total cost function (robot component cost) is then given as follows:

$$C(R) = k_1 e^{k_2 R + k_3} + c_1 e^{c_2 R + c_3}, \qquad (1)$$

where

$R =$ percentage of component reliability compared to the baseline model

$k_1 =$ the weight of component material cost

$c_1 =$ the weight of component production cost

$k_2, k_3, c_2, c_3 =$ parameters to adjust the initial cost (when R=0%) and the cost when R=100%

For analysis purposes, we assume that the component material cost and production cost contribute equally to the total component cost. We also assume that there is still a cost to be incurred even when producing a poor-quality component (i.e., reliability R=0). For this purpose, we set the parameters k_1, k_2, k_3, c_2, c_3 such that the initial component cost (when R=0) is 20% of the total budget. Note that $k_i = c_i$ for $i = 1,2,3$. The details of the parameters are given in Equation 2 and a plot of the cost function for different robot team sizes is shown in Figure 4.

$$C(R) = 2e^{1.61R+2.3}, \qquad (2)$$

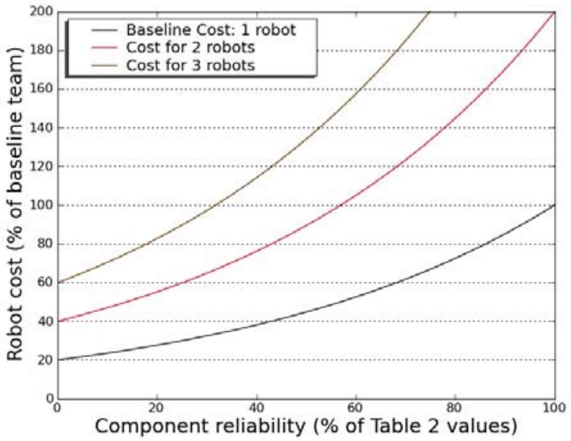


Figure 4. Component cost (=material cost and production cost) as a function of component reliability (%MTTF).

It is noteworthy that our methodology works with any cost function. The cost model described here serves as an alternative example to the cost model used in [5], which was taken from reference [15]. Both cost models are monotonically increasing functions of component reliability. However, the latter has a drastic gradient between low- and high-reliability components such that the cost of a high-reliability component is very high (i.e., an increase in 5% component reliability from 90% reliability to 95% reliability would result in a large price increase from 60% to 100% of the baseline cost) and the cost of a low-reliability component is very low (i.e., an increase in 40% component reliability from 40% reliability to 80% reliability results only in minor price increase from 40% to 47% of the baseline cost). It is possible to attenuate the extreme to some extent by lowering the feasibility parameter provided, but the maximum achievable reliability will also be greatly lowered.

The cost model we propose here provides a more gradual increase in unit expenditure per increase in component reliability. In subsections 4.4 & 4.5, we will observe the outcome of the reliability tradeoff using both cost functions.

4.4 Optimizing Robot Configuration

Using the cost model from the previous subsection, we seek to optimize the robot configuration for the Construction mission class with respect to the criteria posed in subsection 4.2.

A mission designer would presumably like to design as reliable a system as possible under budget constraints while achieving the highest possible mission success rate. This issue relates closely to our tradeoff model between component reliability (%MTTF), robot team size, cost and probability of mission completion (PoMC). The idea is to come up with a robot team size with a certain component reliability that can maximize PoMC under a fixed budget.

In Figure 5, using the data listed in Table 4 as the input variables, we plot several tradeoff relations between component reliability (%MTTF) and mission success rate (PoMC) for different robot team sizes (from 2 to 5 in red, brown, green, yellow lines). We have also fitted a curve to these points, allowing this equalizing %MTTF to be estimated for intermediate points without running additional simulations. The black horizontal line shows the PoMC for the baseline configuration (1 robot with 100% component reliability).

Based on the cost model for one robot (Equation 2), we calculate the respective budget needed for 2 to 5 robots. By doing so, we are able to compute the maximum achievable component reliability (%MTTF) under the budget constraint for each team size. This is shown as dashed vertical lines in Figure 5.

The intersection between the dashed vertical lines and the PoMC curve (a function of %MTTF) then gives the maximum achievable PoMC for each team configuration (using the maximum achievable %MTTF) given the budget constraint.

In this graph, we have 4 intersections (for 2 to 5 robots). Analysis of all the intersections shows that in this mission scenario of the Construction mission class, a configuration of 3 robots with 31.9% MTTF of the baseline reliability (see Table 2) gives the highest probability of mission success under the budget constraint.

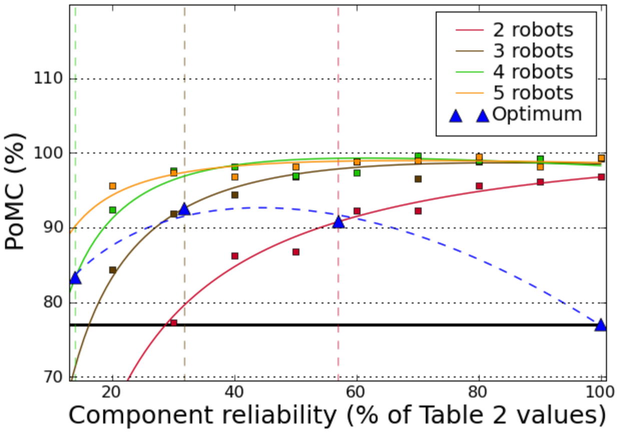


Figure 5. Tradeoff between %MTTF, PoMC and Cost for various team sizes when #Sites = 10.

Mission designers may replace the component reliability – cost model utilized here with their own. Obviously, using a different cost model would potentially result in a different optimum robot configuration. For example, with the same initial component cost (when R=0) of 20% from the total budget, using the cost model we utilized in our previous paper ($R_{min} = 0, R_{max} = 1, f = 0.8$) [5] would result in a robot team size of 4 with 52.7% MTTF of the baseline reliability being the optimum configuration under the budget constraint.

4.5 Robot Team Size-Reliability Tradeoff

Previously in [6], we answered this tradeoff question by providing an example of how a team of 4 robots with less reliable components has a higher mission success rate (PoMC) than a team of 2 robots with more reliable components. Here we would like to corroborate that statement and extend the analysis by observing it through the optimization methodology described above.

Using the same scenario as in the previous subsection and the cost model from Equation 2, we increase the number of construction sites from 10 to 50 and plot the tradeoff graph in Figure 6. Note that increasing the number of construction sites also increases the mission duration. In the figure, we can see that the optimum robot configuration is no longer 3 robots with 31.9% MTTF of the baseline reliability, but 2 robots with 57.1% MTTF of the baseline reliability. In other words, increasing the mission duration causes the preferred configuration to move in the direction of fewer robots with higher reliability.

Our analysis shows that there exists a turning point where a larger robot team with less reliable components will perform exactly the same (in terms of PoMC) as a smaller robot team with more reliable components when we increase the amount of work (i.e., number of sites and/or modules). Going beyond that turning point results in the smaller robot team with more reliable components producing a higher mission success rate.

The turning point can be explained from the reliability engineering point of view. Reliability is a function of time where the reliability of a component with a constant hazard rate is equal

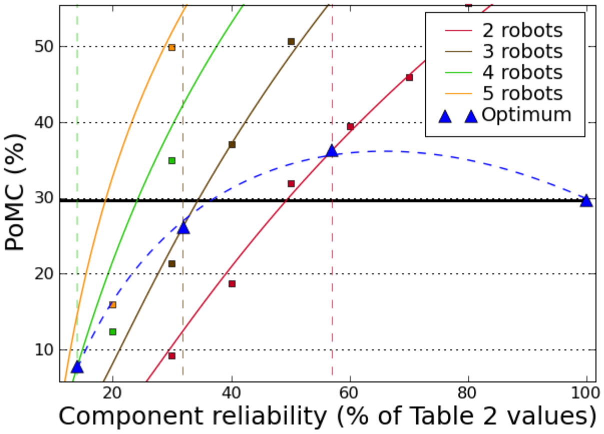


Figure 6. Tradeoff between %MTTF, PoMC and Cost for various team sizes when #Sites = 50.

to one at the beginning of the service life and decays exponentially towards zero. Thus, using components with reduced reliability compared to the baseline component will result in shorter service life and higher failure probability for prolonged usage.

Since we assumed that a failure of any single component leads to a failure of the entire robot, the robot using the components with reduced reliability will likely have a smaller mission success rate for longer mission duration. Likelihood of mission failure can be compensated for by increasing the number of robots. However, this compensation may not be enough to cover the loss in component reliability, especially when the budget is constrained.

Consider a simple example where a robot has a total system reliability expressed in MTTF (hour) of 4202. The budget constraint allows us to double the number of the robots while halving the component reliability of both robots. Now, we compare the mission success rate for both configurations of 1 robot with 4202 MTTF(h) and 2 robots with 2101 MTTF(h). The mission success condition is such that one robot must stay alive in order to complete the mission. For 4000 hours of usage, the mission success rates (PoMC) for both configurations can be calculated as follows:

1 robot, 4202 MTTF(h): $PoMC = e^{-\frac{4000}{4202}} = 0.386$

2 robots, 2101 MTTF(h) each:

For 1 robot, $PoMC = e^{-\frac{4000}{2101}} = 0.149$

For 2 robots,

$$PoMC = 0.149^2 + 2 \times 0.149 \times 0.851 = 0.276$$

Now, we do the same calculation for a mission of 1000 hours:

1 robot, 4202 MTTF(h): $PoMC = e^{-\frac{1000}{4202}} = 0.700$

2 robots, 2101 MTTF(h) each:

For 1 robot, $PoMC = e^{-\frac{1000}{2101}} = 0.621$

For 2 robots,

$$PoMC = 0.621^2 + 2 \times 0.621 \times 0.379 = 0.856$$

Here we can see that for the prolonged 4000 hours mission, for 2 robots 2101 MTTF(h), the loss in PoMC due to the halved component reliability is 0.237 (= 0.386 - 0.149). However, the compensation of doubling the robot number only increases PoMC by 0.127 (= 0.276 - 0.149), which is not enough to cover the loss in PoMC due to reduced component reliability. In the shorter mission of 1000 hours, the loss in PoMC due to halved component reliability is 0.167 (= 0.788 - 0.621) and safely covered by the increase in PoMC of 0.235 (= 0.856 - 0.621) due to the doubling of the number of robots.

In this light, our analysis of various tradeoff cases suggests that, for relatively short missions, the PoMC gain per robot number increase has the likelihood to be larger than the loss of PoMC per component reliability decrease. For relatively long missions, the loss is typically greater than the gain.

Because the maximum achievable number of robots is dependent on the component reliability – cost model, the location of the turning point is also dependent on the cost model. If the cost of mass-producing robots with low-reliability components is cheap enough that PoMC gain per robot number increase is always larger than PoMC loss per component reliability decrease, then the optimization methodology described in the previous subsection would have the general tendency to increase the robot team size and use lower-reliability components.

5. CONCLUSIONS

This paper presents a general framework to explore the tradeoffs among cost, component reliability, and robot team size in planetary robot missions. We comprehensively surveyed every mission instance proposed in NASA's Solar System Exploration Roadmap over the next 30 years [11] and Mars Exploration Program [1] to generate a mission taxonomy. We propose that any mission instances can be characterized by a set of basic mission activities and categorize planetary robot missions based on the time proportion of the activities into three classes: Search and Exploration, Sample Acquisition and Composition Analysis, and Construction. We examined a general mission scenario in the Construction mission class and performed sensitivity analysis of the tradeoff parameters. Our results show the stability of the Construction mission class with respect to the activity time proportions.

In this paper, we also propose a method allowing a mission designer to optimize robot configuration in terms of robot team size and component reliability with respect to probability of mission success (PoMC), given a cost model. The method is not limited to a particular cost model. Mission designers can use an arbitrary cost model and implement the methodology to obtain a specific optimum robot configuration that maximizes mission success rate under a budget constraint.

For the Construction mission class, our analysis shows that for long-term missions, the PoMC loss per component reliability decrease is typically greater than the PoMC gain per robot number increase. Thus, it is more beneficial in terms of the mission success rate to increase the quality of the robot components as opposed to the number of robots, given the budget constraint. For short missions, however, the opposite trend can be observed, thus, configurations of more robots with less reliable components are preferable.

In future work, we will investigate other mission classes and perform sensitivity analysis of the parameters on the mission classes. We intend to compare the time proportion and reliability tradeoff among the classes to see distinguishable characteristics among them as well as to validate the classification scheme proposed in this paper. In addition, we intend to extend the optimization problem to cost and mission duration. For instance, we seek to answer common mission designer questions like “Under a given mission duration, which configurations of robots result in the highest mission success rate?” or “Given a desirable mission reliability standard, which configurations of robots cost the least?”

6. ACKNOWLEDGMENTS

This research was funded by the National Aeronautics and Space Administration (NASA) under award NNX07AV71G S01 in the Advanced Intelligent Systems Research (AISR) program.

7. REFERENCES

[1] Mars Advanced Planning Group. 2006. Robotic Mars Exploration Strategy 2007-2016. Available from: http://mepag.jpl.nasa.gov/reports/3715_Mars_Expl_Strat_GPO.pdf

[2] NASA. 2007. FY 2008 Budget Estimates. Available from: http://www.nasa.gov/pdf/168652main_NASA_FY08_Budget_Request.pdf

[3] NASA. 2009. FY 2010 Budget Estimates. Available from: http://www.nasa.gov/pdf/345225main_FY_2010_UPDATED_final_5-11-09_with_cover.pdf

[4] Green, J. L. 2008. MSL Cost Overrun Status and Plans. Available from: http://www.lpi.usra.edu/pss/presentations/200810/greenMSL.pdf

[5] Stancliff, S. B., Dolan, J., and Trebi-Ollennu, A. 2007. Planning to Fail – Reliability as a Design Parameter for Planetary Rover Missions. In Proceedings of the 2007 Workshop on Measuring Performance and Intelligence of Intelligent Systems. PerMIS '07.

[6] Stancliff, S. B., Dolan, J., and Trebi-Ollennu, A. 2006. Mission Reliability Estimation for Multirobot Team Design. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (Oct. 2006), pp. 2206 - 2211.

[7] Dudek, G., Jenkin, M. R. M., Milios, E., and Wilkes, D. 1996. A Taxonomy for Multi-Agent Robotics. Autonomous Robots, vol. 3, no. 4, pp. 375 - 397.

[8] Balch, T. 2002. Taxonomies of Multirobot Task and Reward. In Robot Teams: From Diversity to Polymorphism, Balch, T. and Parker, L. E., eds. Natick, MA: A K Peters.

[9] Yanco, H. A. and Drury, J. 2002. A Taxonomy for Human-Robot Interaction. In Proceedings of the AAAI Fall Symposium on Human-Robot Interaction, AAAI Technical Report FS-02-03, Falmouth, Massachusetts, November 2002, pp. 111 - 119.

[10] Yanco, H. A. and Drury, J. 2004. Classifying Human-Robot Interaction: An Updated Taxonomy. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics (Oct. 2004).

[11] NASA. 2006. Solar System Exploration. Available from: http://www.lpi.usra.edu/vexag/road_map_final.pdf

[12] Mars Advance Planning Group. 2006. 2006 Update to Robotic Mars Exploration Strategy 2007 - 2016. Available from: http://mepag.jpl.nasa.gov/reports/MAPG_2006_Update_Final-1.pdf

[13] NASA. 2005. NASA’s Exploration Systems Architecture Study, part 4: Lunar Architecture, section 4.2.5.1.4 and 4.3. Available from: http://www.nasa.gov/pdf/140649main_ESAS_full.pdf

[14] Skonieczny, K., DiGioia, M. E., Barsa, R. L., Wettergreen, D. S., and Whittaker, W. L. 2009. Configuring Innovative Regolith Moving Techniques for Lunar Outposts. Aerospace conference, 2009 IEEE, pp. 1 - 11.

[15] Mettas, A. 2000. Reliability Allocation and Optimization for Complex Systems. In Proceedings of the IEEE Annual Reliability and Maintainability Symposium, pp. 216 - 221.

[16] JPL-PACC. 2000. Presentation on Power Management in Past and Present JPL/NASA Missions (Sept. 26, 2000). Available from: http://newport.eecs.uci.edu/impacct/d_research/d_presentation/JPL-PACC092600.ppt

Towards a systematic assessment of the functions of unmanned autonomous systems

Robin JAULMES
+33 1 42 31 95 33
robin.jaulmes
@dga.defense.gouv.fr

Eric MOLINE
+33 1 42 31 96 59
eric.moline
@dga.defense.gouv.fr

Laurent VIELLE
+33 1 42 31 95 84
laurent.vielle
@dga.defense.gouv.fr

Paris Expertise Center (CEP)
16 bis avenue Prieur de la Cote d'Or

ABSTRACT

Being able to assess the performance of the algorithmic components of unmanned autonomous systems is a necessity. Defining repeatable and commonly shared test protocols to assess the performance of the algorithms involved in autonomy is the key to achieve standardization in the unmanned autonomous systems field (Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs)).

This paper proposes a generic methodology to evaluate any function of an autonomous system and illustrates the methodology on two examples: for the evaluation of visual beacon tracking algorithm and for the evaluation of Simultaneous Localization And Mapping (SLAM) algorithms. The lessons learnt from these evaluations are then described.

Categories and Subject Descriptors

I.2.9 **[Artificial Intelligence]** Robotics and Autonomous vehicles – Performance measurement, methodology, standardization.

General Terms

Algorithms, Measurement, Performance, Experimentation, Standardization, Verification.

Keywords

Unmanned systems, Evaluation metrics, Databases, Evaluation protocols, Standardization, SLAM, perception algorithms.

1. INTRODUCTION

Unmanned systems ground and aerial, are more and more mature: their architecture can today involve complex data and image processing, data fusion, optimization, and learning algorithms, allowing them to be used in more and more complex environments. They are able to navigate through urban areas, with very little monitoring, and to execute very high level commands like "rally this goal while following the traffic rules and avoiding obstacles", as was achieved recently in the Urban Challenge [23].

However, for such complex systems, the performance and reliability is very important: the more reliable these systems are, the more they will be adapted to challenging environments and the more an operator will gain confidence using then. We believe that the precise assessment of reliability and performance becomes mandatory to assess the "level of autonomy". Ideally, the more autonomous a system is, the less an operator needs to intervene. In a "fully autonomous" system, a human would only intervene when need to change the current goal arises, the system being able to overcome any difficulty.

As is often done in systems engineering, the overall reliability of the system can be obtained from the assessment of the reliability of its different components; many of the components in unmanned autonomous systems are widely used for other applications, so evaluating their reliability and performance is a "solved" problem. However, the "innovating" algorithms used for perception, navigation, data fusion or planning are, for most of them, specific to the autonomous vehicles field, and evaluating their performance is still to be standardized.

To address this issue, we have designed a generic methodology for the evaluation of elementary components and are applying it.

After presenting briefly prior work in the domain, we present our methodology and illustrate it two examples, one taken from our work in the TAROT advanced study program (Technologies essentielles pour l'Autonomie des RObots Terrestres), the other addressing a classical algorithm of the literature: SLAM using a LIght Detection And Ranging (LIDAR) sensor.

2. PRIOR WORK

There are many algorithms involved in unmanned autonomous systems and some of them have already be evaluated for other applications : for instance, a set of metrics and benchmark data is available for visual vehicle tracking from air [4], for stereo depth estimation [16] and for vehicle detection [5].

Furthermore, the topic of the assessment of SLAM is more and more studied in the literature [2,3,18]. Among them, the more recent is the RAWSEEDS European project that has gathered a data set for the evaluation of SLAM algorithms and has defined a set of metrics to assess mapping and navigation quality (absolute trajectory error, mapping error, relative pose error, rough estimate of complexity, self localization error) [21]. Also note that algorithm [19] and data set repository [20] exist on this topic.

Similar work concerns the definition of good experimental methodologies in robotics [1]. Also note that the Joint Architecture for Unmanned Systems (JAUS) aims at the normalization all the interfaces between components of unmanned systems [22].

An important work on the topic has been made by the National Institute of Standards and Technology (NIST) with the definition of the Autonomous Levels For Unmanned Systems (ALFUS) [11]: this work has defined standard terminology and principles to measure the autonomy of unmanned systems.

The methodology we present here has matured through successive projects: first, we applied a similar methodology within the autonomy axis of the advanced study program "robotique", 2002-2005, to evaluate the performance of perception algorithms, and especially unstructured road detectors [7].

Real maturation has only been obtained within the "TAROT" in 2006-2009 [14], that is the continuation of these projects, and that aims at developing elementary autonomous behaviors for unmanned ground systems: beacon rallying, vehicle following, vertical reference following, SLAM and obstacle avoidance.

3. OUR METHODOLOGY

Our methodology involves four phases: (1) a prior analysis, (2) the preparation of the evaluations, (3) the realization of the evaluations and (4) a posterior analysis.

3.1 Prior analysis

The first phase begins with an analysis of the documentation, of the literature concerning similar algorithms, and of the previous evaluations of the assessed algorithm. It is useful to have as much details as possible on the intrinsic performances of the algorithm and to understand the role of all its independent parameters. The knowledge of the role of the algorithm within the unmanned autonomous vehicle system and its operational use allow the definition of a list of evaluation criteria, which are ideally more or less the technical specifications of the algorithm.

Furthermore, when an algorithm is composed of several elementary components, as it is illustrated on the case of a stereo obstacle avoidance solution in Figure 1, it is very useful to understand clearly the role of each component: only then can the evaluations of these components be used to refine the analysis.

Each criterion is either associated to a wish on the "domain of validity" of the algorithm, or to a wish on its "level of performance", which is linked to the executed function, to the processing time, to the failure rate, and to the computer resources needed.

Each criterion needs to be associated with a set of metrics and with **high level requirements** of the experimentations.

For instance, if we consider the criterion "performance of the ability of the algorithm to detect its own failures[1]", the set of metrics can be:

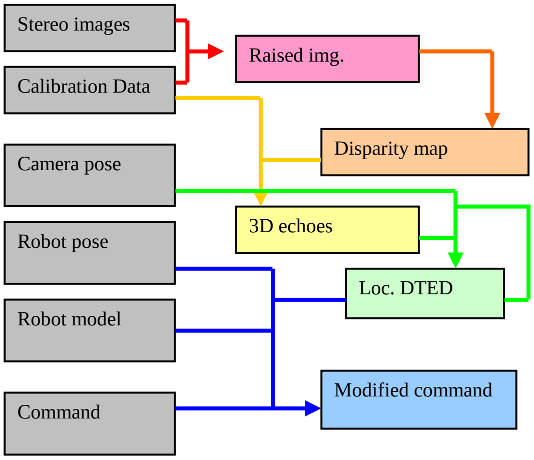


Figure 1 : An analysis illustration on a stereo obstacle avoidance algorithm

(1) Nfu/Nf, where Nfu is the number of times a failure occurs and is undetected in the set of experiments and Nf is the number of times a failure occurs (=1 if no failure occur).

(2) Nfd/Nok, where Nfd is the number of "false failure detection" and Nok is the number of test where no failure occurs.

The same procedure is followed for every criterion, and each criterion is prioritized. We recommend to as much as possible find in the literature the usual criteria that are used to assess the performance of similar algorithms, so that the performance can easily be compared with previous work.

3.2 Evaluation preparation phase

The goal of the second phase is to define precisely the experiments that need to be done and to gather the data and simulation capabilities that are necessary to process the evaluations.

Assessment uses "offline tests", which are processed with technical simulation as well as with open-loop database replay, but not only: it is in most cases necessary to also proceed to "online testing", in which the algorithm is integrated in the system. In order to give more flexibility in the integration and validation phases of projects, it is however absolutely necessary to be able to make a full evaluation according to every criterion without proceeding to online testing, which costs more. That is why we insist on offline testing in the scope of this paper.

The preparation of the evaluations includes a data gathering phase. When public online databases exist [15] or database gathered through previous projects [7] it is always good to re-use them. But specific data gathering may need to be done: when it is the case, we proceed to "data gathering runs" for the robot, in which we record representative multi-sensory data for the algorithms. The data gathering can even be done using the sensors of the future system on a remote-controlled or manned vehicle. Data from multiple sensors (cameras, inertial sensors, wheel encoders, GPS, LIDAR, …) need to be precisely timestamped in a common frame using a dedicated mean [6].

[1] The ability of a given algorithm to detect its own failures allows the operator or other algorithms in the system to be able to correct these failures when they occur.

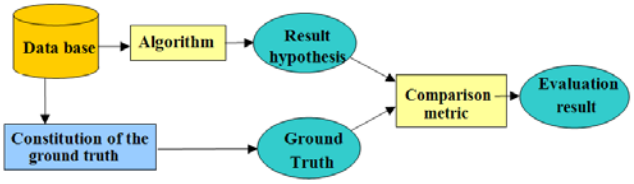


Figure 2 : The evaluation with ground truth

Even if evaluations can be processed simply with a qualitative analysis, it is necessary, for any quantitative measurement to proceed to evaluations *with ground truth (GT)*. The principle of such evaluations is presented in Figure 2.

The *ground truth* is what the algorithm *should* generate, to which the result of the algorithm on the data base needs to be compared. To evaluate, for instance, the performance of navigation and perception functions, the ground truth can be the reference positions of the robot and of elements of the environment. For image tracking and detection algorithms, it can be a reference GT bounding box, that represents the position of the element to detect or track The GT bounding box often needs to be designated by hand, using an interface like the one illustrated on Figure 3.

The simulators we use, which are illustrated on Figure 4, are either well-documented open-source simulators like Gazebo [10] or proprietary software. They are needed when the evaluated algorithms are closed-loop controls and when it is difficult to have the perfect ground truth. We sometimes have to invest on specific simulators in order to evaluate algorithms on precise criteria. It is for instance the role of the *Cadise* simulator, that automatically produces the bounding box ground truth in order to proceed to visual tracking evaluations.

Another useful feature is the ability to "post-process" the real data from the database. For instance, representative noise or additional vibrations[2] can be added to the sensor measurements and images. These post-processing abilities increase the amount of possible experiments (on Figure 5, is illustrated possibilities brought by the addition of noise (upper-left), the use of blur (upper-right), the diminution of contrast (lower-left) and the addition of vibration (lower-right).

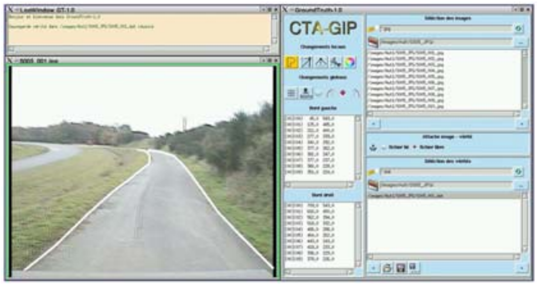


Figure 3 : A ground truth designation interface

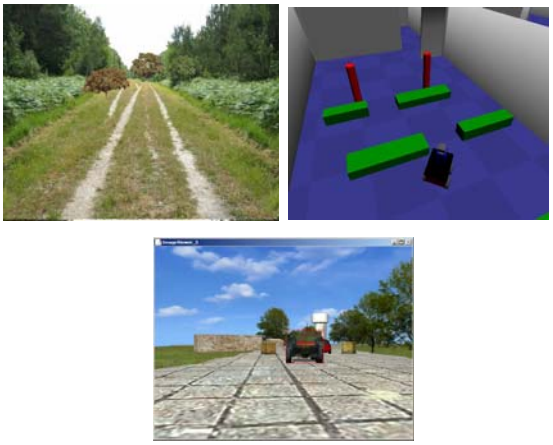


Figure 4 : simulator *Cadise* (up left), *Gazebo* (up right) and *Marilou Robotics Studio* (down)

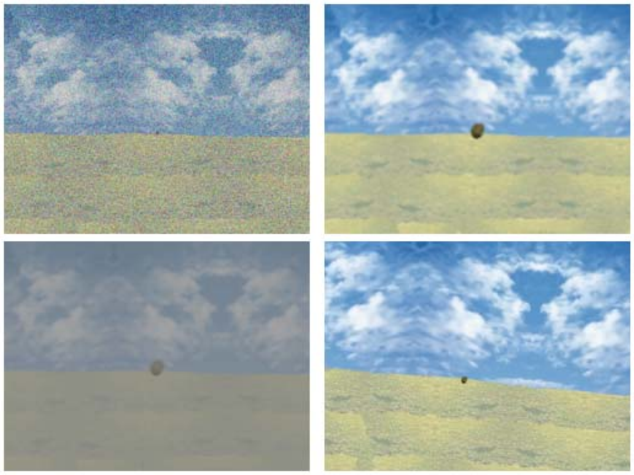


Figure 5 : Illustration of post-processing possibilities

Once the database and the simulation possibilities are designed, the list of experiments is established, taking into account priorities when necessary. We then have to verify that every important criterion is evaluated with a sufficient quality. In order to achieve as most efficiency as possible, each experiment usually evaluates the algorithm according to several criteria, and each criterion is usually evaluated on several experiments.

3.3 Evaluation processing phase

Once the database, the simulations and the test protocols are prepared, the evaluation processing is simple. Specific tools [6], as illustrated on Figure 6, are used to automatically produce the test results.

[2] Vibrations can be added with dynamical cropping and resizing.

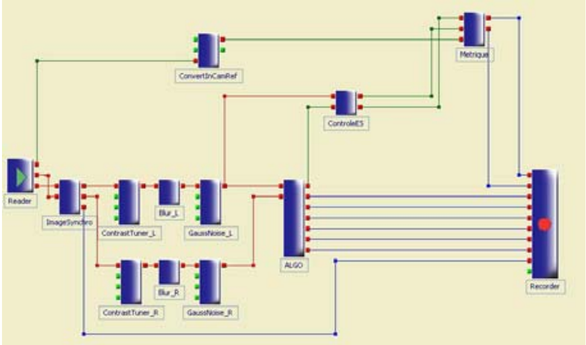


Figure 6 : Illustration of the evaluation software (*RT MAPS*)

3.4 Posterior analysis phase

The test results need to be analyzed in very much depth in order to make a good interpretation of the value of the metrics on each experiment. If inside parameters of the algorithms was accessible and has been measured, this data can be analyzed in order to identify the causes of failures and propose corrections.

Sometimes, during the posterior analysis phase, complementary evaluations appear necessary, for instance to validate hypothesis of unexpected failures: because of this, we insist that the posterior analysis phase should begin before the evaluation processing phase is finished: therefore, the experiment schedule can be modified, when necessary, at the last minute. Furthermore, to guarantee the good quality of the overall analysis, the responsibility of the posterior analysis should be given to a different team than the team realizing the evaluation processing.

We insist on the importance of the posterior analysis phase: this phase gathers all the information and brings a conclusion about the performance of the algorithm: the "resulting" specifications are detailed. During this phase, the experimental protocol that has been followed needs to be analyzed: the conclusions need to take into account the quality and the amount of the tests that have been made. The analysis determines if the algorithm is mature enough to answer an operational need, and identifies a list of possible evolutions. This list is usually discussed with the producer of the assessed algorithm.

3.5 Results of the evaluation

In the TAROT program, most of the implementation choices on the final robot have been made according to the conclusions of these posterior analyses that were made on the different algorithmic component we assessed.

Outside programs, our team also evaluates the performance of open source algorithms and of published algorithms within the literature in order to sketch the possible performances of future systems. The developments are then capitalized within the HNG architecture [11], which allows the algorithms to be easily adapted from one system to another and to run hybrid simulations.

4. APPLICATIONS

We now illustrate some aspects of the methodology on two examples. The first one has been used within the TAROT program (evaluation of tracking algorithms). The other, more detailed, is an evaluation of classical LIDAR SLAM algorithms.

4.1 Visual beacon tracking (VBT)

The purpose of the VBT is to track a given beacon in a sequence of images. It is an elementary component that can be either used for the mission or for the navigation. It can also be at the core of "intelligent behaviours" like vehicle following and beacon rallying.

4.1.1 Prior analysis

The criteria we have identified for this algorithm are the following:

(1) Reliability

(2) Real-time ability

(3) Ability to track varying targets (poorly textured targets, targets of varying apparent size, with a bad initial designation, etc…)

(4) Precision of the tracking: we have used 2 different metrics, which are illustrated on Figure 7. One of them is the inter-barycentre distance, and the other is the following:

$$m(AR, GT) = \frac{S\left(AR \wedge GT\right)^2}{S(AR)\,S(GT)}$$

S(AR) is the surface of the algorithm area, S(GT) the surface of the ground truth area, and S(AR^GT) is the surface of the intersection area (in blue on Figure 7).

(5) Aptitude to detect its own failures and pertinence of the confidence indicator[3] : we compare the evolution of this indicator with the evolution of the precision of the tracking to measure the performance according to this criterion.

(6) Robustness to a varying environment (low luminosity, fog, back light, shadows, distractions near target, occlusions, noise, blur, vibrations, etc…)

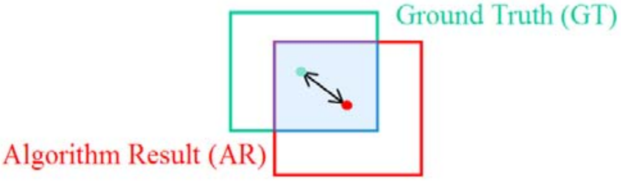


Figure 7 : VBT performance metric illustration

4.1.2 Evaluation preparation

We have gathered and acquired a set of sequences with varying environments (various trajectories and types of roads in different weathers) , varying targets (vehicles, buildings, trees, fence, water

[3] In the TAROT program, we have asked that every algorithm would give a "confidence indicator" representative of the

tower, …), with varying perturbations (occlusions, distracting vehicles, …) and have obtained around 30 sequences, of which a subset is illustrated on Figure 8.

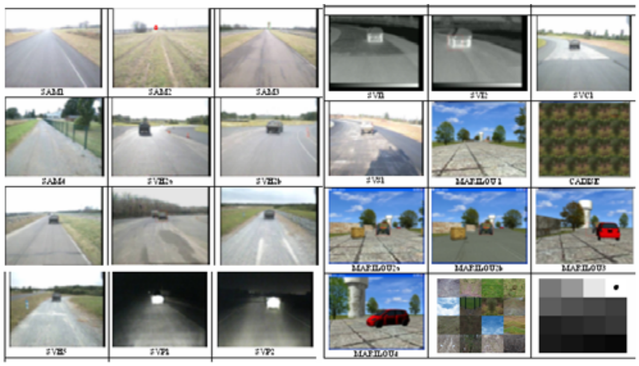


Figure 8 : Extract of the gathered database

4.1.3 *Evaluation processing*

Table 1 gives a sample of summarized results for Algorithms 1 to 4. For the sake of clarity, the metrics have here been translated, using thresholds, into the following qualitative formulations: OK means that the requirement is fully met; AVG means that the result is average, within the “tolerance” zone of the requirement. NOK means that the requirement is not met. “Q of Eval” represents the confidence in the evaluations realised and is equal to “Good” when 3 or more sequences have been used to assess the criterium; and “Average” otherwise.

Table 1 : Extract of assessment results

Criteria	**Q of Eval**	Alg.1	Alg.2	Alg.3	Alg.4
Reliability	Good	OK	OK	OK	OK
Own failures detection	Average	NOK	NOK	NOK	AVG
Quality of confidence indicator	Good	AVG	AVG	NOK	AVG
Real-time capacity	Good	AVG	AVG	OK	OK
Mean perf."beacon" application	Good	AVG	AVG	AVG	AVG
Mean perf. "vehicle" application	Good	AVG	AVG	AVG	AVG
Aptitude to track turning vehicle	Good	NOK	OK	OK	OK
Robustness low brightness	Average	OK	AVG	OK	OK
Compatibility with thermal	Average	OK	AVG	OK	AVG
Robustness to shadows	Average	OK	AVG	OK	AVG
Robustness to clouds	Good	OK	OK	AVG	OK
Robustness to distraction	Average	NOK	AVG	OK	AVG
Robustness to occlusions	Average	NOK	AVG	AVG	AVG
Target texture similar to background	Average	NOK	OK	AVG	NOK
Target of increasing size	Good	OK	AVG	AVG	OK
Very small target	Average	OK	AVG	OK	OK
Very big target	Good	OK	AVG	OK	AVG
Robustness to noise in image	Average	OK	OK	NOK	AVG
Robutness to blur in image	Average	OK	OK	AVG	OK
Robustness to bad designation	Average	AVG	OK	OK	AVG

4.1.4 *Posterior analysis*

The protocol has been used to proceed to the evaluation of, until now, around 10 different algorithm versions. The partners involved in the project have appreciated having a feedback on their developments and the average quality of the algorithms has increased with each successive version, beneficiating from the expertise.

The algorithms which had the most successful evaluations at the end of developments have been associated with other algorithms to build two behaviours (vehicle following and beacon rallying). These behaviours have been demonstrated online on the robotic platform with the validated versions; the limitations observed were exactly the limitations identified through the evaluations. We are confident that our method and metrics captured efficiently the quality of the assessed algorithms.

4.2 LIDAR SLAM component

As we explain in [13], we have evaluated 2 popular LIDAR SLAM algorithms (DPSLAM [8] and FastSLAM [15]) using a similar methodology using our Hybrid Network-based Generic architecture framework [12].

4.2.1 *Prior analysis*

The criteria we have investigated to evaluate SLAM techniques are the following:

- Processing time ;
- Allocated resources (processors, memory) ;
- Precision of the localization (and drift) ;
- Precision of the produced map ;
- Robustness to noise in the wheel encoders ;
- Robustness to noise in the LIDAR scans ;
- Ability to work in cluttered environments ;
- Ability to correctly map loops.

Assessing the precision of the localization brought by SLAM is made like this: the error between the real position and the estimated position is measured. The drift is defined as this error divided by the distance travelled.

Measuring the quality of the produced map is harder and no universally accepted quantitative approach exists yet. For this study we defined the utility metric Q to measure between 0 and 1 the quality of a produced map: this metric was chosen after considering several utility metrics: to neutral persons represented relatively well the correctness of a given map (a map with Q=1 being a perfect map, and a map with Q=0 being completely wrong). This metric uses the same information as the mapping error defined within the RAWSEEDS project [21], which is also based on “control points” but summarizes the quality of the map with a single number, which is independent of the mapped environment.

To define Q, N control points are chosen from the “distinctive” points (e.g. corners) of the ground truth map. The produced map, result of the algorithm, is scaled and superposed to the ground truth map, and the error on the position of each ground truth point d_i is measured (as showed on Figure 9). Let W be the width of the ground truth map and H be its height; let k be a constant[4]. Then,

[4] We chose k=3. With this value, a map in which the mean error is equal the diagonal of the ground truth map has a value of approximately 0.001.

$$Q = \exp\left(-\frac{k}{\sqrt{W^2 + H^2}} \frac{1}{N} \sum_{i=1}^{N} d_i \right)$$

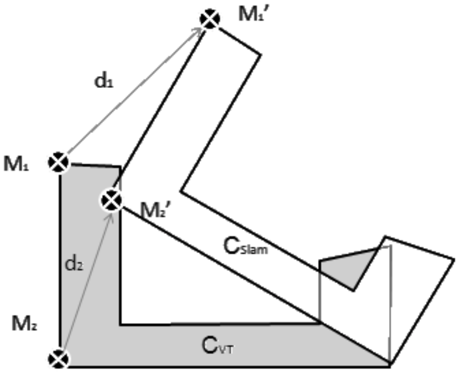


Figure 9 : SLAM performance metric

Note that in order to measure Q, an operator has to designate, for each produced map, the position of each of the control point. When two or more produced points correspond to the same point on the ground truth map (which happens in poorly mapped loops), the farthest point is selected, and we also record the maximum distance between them, which is an indicator of the poorness of the algorithm to map loops. When no corresponding point can be found for one of the control points[5], Q=0

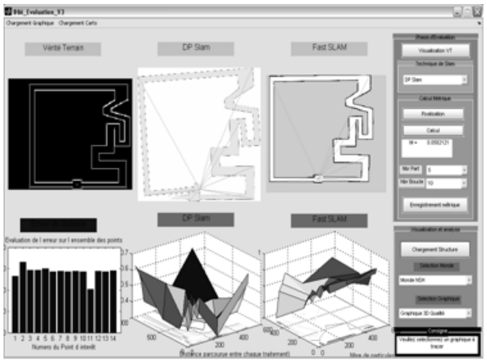
4.2.2 Evaluation preparation

We have defined a user interface to do a systematic assessment of the quality of the produced map, of which a screenshot is given on Figure 10. For more vast experiments, the process would need to be automatic.

To assess the robustness to noise in the wheel encoders, we have added several times on one of the simulated log a controlled noise. We have added an additive Gaussian noise of standard deviation σ at each step for heading and movement, and from one experiment to the following we have increased the value of the variance.

We have also acquired a dataset from our platform in our laboratory (with “real” noise). The ground truth was obtained by scanning the map of the building.

We have built specific maps to assess the abilities to map in cluttered environment and to correctly map loops. We also used a data set (NSH_level_a) from the robotic data repository website [19]. Some of the ground truth maps of our data set are illustrated on Figure 11



DP Slam

M = 0.0592121

Nbr Part 5

Nbr Boucle 10

Monde NSH

Graphique 3D Qualité

Veuillez sélectionnez un graphique à tracer

Figure 10 : the control point designation interface

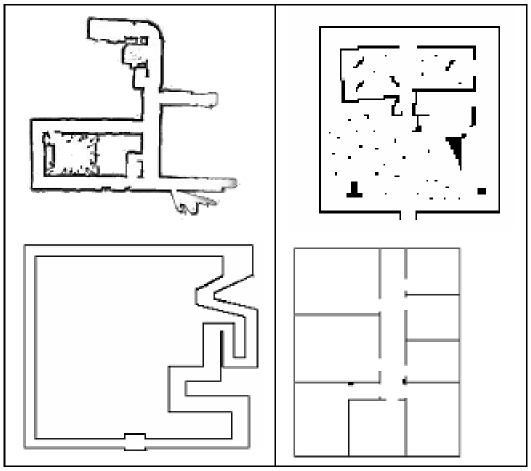


Figure 11 : sample maps from our dataset

(Upper left: NSH_level_a, upper right: cluttered, bottom left: loop, bottom right: Arcueil laboratory

4.2.3 Evaluation processing

We have measured the computing time, the memory resources we used, and the quality of the produced maps on the different maps of our dataset for different set of parameters : some of our results are shown on Figure 12.

On the real world data we acquired, we have had varying performances depending on the experimental conditions. On the first data set we acquired, FastSLAM failed (Q=0) but DPSLAM managed to build something on which we can nearly recognize the different rooms: it is shown on Figure 13.

We believe that this difference was due to the presence on this particular trajectory of abrupt curves. With other tests we made, without hard curves, the performance of both algorithms was correct. We believe that the dynamic model used by default by FastSLAM is not adapted to harsh turns made by our platforms (that are poorly measured by the wheel encoders).

[5] This is usually the symptom of a serious failure in the algorithm.

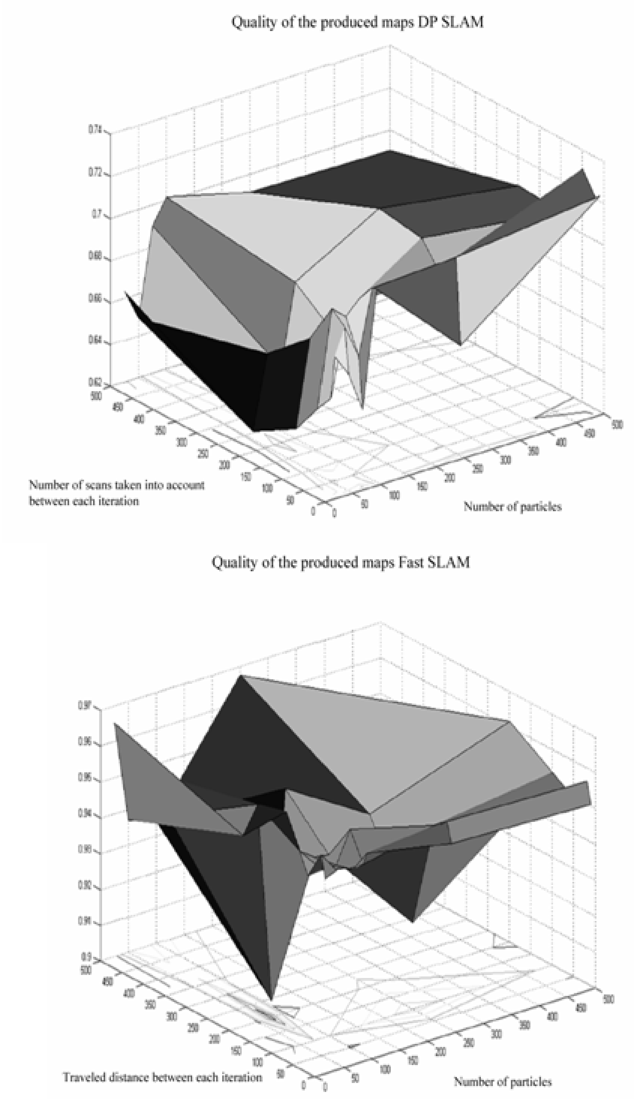


Figure 12 : Sample results on SLAM evaluation

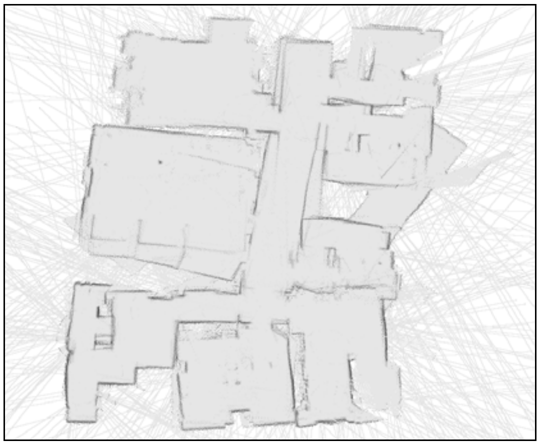


Figure 13 : map built by DPSLAM on real world acquisition "Arcueil laboratory n°1"

4.2.4 Posterior analysis

Our result with the "Arcueil laboratory n°1" sequence raises the issue of the dynamic models of the platform, which is a very important parameter in SLAM algorithms, and which is rarely easy to modify: we believe that one of the normalization actions that should be done about SLAM algorithms is to define precisely a standard format for the dynamic model of vehicles.

As we can see on Figure 13, we saw that the metric we defined didn't capture every aspect of the quality of a map. Actually, our metric don't penalize *artifacts* properly: the surface of the area of the produced map that has no counterpart in the real world should be measured and have an impact on the measure of quality.

Another issue brought by our experiments is the choice of the "control points". We tried to pick as many of them as we could, to keep to corners and distinctive features of the environment, and to have them sufficiently spread. However, a more systematic way to define the position of the control points could be used, especially if we want to achieve a certain degree of normalization. This could be associated with the automation of the control point designing process.

5. CONCLUSION

We have only presented a small set of our recent works on the assessment of the algorithms of unmanned autonomous systems; in the TAROT program the method has been systematically used and among the techniques we have evaluated there have been vision-based SLAM, obstacle avoidance algorithms, and vertical reference servoing.

The evaluation methods we formalize here already help us assess and compare robustness and performance of algorithms based on relevant quantitative information. The methodology we have briefly presented here was developed during several years and has evolved, taking incrementally into account the lessons learnt: using it systematically has allowed us to build databases of metrics, raw data, and evaluation results. We hope that soon we will be able to define testing protocols able to characterize the performance of most unmanned autonomous system algorithms, for both UGVs and UAVs.

We believe that the kind of work we do here is absolutely necessary in order one day to have certified unmanned autonomous systems. Initiatives like JAUS [22] identify standard messages and services, yet they do not precisely define the level of performance of the services. Nor do they define ways to compare the performance of the defined components; we believe it is a step that will need to be taken at one point.

Today, it is hard to evaluate whether or not two given components are compatible in a given architecture. Certification and clear definitions of level of autonomy will also become possible and with it an overall improvement of the performance of unmanned autonomous systems.

Furthermore, the capabilities of unmanned autonomous systems have not yet reached their full potential. They are most certainly useful algorithms that could be applied to this domain and that have not been discovered yet: having a "closed" methodology that would not allow incremental definition of new metric would therefore not be appropriate : the "open" methodology we propose, which capitalizes experience and data as more and more evaluations are made, is, we think, a good way to follow the dynamic field of unmanned autonomous systems.

6. ACKNOWLEDGMENTS

Our thanks to Jonathan Obriet-Leclef for his assistance in the SLAM experiments described here, to the people from ETAS (technical expertise center of Angers) for their help in the gathering of evaluation data and ground truth for the TAROT program, and to our partners within the TAROT program, whose feedback helps us increasingly improve our methodology, metrics and testing procedures. Thanks also go to Nick Roy for providing some of the data we used for our experiments on SLAM and to the anonymous reviewers for their useful comments.

7. REFERENCES

[1] Amigoni, F., Gasparini, S., Gini, M., « Good Experimental methodologies for robotic mapping: a proposal. » GEMBENCH 2008

[2] Balaguer, B., Carpin, S., Balakirsky, S., "Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems" , IROS 2008

[3] Collins, T., Collins, J.J., Ryan, C. "Occupancy Grid Mapping: An Empirical Evaluation" MED 2007

[4] Collins, R., Xuhui Zhou, Seng Keat The: «An Open Source Tracking Testbed and Evaluation Web Site », Robotics Institute, Carnegie Mellon University.

[5] D'Angelo, E., Herbin, S., Ratiéville, M., « ROBIN Challenge » : evaluation principles and metrics. http://robin.inrialpes.fr (2006).

[6] Dulac, N. "Real time, multisensor, advanced prototyping software. " First National Workshop on Control Architectures of Robots. Montpellier (France), 2006.

[7] Dufourd, D. and Dalgalarrondo, "Results and lessons learned from the quantitative evaluation of road detection and tracking algorithms.", Performance Metrics for Intelligent Systems Workshop (PerMIS'03). Gaithersburg (MD, USA), 2003.

[8] Eliazar, A., Parr, R. "DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks", IJCAI 2003.

[9] Fontana, G., Matteucci, M., Neira, J., Sorrenti, D. « Proposals for benchmarking SLAM », GEMBENCH 2008.

[10] Gerkey B.P., Vaughan, R.T., Howard, A., « The Player/Stage Project: Tools for Multi-Robot and Distributed Sensors Systems", ICAR 2003.

[11] Hui-Min H., Albus, J., Messina, E., « Toward a generic model for autonomy levels for unmanned systems (ALFUS) », PerMIS workshop, 2003.

[12] Jaulmes, R., Moliné, E. « HNG: A Robust Architecture for Mobile Robots Systems », EUROS 2008.

[13] Jaulmes R., Moliné, E., Obriet-Leclef, J. "Towards a quantitative evaluation of simultaneous localization and mapping methods", Control Architecture of Robots national conference, 2009.

[14] Lambert, M., Jaulmes, R., Godin, A., Moliné, E., Dufourd, D. "A methodology for assessing robot autonomous functionalities" , IAV 2007.

[15] Montemerlo, M., Thrun, S. « FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem », AAAI 2002.

[16] Scharstein, D., Szeliski, R. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.

[17] Thrun, S. "Robotic mapping : a survey", CMU-CS-02-111

[18] Wulf, O., Nuchter, A., Hertzberg, J. and Wagner, B. "Ground Truth Evaluation of Large Urban 6D SLAM", IROS 2007

[19] http://openslam.org/

[20] http://radish.sourceforge.net/

[21] http://www.rawseeds/org/ Website of the RAWSEEDS project.

[22] Http://www.jauswg.org/

[23] http://urban-challenge.com/

Performance Measures Framework for Unmanned Systems (PerMFUS)

Initial Perspective

Hui-Min Huang, Elena Messina, Adam Jacoff
National Institute of Standards and Technology (NIST)
100 Bureau Drive, MS 8230 Gaithersburg, Maryland 20899 USA
+1.301.975. {3427, 3510, 4235}
{hui-min.huang, elena.messina, adam.jacoff}@nist.gov

Abstract

The performance of intelligent unmanned systems (UMSs) must be able to be measured to ensure that they can meet the operational requirements. A generic framework to enable capturing and organizing the performance metrics is highly desirable. In this framework, unmanned system (UMS) performance can be attributed to the missions that it is commanded to perform, the environments that the missions are to be performed in, and the capabilities of the system itself. These attributes constitute a “three-axis” UMS performance metrics model. This framework, once further populated, can benefit the UMS community by allowing capture of the UMS performance, from the technical and operational perspective.

Categories and Subject Descriptors

J.2 [physical sciences and engineering] unmanned systems performance

General Terms

Measurement, Performance, Design, Human Factors, Standardization, Verification

Keywords

ALFUS, autonomy, communication, environment, goal, metrics, mission, mobility, robot, performance, sensor, terminology, test, unmanned system (UMS), urban search and rescue (US&R)

1. INTRODUCTION

UMSs have been deployed in many application domains, ranging from military, homeland security, manufacturing, medical, to general service applications. They either complement human operators to enhance mission performance or replace humans in dangerous or difficult situations (see Figure 1 and Figure 2). It is highly desirable that the UMS performance is able to be systematically and comprehensively measured to ensure that they can meet the operational requirements. The performance measures also facilitate understanding the UMS effectiveness, devising technological improvements, and inspiring innovations. A robot conducting an urban search and rescue (US&R) operation must be able to maneuver in confined space. An UMS conducting a perimeter surveillance operation of a large complex must have sufficient power and long-range communication capabilities.

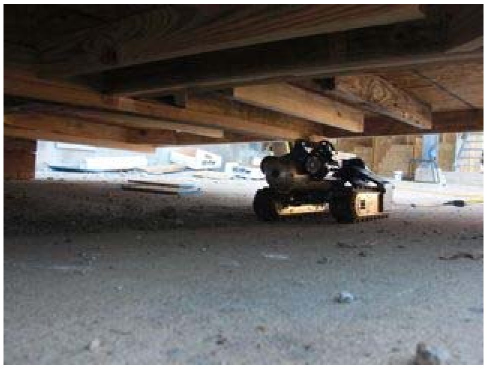


Figure 1: Robot performing mission in confined space

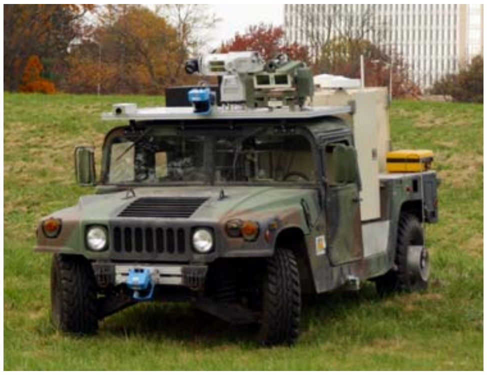


Figure 2: UMS equipped for operating in unstructured environments

The performance measurement is based, to a great extent, on the requirements. Requirements can be established with various kinds of documents such as Operational Requirements Documents (ORDs), Concepts of operation (CONOP), scenario descriptions, use case models, or operational vignettes [1]. Those documents can be generated with various degrees of formality and capture a range of levels of domain expert experience and observation. Operational requirements might cover UMS functional areas such as speed, stealthy maneuvering, explosive sensing, peak power, etc. Performance measures, on the other hand, deal with whether particular system design can meet the requirements, how to evaluate whether these requirements are met, and devising methods for the evaluation.

Performance issues have been dealt with in other non-robotic areas. In the software engineering discipline, metrics and requirement analysis have been studied. Traditionally, simple metrics such as lines of code, functional points, or cyclomatic complexity are used to measure the sizes of programs or their complexities in a context independent manner [2][3][4][5][6]. Metrics for measuring software quality, productivity, and reliability also exist [7][8][9]. Requirement analysis is also an area of interest. Giorgini P., et al., described a goal-oriented technique for requirement analysis. Goals are identified with domain stakeholders before being modeled. Detailed analysis correlates the system functionality to the goals [10].

Though these techniques can serve as references or be applied to particular, local metric issues, they lack a comprehensive system level approach.

It is extremely beneficial to have a general framework that establishes sets of metrics, describes an approach, and provides a set of guidelines to facilitate UMS performance measurement. The envisioned Performance Measure Framework for Unmanned Systems (PerMFUS) aims at serving these purposes. PerMFUS describes how one might organize and analyze the requirements, instantiate from the established generic metrics, generate additional program specific metrics, and devise methods to test and evaluate the UMS.

2. PERMFUS: OVERALL CONCEPTS

PerMFUS stems from the Autonomy Levels for Unmanned Systems (ALFUS) work [11][12]. As such, autonomy may be considered a specific metric under PerMFUS that has a wider scope. See Figure 3 and Figure 4 for a comparison. Earlier PerMFUS results involve metrics for mission goals [13].

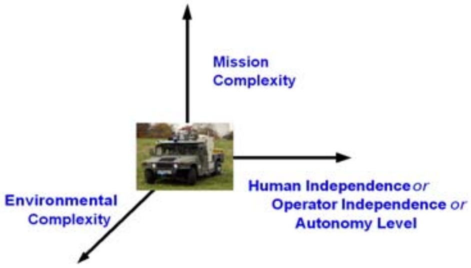


Figure 3: ALFUS overall concept

2.1 Three-Aspect Model

An UMS executes missions to accomplish the goals. Therefore, the key is to measure the UMS performance via its execution behaviors, in other words, how the system interacts with the environment and with humans while aiming to achieve the mission goals.

In this sense, the UMS performance can be attributed to the missions that it is commanded to perform, the environments that the missions are to be performed in, and the physical and logical capabilities of the system itself. These attributes constitute a "three- aspect" (or three- axis) UMS performance metrics model (Figure 4).

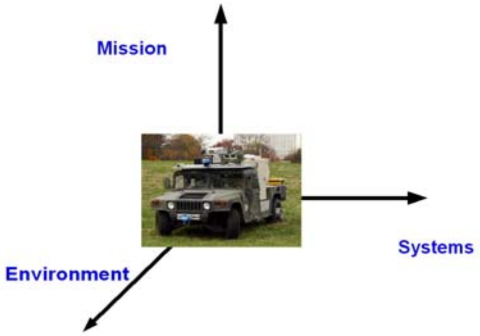


Figure 4: PerMFUS Main Aspects

Note that the three-axis graphic serves the purpose of highlighting the aspects that characterize the UMS performance. They do not represent that the aspects are independent from each other. Instead, they overlap.

Later sections will further elaborate on these axes.

2.2 Performance Characterization Parameters vs. Performance Metrics

Metrics are the parameters identified for measuring the performance. A metric can correspond to one or multiple system or environmental parameters. In other words, those system or environmental parameters can affect particular performance areas to different extents. For example, the performance metric might be how steep a slope an UMS can climb. The corresponding performance characterization parameters include weight, the center of gravity (which, itself, can be affected by the onboard payload or other configuration parameters), and the traction on the wheels or tracks.

The challenge is that the separation between the two aspects, system and environmental, might not always be clear. Particular system requirements can either specify the slope climbing performance or limit the weight. The requirements can either specify path planning performance or specify sensor ranges. For this reason, our current, first effort would be to identify all the performance characterization parameters.

2.3 Increasing Complexity

The axes convey increasing complexity starting from the origin. An example of primitive performance for an unmanned ground vehicle (UGV) (close to the origin) might be applying the mobility capability (wheels, for example) to a forward traverse task on a flat and paved surface. An example of higher level performance might be applying the mobility capability to an autonomous road following task. Yet an even higher performance might involve more complex missions such as emergency response, warfighting, or parts transfer on a manufacturing shop floor [32]. From the Framework development viewpoint, we take an bottom-up approach; the subsystem/component level is where the current focus is.

2.4 Performance Areas

Given that the performance is to be measured according to mission behaviors, PerMFUS decomposes a mission performance into a set of areas according to the UMS functions. Later sections will further elaborate these areas.

2.5 Testing and Evaluation

Besides the metrics and the performance characterization parameters, we envision that PerMFUS would also include testing and evaluation (T & E) methods. These methods would describe how to develop the T & E processes, how to set up the T & E environments, and how to conduct the T & E. The T& E methods would also guide how to instantiate these generic concepts and processes to specific program situations, including the metrics and the procedures.

2.6 Technical and Operational Aspects

We envision that PerMFUS will provide a comprehensive perspective of the UMS performance, technically and operationally. This framework, in its current scope, does not consider other aspects of the performance, such as program management or lifecycle costing.

3. PERFORMANCE AREAS

From the mission behavioral and execution perspective, the UMS performance can be divided into the following functional areas [26][27][28][29][30][31]:

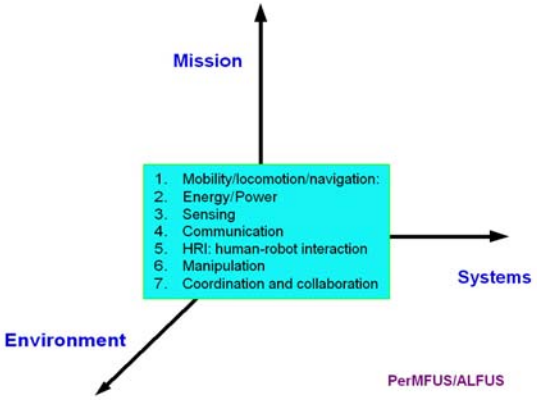

Mission
1. Mobility/locomotion/navigation:
2. Energy/Power
3. Sensing
4. Communication
5. HRI: human-robot interaction
6. Manipulation
7. Coordination and collaboration
Systems
Environment
PerMFUS/ALFUS

Figure 5: PerMFUS performance areas

1. Mobility/locomotion/navigation: the performance of traversing space (ground, air, water) to achieve the spatial and temporal goal. This performance area includes various levels of path planning when necessary. A path plan that takes shorter amount of execution time, avoids required adversaries, and spends a lower level of energy is a higher level of performance than another plan that costs more.
2. Energy/Power: to provide energy/power to enable all the other performance behaviors. Some tasks might require a very high level of peak power while some others might require long periods of steady levels of energy supplies. The performance involves whether and how the energy/power is supplied and managed.
3. Sensing and perception: the performance of onboard sensing to support UMS mission goals. The sensing and perception performance should cover many aspects of situational awareness (SA) such as identifying mobility obstacles, navigation paths that are energy efficient, and areas that might have low communication coverage. The respective UMS functions should, then, plan to respond according to the SA gained. Human speech intelligibility is an important performance area in the situation of human-robot interaction (HRI) [31]. Note that this performance area concerns sensing and perception for the basic UMS functions only. Mission types of sensing such as chemical, biological, radioactive, nuclear, and explosive (CBRNE) sensing is considered a separate performance area called Mission Package or Payload.
4. Communication: the performance of collaboration and information sharing among UMS subsystems, with other UMSs, or with the remote OCU, including transmitting maps or other types of information.
5. Human-Robot Interaction (HRI): A higher performing UMS may be when an operator is able to provide all the required interactions in a timely fashion. In another aspect, features such as easy to use, crash resistant, and responsive push buttons that support all required human-robot interactions may be indications of a higher performing OCU. See the ALFUS Framework for the established metrics [12].
6. Manipulation: UMSs often employ manipulator arm(s) for missions involving handling the environment, including objects and media (swimming). The types of manipulation include grasping, lifting, pushing, throwing, etc. The performance of the arm(s) should maximize working volume(s), strength, and dexterity, allow easy changing of grippers, etc.
7. Coordination and collaboration: interacting harmoniously toward goals either among the subsystem or among the UMS team members.
8. Mission Package or Payload: this performance area is application specific and, hence, is only identified here and will not be expanded until a later version of PerMFUS.

Figure 5 illustrates the concept. These performance areas will be further elaborated in the following sections.

An issue is whether the system internal processing capabilities (software and hardware) should be evaluated as part of the UMS performance. A system that is highly capable of processing the sensory data and generating information to support decision making should be likelier to have a higher level of performance. This argues for the world modeling and knowledge base as a part of performance evaluation. However, these capabilities should also be reflected in UMS actions. We must ensure the proper correspondences.

4. PERFORMANCE CHARACTERIZATION PARAMETERS ALONG THE SYSTEMS AXIS

The objective of this axis is to explore how the physical or logical properties of an UMS contribute or affect its performance. Both the hardware and software aspects are reviewed according to the aforementioned performance areas. The overall structure is shown in Figure 6 [14][15][16][17][18][19][20][21][22].

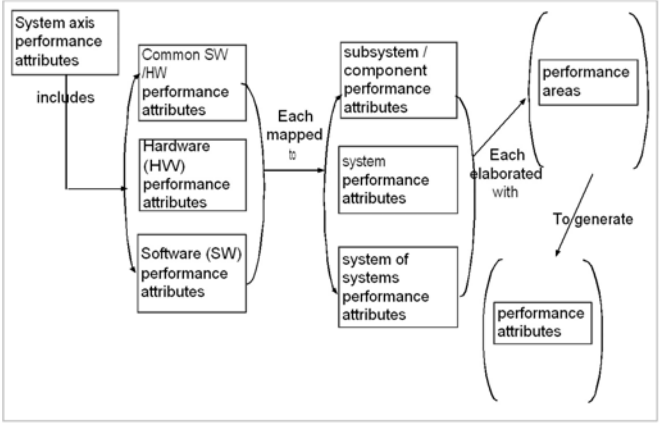

System axis performance attributes
includes
Common SW /HW performance attributes
Hardware (HW) performance attributes
Software (SW) performance attributes
Each mapped to
subsystem / component performance attributes
system performance attributes
system of systems performance attributes
Each elaborated with
performance areas
To generate
performance attributes

Figure 6: system axis metrics attribute structure

1. Common to software and hardware:

a. Configuration scalability, enable complex structure, easy integration

b. Seamless interoperability: standard interfaces vs. homogeneous system

c. Life expectancy, deterioration resistance to environmental conditions (corrosion) or software modifications.

d. Security

2. Hardware:

a. Subsystem/component level

i. Mobility performance: Wheel/track sizes/widths can determine whether particular terrain is traversable for a ground UMS.

ii. Energy/Power performance: The fuel tank or the equivalent can affect the system's endurance and is a key factor for mission planning.

iii. Communication: Signal strength, range, and bandwidth can determine whether particular mission-related areas are reachable.

iv. Sensing and perception: Sensor types, resolution, and range can help understanding particular mission areas. Interface types or plug-and-play ability can affect whether a mission can succeed.

v. HRI: Numbers and types of the human input/output (I/O) mechanisms and their responsiveness on an OCU can be important to the operator executing the mission.

vi. Manipulation: Size/weight of the target that can be grasped are performance factors.

vii. Coordination and collaboration: spacing among the subsystems allowing physical interactions.

b. System level

i. Mobility performance: System level physical characteristics including total weight, overall dimensions, turning radius, distances among wheels, speed ratings, etc., are all performance factors.

ii. Energy/Power performance factors: replenish time, system maximum power output.

iii. Communication: coverage area and the associated strength profile

iv. Sensing and perception: coverage area and resolution profile

v. HRI: usability of the OCU

vi. Manipulation: profile of grasping dexterity throughout work volume

vii. Coordination and collaboration: spacing among the systems allowing physical interactions

c. Team level

i. Mobility performance: a team of UMSs to march in an optimal formation

ii. Energy/Power performance factors: team endurance time to achieve a common goal

iii. Communication: team coverage area, strength profile

iv. Sensing and perception: team coverage area, resolution profile

v. Manipulation: inter-UMS coordinated reach/work volume; e.g., two UMSs grasping a large object together

vi. HRI: team usability

vii. Coordination and collaboration: spacing among the teams allowing physical interactions

3. Software Architecture [25]

    a. Common to all performance areas:

        i. perception: multiple layers of abstraction for knowledge or intelligence

        ii. control: open vs. closed loop; on/off vs. continuous; central vs. distributed control; planning and coordination

        iii. communication: contextual or transport layers

        iv. integration of heterogeneous software elements

        v. enable goal adjustment and replanning in real-time

        vi. responsiveness, real-timeliness

        vii. enabling complexity: functional points can be a useful metric for generally measuring the size of computer software. It can also be used to reflect the scope.

    b. Subsystem, system, and team levels:

        i. Mobility and navigation performance: accommodate multiple, different scales.

        ii. Energy/Power performance: priority based management strategy

        iii. Communication: latency, dynamic/static

        iv. Sensing and perception: from raw data to multiple layers of information process enabling decision making;

        v. HRI: allowing for multiple modalities of human interactions

        vi. Manipulation: joint vs. coordinated control based on advanced sensing, singularity resistant

        vii. Coordination and collaboration: allowing for multiple types of UMS interactions; data sharing effectiveness and efficiency among subsystems/components or among UMSs.

Note that, for an existing UMS, the characterization of its capability can facilitating understanding on how missions may be planned and executed; whether they can be done and how effectively and efficiently they can be accomplished. For new system acquisition, the system performance characteristics can help specifying the capability.

5. PERFORMANCE CHARACTERIZATION PARAMETERS ALONG THE ENVIRONMENT AXIS

In this axis, we seek to characterize UMS operating environments and determine how they might affect UMS performance.

1. Media

    a. Type(s): air/ground/surface/underwater and the number of the domains that is/are involved; ground robots encountering streams, etc.

    b. Uniformity: paved/dirt/grass for ground systems

    c. Density: fresh water/sea with various levels of salinity; clear/misty sky; density of bushes

    d. Continuity: gap/pot hole/rock/man-made structure

    e. Dimensions: two, two and a half, and three: slope, steps, rolling field, confined space under collapsed structure for ground; no-flight zone for air

    f. Dynamicity: frequency and scope of changes ; wind direction and speed

2. Anomaly/obstacle

    a. Discrete: rock, tree, river

        i. sizes

        ii. numbers

        iii. types

        iv. dynamicity: frequency and scope of changes

        v. adversity severity

    b. continuous: fog, rain, electro-magnetic interference, maze

        i. density

        ii. dynamicity; frequency and scope of changes

        iii. adversity severity

The environmental characteristics can be applied to analyze the UMS performance areas, as described in the following:

1. Mobility: traversability

2. Fuel/Power: power requirements, energy consumption

3. Communication: radio interference, multipath interference, scattering, and attenuation can all be caused by the environmental factors.

4. Sensing and perception:

    a. Feature identification: correctness, able to track its dynamics

    b. quality of maps—coverage area size, resolution, completeness, coordinate accuracy throughout map, correctness or mis-identification of features, update-able/real-timeliness [23][24]

5. HRI: the interaction time, levels of effort (workload), percentage of task execution, and initiation can all affected by the types of the environments.

6. Manipulation: types of objects that can be handled to help goal achievement

7. Coordination and collaboration: air and ground collaboration; ground team to search a commanded area

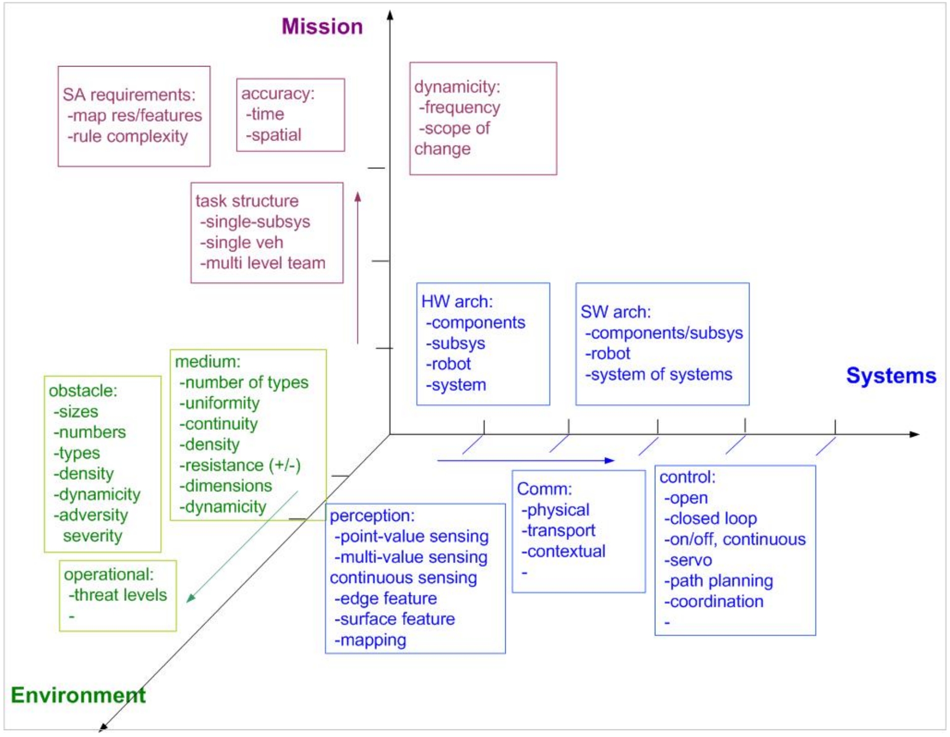


Figure 7: overall illustration of PerMFUS perspective

6. PERFORMANCE CHARACTERIZATION PARAMETERS ALONG THE MISSION AXIS

UMSs execute missions to achieve their goals. Therefore, the primary common metrics are whether and how well the goals are achieved. The following provides a list of metrics:

1. Accuracy: goal state in time, space, and logically; tolerance
2. Efficiency: time, costs
3. Effectiveness: % completeness
4. Reliability: % of trials accomplishing goal
5. Autonomy
6. Safety
7. Handling complexity

The mission metrics can be applied to the performance areas for detailed analysis by the particular programs. The metrics are applied to the particular UMS performance areas.

1. Mobility: accuracy of own position;
2. Energy/Power: power requirements, energy consumption
3. Communication: allowed sizes of map to be transmitted
4. Sensing and perception: correctness and resolution; able to perceive dynamic situations
5. HRI: complexity of HRI rules
6. Manipulation: reliably grasping required complex objects; pushing open a gate to enable navigation
7. Coordination and collaboration: complex task structure: number of involving subtasks, numbers of involving subsystem, numbers of involving vehicles, numbers of involving teams.

7. COMPREHENSIVE VIEW

Figure 7 illustrates a comprehensive concept of the metric framework. Metrics are categorized along the three axes.

Note that, UMSs may have additional performance areas. A wide variety of mission package(s) exists, such as CBRNE detection, lethality, RSTA (reconnaissance, surveillance, and target acquisition), etc. For these additional areas, the performance can be characterized using the same structure.

8. TEST AND EVALUATION ISSUES

To properly apply the metric sets, it is imperative to collect and organize the requirements. As stated in the earlier sections, the requirements can be in the formats of CONOPs, vignettes, scenarios, use cases, etc. The organization of the requirements may be the decomposition from high level requirements to series of low level but detailed sets.

Test and evaluation methods may start from the low level requirements. The test methods include such elements as metrics, tasks or procedures, equipment or personnel involved, setups or apparatuses, safety guidelines, and accuracy statements.

Beyond those elemental tests, high level, scenario based tests should also be designed to evaluate the UMS integrated performance.

Certain considerations must be given during the design of the test and evaluation design. They include:

- User requirements may or may not have one-to-one correspondence with testing and evaluation metrics. The former is described in the operational domain whereas the latter in the technical domain. The objective should be to identify a collection of metrics to sufficiently evaluate for a particular requirement.

- Systems are developed to meet particular requirements, therefore, the metrics may be weighted.
- The more details and the more specific the requirements can be decomposed into, the easier it is to develop the test and evaluation methods. It is desirable that an elemental test method can be executed with a single task, which, in turns, can minimize correlation with outside concerns.
- The more critical the designed missions are, the higher degrees of rigor the test methods should have. This may entail more detailed metrics and testing tasks, higher levels of precision for the testing setups and apparatuses, or higher numbers of test runs for obtaining higher levels of confidence of the test results.

These issues must be further examined and specified in the framework.

9. A USE CASE

Table 1 is a sample set of requirements for emergency response robots, particularly, applicable to a collapsed structure. The left column describes the operational requirements as stated by the would-be users of the robots, the emergency responders. The right column shows the corresponding performance areas.

For an example, for the Requirement A, tumble recovery, a set of test methods must be devised that are representative of the terrain types in emergency situations. The testing terrains must possess different degrees of complexity so that robots with various mobility capabilities can be evaluated for the tumble recovery feature.

For Requirement C, test methods must be designed to evaluate that the video link allows displaying the scenes of the confined space in sufficient resolutions and with sufficiently wide field-of-view. These correspond to the metrics described in the earlier sections.

In addition, to correspond to the Mobility-confined space performance area, the testing apparatuses must include confined space with various kinds of testing metrics such as gaps with various sizes, slopes with various degrees, different types of stairs, floor conditions of wetness, slipperiness, containing hard-to-detect trip wires, etc.

For Requirement E, test methods must be designed to evaluate the robot's capability to traverse a long distance and to communicate with its onboard radio or tether-based communication system.

Requirements	**Performance Areas**
A. Tumble recovery within Terrain Type	Mobility
B. Maintain operations beyond basic mobility requirements within a given terrain type. The system must have sufficient power to operate for the specified number of hours, assuming one power charge for one out and back mission.	Energy/Power

C. To project remote situational awareness into compromised or collapsed structures or to convey other types of information. To be able to ingress a specified number of meters into the worst case collapse, which was further defined as a reinforced steel structure. To operate around corners of buildings and other locations beyond line of sight.	Sensing-video;
	Mobility-confined space;
	Communication-NLOS;

D. To enable use of video in confined spaces and for short-range object identification, which can wash out from excessive illumination of the scene; therefore, adjustability is required	Mission package-variable illumination;
	Communication-NLOS;

E. To project remote situational awareness or to convey other types of information down range within line of sight	Sensing-video;
	Energy/Power-endurance
	Communication-LOS, long range;

Table 1: Requirement Statement Samples for Emergency Response Robots

10. SUMMARY

An initial perspective of the performance measures framework for unmanned systems, PerMFUS, is described. In such a framework, performance metrics are characterized from the three aspects, the system (software and hardware), the operating environment, and mission. They are further elaborated according to a set of performance areas. The combination aims at a comprehensive structure and sets of metrics. In addition, an approach is described for applying the metrics to the testing and evaluation process.

A lot of the identified areas contain exemple metrics, which must be systematically expanded. Much more effort is planned to further develop PerMFUS.

ACKNOWLEDGEMENT

NIST Response robotics project team and the associated users/developers forum shared hands-on implementation experience. SAE AS4-D membership participated in the Framework development work.

REFERENCES

[1] Messina, E. R., et al., Statement of Requirements for Urban Search and Rescue Robot Performance Standards, NIST Draft Report, May 2005

[2] Albrecht, A. J. and J. E. Gaffney, Jr. "Software Functions, Source Lines of Code, and Development Effort Prediction: A Software Science Validation." *IEEE Trans. Software Eng. SE-9*, 6, Nov. 1983

[3] McCabe, T. J. "A Complexity Measure." *IEEE Trans. Software Eng. SE-2*, 4 Dec. 1976

[4] U.S. Army, Practical Software and System Measurement: A Foundation for Objective Project Management, Version 4.0b, U.S. Army, Dept. of Defense, Washington DC, 2000

[5] Software Engineering Institute (1992), Software Measurement for DoD Systems: Recommendations for Initial Core Measures, CMU/SEI-92-TR-19

[6] Software Engineering Institute, Software Metrics, SEI Curriculum Module SEI-CM-12-1.1, Pittsburg, Pennsylvania, Dec. 1988

[7] IEEE Standard 1061-1998, IEEE Standard for Software Quality Metrics Methodology

[8] IEEE Standard 1045-1992, IEEE Standard for Software Productivity Metrics

[9] IEEE Standard 982.1-1988 and 982.2-1998, IEEE Standard for Dictionary of Measures to Produce Reliable Software

[10] Giorgini, P., et al., "Goal-Oriented Requirement Analysis for Data Warehouse Design," Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, 2005

[11] *Autonomy Levels for Unmanned* Systems *Framework, Volume I: Terminology, Version 2.0*, Huang, H. Ed., NIST Special Publication 1011-I-2.0, National Institute of Standards and Technology, Gaithersburg, MD, Oct. 2008

[12] *Autonomy Levels for Unmanned Systems (ALFUS) Framework, Volume II: Framework Models Version 1.0, NIST Special Publication 1011-II-1.0,* Huang, H. et al., Ed., National Institute of Standards and Technology, Gaithersburg, MD, December 2007

[13] Hui-Min Huang, et al., "Ontological Perspectives for Autonomy Performance" PerMIS'08 Workshop, Gaithersburg, Maryland, October 2008

[14] MIL-Std-3014, Department of Defense Interface Standard for Mission Data Exchange Format, 2004

[15] SAE Aerospace Information Report AIR5664, JAUS History and Domain Model, 2006

[16] SAE Aerospace Information Report AIR5665A, Architecture Framework for Unmanned Systems, 2009

[17] SAE Aerospace Recommended Practice APR6012, JAUS Compliance and Interoperability Policy, 2009

[18] SAE Aerospace Information Report AIR5645, JAUS Transport Considerations, 2007

[19] SAE Aerospace Standard AS5669A, JAUS/SDP Transport Specification, 2009

[20] SAE Aerospace Standard AS5684, JAUS Service Interface Definition Language, 2008

[21] SAE Aerospace Standard AS5710, JAUS Core Service Set, 2008

[22] SAE Aerospace Standard AS6009, JAUS Mobility Service Set, 2009

[23] Thrun, S., Learning Metric-Topological Maps for Indoor Mobile Robot Navigation. Artificial Intelligence, pages 21–71, February 1998

[24] I. Varsadan, A. Birk, and M. Pfingsthorn. Determining Map Quality through an Image Similarity Metric. In Lecture Notes in Artificial Intelligence (LNAI): RoboCup 2008: Robot WorldCup XII, 2008

[25] Albus, J., et al., 4D/RCS: A Reference Model Architecture For Unmanned Vehicle Systems Version 2.0, NISTIR 6910, 2002

[26] Messina, E. R. and Jacoff, A. S., **"**Measuring the Performance of Urban Search and Rescue Robots," Proceedings of the IEEE Conference on Homeland Security Technologies, 2007

[27] Jacoff, A. S. and Messina, E. R., "Urban Search and Rescue Robot Performance Standards: Progress Update," SPIE Defense and Security Conference, 2007

[28] Remley, K. A., et al., "Standards Development for Wireless Communications for Urban Search and Rescue Robots," International Symposium on Advanced Radio Technologies (ISART) 2007

[29] Molino, V., et al., "Traversability Metrics for Urban Search and Rescue Robots on Rough Terrain," Performance Metrics for Intelligent Systems (PerMIS) Proceedings, 2006

[30] Shneier, M. O., et al., "Performance Evaluation of a Terrain Traversability Learning Algorithm in The DARPA LAGR Program," Performance Metrics for Intelligent Systems (PerMIS) Workshop 2006

[31] ANSI S3.5 1997 American National Standard Methods for Calculation of The Speech Intelligibility Index

[32] Weiss, B. A. and Schlenoff, C. I., "Evolution of the SCORE Framework to Enhance Field-Based Performance Evaluations of Emerging Technologies," Performance Metrics for Intelligent Systems (PerMIS) Workshop 2008

Optimum Combination of Full System and Subsystem Tests for Estimating the Reliability of a System

Coire J. Maranzano
The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Rd
Laurel, Maryland, 20723-6099, USA
coire.maranzano@jhuapl.edu

James C. Spall
The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Rd
Laurel, Maryland, 20723-6099, USA
james.spall@jhuapl.edu

ABSTRACT

This paper develops a method for finding an optimum test plan, which consists of a mixture of full system and subsystem tests, to estimate the reliability of a system. An optimum test plan is developed by trading off the number of full system and subsystem tests to minimize the mean-squared error (MSE) of the maximum likelihood estimate (MLE) of system reliability and testing costs. The MSE is decomposed into the variance of the MLE and a bias from incorrectly specifying the function that relates the subsystem reliabilities to the full system reliability (series, parallel, other). The variance of the MLE comes from Fisher theory. The bias is due to the modeling error. Optimum test plans involve trade offs between the MSE (estimation accuracy), the degree of modeling error, and the cost of doing system and subsystem tests. A Pareto frontier can be identified, as illustrated in the paper.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]; J.2 [**Physical sciences and engineering**]

Keywords

Reliability, Test Sizing, Model Bias, Mean-Squared Error, Maximum Likelihood Estimation

1. INTRODUCTION

System, subsystem, component, interface, and other[1] tests are often carried out on complex systems to ensure that an operational reliability requirement is satisfied. Fusing full system and subsystem test data to evaluate the reliability of a complex systems is desirable when full system testing can be very costly, dangerous or requires the destruction of the system itself. Additionally, it is desirable to include full system testing in an overall reliability assessment to help guard against possible mis-modeling of the relationship between the subsystems and full system in calculating overall system reliability. One method of fusing full system and subsystem reliability test data to form a full system estimate of reliability is the method of maximum likelihood [9]. This general maximum likelihood formulation for the fusion of reliability test data applies across all system configurations (series, parallel, etc); only the optimization constraints change, leading to an appropriate maximum likelihood estimate (MLE). The MLE method provides a characterization of the estimation uncertainty—statistical uncertainty about the model parameters—through the Fisher Information on the parameters of the system reliability model. If the reliability of the system must be known within a specified confidence interval or if the test plan is limited by cost, there exists an inherent trade off between performing full system tests or subsystem tests. This paper develops a method for finding an optimal test plan consisting of a mixture of full system and subsystem tests, to estimate the full system reliability.

A necessary step in the process of developing an optimal test plan is to quantify the uncertainty in the estimate of full system reliability. Uncertainty arises from randomness in the data and lack of knowledge of the model for the system reliability. Randomness in the data results in uncertainty in the parameter estimates. Lack of knowledge of the model arises from incomplete understanding of the system and is called model uncertainty. A formal theory for quantifying estimation uncertainty exists within general statistical theory (specifically, within the framework of general maximum likelihood estimation through the asymptotic distribution of the parameter estimates [7]). In general, a formal theory for quantifying model uncertainty does not exist. Significant uncertainty results from how well the model for the system actually represents the system's true behavior. Ref. [2] identifies six major characteristics of modeling errors that lead to model uncertainty: model topology, model parameters, model scope or focus, data, optimization technique, and human subjectivity. Ref. [2] goes on to point out that a modeling error exhibiting at least one of the aforementioned characteristics can potentially contribute the most uncertainty to a quantitative prediction, and its reduction is not

[1] To avoid the need to repeatedly refer to tests on subsystems, components, processes, and other aspects of the system as the key source of information other than full system tests, we will usually only refer to subsystem tests; subsystem tests in this context should be considered a proxy for all possible test information short of full system tests.

PerMIS'09, September 21–23, 2009, Gaithersburg, MD, USA
Copyright 2009 ACM 978-1-60558-747-9/09/09 ...$10.00.

straitforward or simple.

Theoretical frameworks to quantify the model uncertainty typically rely on being able to observe and record the model error [4, 5]. Such an approach can be applied to reliability estimation, though, for complex systems it is often not practical because the system may be so unique that no prior model error analysis is valid and/or there may be insufficient tests to observe model error. Alternatively, a Bayesian approach can be taken to incorporate model uncertainty in a reliability estimate by "us[ing] a weighted average of all possible models" [11]. This, too, is often not practical because the analysis must specify all the possible reliability models and provide a prior probability for each model. Also, the approach is dangerous because, although it is possible to state that one model is better than another, it is not possible to state that one model is more probable than another [3, 6].

The purpose of this paper is to identify an optimal test plan by trading off between subsystem testing and full system testing, within a general maximum likelihood reliability estimation paradigm, to minimize estimation uncertainty, the effect of modeling error, and the cost of testing. The general maximum likelihood method of reliability estimation described in [9], fuses data from subsystem tests and full system tests via a model that reflects the constraints associated with the operation of the full system. The quality of the MLE is assessed via the mean squared error (MSE) of the estimate, this leads to a decomposition involving the variance of the estimate and the bias of the estimate. In particular, the MLE of system reliability is decomposed into the modeling bias from incorrectly specifying the function that relates the subsystem reliabilities to the full system reliability (series, parallel, other) and the asymptotic variance of the model parameters. Modeling error contributes to the bias because an error in the model results in a biased full system reliability estimate even if the parameters of the model are known perfectly. An optimal test plan is developed by trading off the number of full system and subsystem tests to minimize the MSE of the MLE of system reliability and testing costs.

Section 2 of this paper presents the general MSE formula for a system reliability estimate in terms of the system reliability model and parameter estimates. In Section 3, the MSE of the maximum likelihood estimator of system reliability based on full system tests is compared with the maximum likelihood estimator based on subsystem tests for a series system. Assuming no model uncertainty, the comparison reveals that, for a truly series system, performing a set of subsystem tests (one for each subsystem) always reduces the variance of the full system reliability estimate more than performing a full system test. Section 4 establishes the MSE for the general maximum likelihood estimator of system reliability due to [9], given a maximum modeling error. In Section 5, the MSE for the general maximum likelihood estimator of system reliability is computed for a hypothetical system and the results are used to determine optimal test plans in terms of the MSE and testing costs.

2. GENERAL MEAN SQUARED ERROR FORMULATION

Consider a system composed of p subsystems, typically a system process and/or components of subsystems. The general estimation formulation involves a parameter vector $\boldsymbol{\theta}$, representing the parameters to be estimated. Let ρ and ρ_j represent the reliabilities (success probabilities) for the full system and for subsystem j, respectively, $j = 1, 2, \ldots, p$. The vector $\boldsymbol{\theta} = [\rho_1, \rho_2, \ldots, \rho_p]^T$. Let $\boldsymbol{\Theta}$ represent the feasible region for the elements of $\boldsymbol{\theta}$. To ensure that relevant logarithms are defined and that the appropriate derivatives exist, it is assumed, at a minimum, that the feasible region $\boldsymbol{\Theta}$ includes the restriction that $0 < \rho_j < 1$ for all j. The system reliability ρ is not included in $\boldsymbol{\theta}$ because it is uniquely determined (or bounded) by the subsystem reliabilities ρ_j for $j = 1, 2, \ldots, p$ and possibly other information via relevant constraints. Herein, the relation is restricted such that ρ is uniquely determined by a function $h(\boldsymbol{\theta})$. The mapping, h, between $\boldsymbol{\theta}$ and ρ dictates the arrangement of the system, which may be configured in series, parallel, combination series/parallel, or some other configuration, and it is analogous to a *model* of system reliability in terms of its subsystems. To mirror the commonly used lexicon in the literature, h will be referred to as the model for the system reliability.

Thus, an estimate of the system reliability $\hat{\rho}$ is found by evaluating $h(\cdot)$ at the estimate $\hat{\boldsymbol{\theta}}$. Further consider the test data on the system and its subsystems. Let Y be the number of successes in n independent identically distributed (i.i.d.) tests of the system, and let X_j be the number of successes in n_j i.i.d. tests of the j^{th} subsystem, for $j = 1, \ldots, p$. And, let $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\boldsymbol{Y})$ be a function that produces an estimate of $\boldsymbol{\theta}$, where $\boldsymbol{Y}$ is the set of test data on the system and its subsystems $\{X, X_1, \ldots, X_p\}$.

A measure of the effectiveness of an estimator is the mean squared error. The MSE for an estimate of system reliability, given the model, h, and the data, $\boldsymbol{Y}$, is $E\left[\left(h\left(\hat{\boldsymbol{\theta}}(\boldsymbol{Y})\right) - \rho\right)^2\right]$, where the expectation is computed with respect to the random variable $\hat{\boldsymbol{\theta}}$. Ref. [8, Chapter 13] shows that the MSE consists of two terms,

$$E\left[\left(h\left(\hat{\boldsymbol{\theta}}(\boldsymbol{Y})\right) - \rho\right)^2\right] = \underbrace{E\left[\left(h\left(\hat{\boldsymbol{\theta}}(\boldsymbol{Y})\right) - E\left[h\left(\hat{\boldsymbol{\theta}}(\boldsymbol{Y})\right)\right]\right)^2\right]}_{\text{Variance of Estimate}} + \underbrace{\left(E\left[h\left(\hat{\boldsymbol{\theta}}(\boldsymbol{Y})\right)\right] - \rho\right)^2}_{\text{Bias of Estimator}^2}, \tag{1}$$

that are the variance of the estimate (a measure of estimate uncertainty) and the bias of the estimate. The bias term of the MSE is zero if the estimator is unbiased, that is $E\left[h\left(\hat{\boldsymbol{\theta}}(\boldsymbol{Y})\right)\right] = \rho$.

Eqn. (1) forms the basis for evaluating estimators of the system reliability given the model for the system and the test data. As such, it can be used to determine the number of system and subsystem tests needed to achieve a minimum MSE estimate given the presumed system reliability (specifically, the presumed value of $\boldsymbol{\theta}$). In other words, Eqn. (1) is used as a criterion for determining optimal test plans. In Section 5, optimal test plans for maximum likelihood reliability estimation are found by selecting the combination of system and subsystem tests that minimize the MSE (variance and bias) and testing costs.

3. MSE FOR A SERIES SYSTEM WITHOUT MODEL UNCERTAINTY

Consider estimating the reliability for a fully series system comprised of stochastically independent subsystems by performing only full system tests or only tests of the subsystems using maximum likelihood estimation (and assuming that the model for the system is perfect). In this case, the system reliability model is $h(\boldsymbol{\theta}) = \prod_{i=1}^{p} \rho_i$. An optimal test plan can be found by selecting the combination of tests—tests of the full system or tests of each subsystem—that minimizes the MSE of the estimate of system reliability.

3.1 Variance of System Reliability Estimates

Given only tests of the full system, the MSE of the maximum likelihood estimator of full system reliability, $\hat{\hat{\rho}} \equiv X/n$, is the variance of the estimate,

$$\text{var}\left(\hat{\hat{\rho}}\right) = \frac{\left(\prod_{i=1}^{p} \rho_i\right)\left(1 - \prod_{j=1}^{p} \rho_i\right)}{n}, \tag{2}$$

since the estimator is unbiased. Given only tests of the subsystems, the maximum likelihood estimate of system reliability for a series system is the product of the subsystem reliability estimates. Let $\hat{\rho}_i = X_i/n_i$ for $i = 1, \ldots, p$ be the the maximum likelihood estimates of the subsystem reliabilities. Then, the maximum likelihood estimate of the system reliability ρ is $\hat{\rho} = h(\hat{\boldsymbol{\theta}}) = \prod_{i=1}^{p} \hat{\rho}_i$. This estimator is unbiased, and so its variance, $\text{var}\left(\prod_{i=1}^{p} \hat{\rho}_i\right)$, is the only contributor to the MSE.

The variance of $\prod_{i=1}^{p} \hat{\rho}_i$ is found by computing a related variance: the variance of the product of binomial success. Let X_i for $i = 1, \ldots, p$ be a binomial random variable that is the number of successes in n_i independent Bernoulli trials with probability ρ_i of success on each trial. Then, the variance of the product of binomial random variables is

$$\begin{aligned}
\text{var}\left(\prod_{i=1}^{p} X_i\right) &= \prod_{i=1}^{p} E\left(X_i^2\right) - \prod_{i=1}^{p} \left[E\left(X_i\right)\right]^2, \\
&= \left(\prod_{i=1}^{p} n_i\rho_i\left[\left(n_i - 1\right)\rho_i + 1\right]\right) - \left(\prod_{i=1}^{p} n_i\rho_i\right)^2, \\
&= \left(\prod_{i=1}^{p} n_i\rho_i\right)\left[\left(\prod_{i=1}^{p}\left[\left(n_i - 1\right)\rho_i + 1\right]\right) - \prod_{i=1}^{p} n_i\rho_i\right],
\end{aligned} \tag{3}$$

and the variance of the product of maximum likelihood estimates is,

$$\begin{aligned}
\text{var}\left(\prod_{i=1}^{p} \hat{\rho}_i\right) &= \text{var}\left(\prod_{i=1}^{p} \frac{X_i}{n_i}\right), \\
&= \prod_{i=1}^{p} \frac{1}{n_i^2}\left(\text{var}\left(\prod_{i=1}^{p} X_i\right)\right), \\
&= \left(\prod_{i=1}^{p} \frac{\rho_i}{n_i}\right)\left[\left(\prod_{i=1}^{p}\left[\left(n_i - 1\right)\rho_i + 1\right]\right) - \prod_{i=1}^{p} n_i\rho_i\right].
\end{aligned} \tag{4}$$

Given Eqn. (2) and Eqn. (4), the variance of the maximum likelihood reliability estimates can be compared to determine the type of test, a full system test or a set of subsystems tests, that produces the largest reduction in the variance of the full system reliability MLE (equivalently the MSE). The comparison reveals that, when the model for system reliability is a set of p subsystems in series and there is no modeling error, performing a set of subsystem tests (one for each subsystem) always reduces the variance (equivalently, MSE) more than a single full system test. The next section is devoted to a formal proof. And so, for a series system, assuming no modeling error, choosing to perform sets of subsystem tests instead of full system tests yields the minimum MSE test plan. Also, if the cost of a set of subsystem tests is less than a full system test, it is also the minimum cost test plan. As shown in the next subsection, these two conclusions, which prescribe an optimum test plan for a series system, are weakest when the true reliability of the system is very close to 1 or 0.

3.2 Comparison the Variance of Series System Reliability Estimates

Theorem 1 presents the conditions that relate $\text{var}\left(\hat{\hat{\rho}}\right)$ and $\text{var}\left(\prod_{i=1}^{p} \hat{\rho}_i\right)$.

Theorem 1. *Let ρ be the probability of full system success where $\rho = \prod_{i=1}^{p} \rho_i$. Assuming that the number of trials for the full system, n, is the same as for each subsystem, that is $n = n_1 = \cdots = n_p$, then for $n \geq 2, p \geq 2$, and $0 < \rho_i < 1$ for $i = 1, \ldots p$,*

$$\text{var}\left(\hat{\hat{\rho}}\right) > \text{var}\left(\prod_{i=1}^{p} \hat{\rho}_i\right). \tag{5}$$

Substituting Eqn. (2) and Eqn. (4) with $n = n_1 = \cdots = n_p$ into Eqn. (5) yields an equivalent inequality to consider:

$$\frac{\prod_{i=1}^{p} \rho_i\left(1 - \prod_{i=1}^{p} \rho_i\right)}{n} > \prod_{i=1}^{p} \frac{\rho_i}{n}\left[\prod_{i=1}^{p}\left[\left(n - 1\right)\rho_i + 1\right] - \prod_{i=1}^{p} n\rho_i\right]. \tag{6}$$

This statement can be simplified algebraically to yield a statement equivalent to Eqn. (5). First, divide each side of Eqn. (6) by $\prod_{i=1}^{p} \rho_i/n$ to obtain,

$$1 - \prod_{i=1}^{p} \rho_i > \frac{1}{n^{p-1}}\left[\prod_{i=1}^{p}\left[\left(n - 1\right)\rho_i + 1\right] - \prod_{i=1}^{p} n\rho_i\right], \tag{7}$$

and then, multiply each side by n^{p-1} and add to each side $\prod_{i=1}^{p} n\rho_i$ to obtain,

$$n^{p-1}\left(\left(n - 1\right)\prod_{i=1}^{p} \rho_i + 1\right) > \prod_{i=1}^{p}\left[\left(n - 1\right)\rho_i + 1\right]. \tag{8}$$

And so, proving that the inequality in Eqn. (8) is true for $n \geq 2, p \geq 2$, and $0 < \rho_i < 1$ for $i = 1, \ldots p$ is equivalent to proving that Theorem 1 is true. Proof proceeds by induction on p.

Proof. The base case is $p = 2$. For probabilities $0 < \rho_1 < 1$ and $0 < \rho_2 < 1$,

$$\rho_1\rho_2 + 1 > \rho_1 + \rho_2, \tag{9}$$

because $(\rho_1 - 1)(\rho_2 - 1) > 0$. Manipulating Eqn. (9) algebraically yields an inequality, which is then also true when $n \geq 2$,

$$\begin{aligned}
&(n - 1)^2\rho_1\rho_2 + (n - 1)\left(\rho_1\rho_2 + 1\right) + 1 > \\
&\qquad (n - 1)^2\rho_1\rho_2 + (n - 1)\left(\rho_1 + \rho_2\right) + 1.
\end{aligned} \tag{10}$$

After factoring the left side of Eqn. (10) and recognizing that the right side of Eqn. (10) is a product, it is easy to see that the following inequality is true,

$$n\left((n-1)\rho_1\rho_2+1\right) > \left[(n-1)\rho_1+1\right]\left[(n-1)\rho_2+1\right], \tag{11}$$

and so, for $p=2$, the inequality in Eqn. (8) is true, and Theorem 1 holds.

For the inductive step, assume that Theorem 1 is true for some $p \geq 2$. This is equivalent to assuming that,

$$n^{p-1}\left((n-1)\prod_{i=1}^{p}\rho_i+1\right) > \prod_{i=1}^{p}[(n-1)\rho_i+1], \tag{12}$$

is true for $n \geq 2$ and $0 < \rho_i < 1$ for $i = 1, \ldots p$. The next step in the inductive proof is to show that Theorem 1 holds for $p+1$. Proceeding from Eqn. (12), for $n \geq 2$ and for $0 < \rho_{p+1} < 1$, it follows that,

$$n\rho_{p+1}n^{(p-1)}\left((n-1)\prod_{i=1}^{p}\rho_i+1\right) > n\rho_{p+1}\prod_{i=1}^{p}[(n-1)\rho_i+1]. \tag{13}$$

Given that, for $n \geq 2$ and $0 < p_i < 1$ for $i = 1, \ldots p$,

$$n^p > \prod_{i=1}^{p}[(n-1)p_i+1], \tag{14}$$

and, given the inequality in Eqn. (13), the following holds,

$$n\rho_{p+1}n^{(p-1)}\left((n-1)\prod_{i=1}^{p}\rho_i+1\right)+n^p\left(1-\rho_{p+1}\right) >$$
$$n\rho_{p+1}\prod_{i=1}^{p}[(n-1)\rho_i+1]+\prod_{i=1}^{p}[(n-1)\rho_i+1]\left(1-\rho_{p+1}\right). \tag{15}$$

Factoring out $n^p\rho_{p+1}$ from the left side of Eqn. (15) and multiplying through the second term on the right side of Eqn. (15) gives the following inequality,

$$n^p\rho_{p+1}\left((n-1)\prod_{i=1}^{p}\rho_i+\frac{1}{\rho_{p+1}}+1-1\right) >$$
$$(n-1)\rho_{p+1}\prod_{i=1}^{p}[(n-1)\rho_i+1]+\prod_{i=1}^{p}[(n-1)\rho_i+1]. \tag{16}$$

Multiplying through by ρ_{p+1} on the left side of Eqn. (16) and multiplying the through the first term on the right side of Eqn. (16) gives the following inequality,

$$n^p\left((n-1)\rho_{p+1}\prod_{i=1}^{p}\rho_i+1\right) >$$
$$\left[(n-1)\rho_{p+1}+1\right]\prod_{i=1}^{p}[(n-1)\rho_i+1], \tag{17}$$

and finally, replacing p with $p+1$ gives,

$$n^{(p+1)-1}\left((n-1)\prod_{i=1}^{p+1}\rho_i+1\right) > \prod_{i=1}^{p+1}[(n-1)\rho_i+1]. \tag{18}$$

Thus, the condition in Eqn. (8) for $p+1$ subsystems follows from the assumption that it is true for some $p \geq 2$ subsystems, and thus, it follows that Theorem 1 holds for $p+1$ subsystems given the assumption that it is true for some $p \geq 2$ subsystems. And since Theorem 1 is true for $p=2$ subsystems, it follows that it is true for all $p \geq 2$ subsystems. □

Theorem 1 indicates that for series systems a set of subsystem tests reduces the variance of the full system reliability estimate (and, hence, reduces the MSE and the confidence bound on the estimate) more than a single full system test. Of interest is the fact that as $\rho \to 1$ or $\rho \to 0$, the variances var $\left(\hat{\tilde{\rho}}\right)$ and var $\left(\prod_{i=1}^{p}\hat{\rho}_i\right)$ converge to zero. In the limit of a very reliable or a very unreliable system there is little difference in the reduction of the variance of the maximum likelihood reliability estimate (equivalently, reduction in the MSE and the confidence bound on the estimate) from performing a set of subsystem tests instead of a full system test.

4. MSE FOR THE GENERAL MLE OF SYSTEM RELIABILITY

Consider the following general maximum likelihood estimator of the parameter vector $\boldsymbol{\theta}$,

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\boldsymbol{Y}) \equiv \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathfrak{L}(\boldsymbol{\theta})$$
$$\text{subject to } \rho = h(\boldsymbol{\theta}), \tag{19}$$

where $\mathfrak{L}(\boldsymbol{\theta}) \equiv \log\left(p(\boldsymbol{Y}|\boldsymbol{\theta},\rho)\right)$ [9]. Given both system and subsystem test data, $\boldsymbol{Y}$, the estimate of ρ is derived from the MLE for $\boldsymbol{\theta}$ through the model for the system, h. The model dictates how subsystems are arranged in the full system (i.e., $\mathfrak{L}(\boldsymbol{\theta})$ is the same regardless of whether, the subsystems are in series or parallel). For a given parameter vector $\boldsymbol{\theta}$, the definition of $\mathfrak{L}(\boldsymbol{\theta})$ does *not* depend on the model for the system. However, the MLE does change as a function of the model for the system. This is a consequence of the system model being used as a constraint in the optimization problem that is solved to produce the MLE. From the assumption of independence of all test data, the probability mass function is:

$$p\left(\boldsymbol{Y}|\boldsymbol{\theta},\rho\right) = \underbrace{\binom{n}{Y}\rho^{Y}\left(1-\rho\right)^{(n-Y)}}_{\text{system}}$$
$$\times\underbrace{\binom{n_1}{X_1}\rho_1^{X_1}\left(1-\rho_1\right)^{(n_1-X_1)}\cdots\binom{n_p}{X_p}\rho_p^{X_p}\left(1-\rho_p\right)^{(n_p-X_p)}}_{p\text{ subsystems}}, \tag{20}$$

leading to the log-likelihood function $\mathfrak{L}(\boldsymbol{\theta}) \equiv \log p(\boldsymbol{Y}|\boldsymbol{\theta},\rho)$:

$$\mathfrak{L}(\boldsymbol{\theta}) = Y\log\rho+(n-Y)\log(1-\rho)+$$
$$\sum_{j=1}^{p}\left[X_j\log\rho_j+(n_j-X_j)\log(1-\rho_j)\right]+\text{ constant}, \tag{21}$$

where the constant is not dependent on $\boldsymbol{\theta}$. The MLE is determined by finding a root of the score equation $\partial\mathfrak{L}(\boldsymbol{\theta})/\partial\boldsymbol{\theta} = \boldsymbol{0}$ (or a normalized form of this equation in the asymptotic sample size case), where

$$\frac{\partial\mathfrak{L}}{\partial\boldsymbol{\theta}} = \frac{Y}{\rho}\frac{\partial h}{\partial\boldsymbol{\theta}}+\frac{n-Y}{1-\rho}\frac{\partial h}{\partial\boldsymbol{\theta}}+\begin{bmatrix}\frac{X_1}{\rho_1}-\frac{n_1-X_1}{1-\rho_1}\\ \vdots\\ \frac{X_p}{\rho_p}-\frac{n_p-X_p}{1-\rho_p}\end{bmatrix}. \tag{22}$$

The solution to $\partial\mathfrak{L}(\boldsymbol{\theta})/\partial\boldsymbol{\theta} = \boldsymbol{0}$ must generally be found by numerical search methods.

Except in trivial cases, the analytic expression for the variance of the general MLE for system reliability is not easily found. The likelihood function for the general MLE is twice differentiable with respect to $\boldsymbol{\theta}$ [9]. Assuming that the general MLE of system reliability is asymptotically unbiased (and assuming no modeling error), the Fisher Information, $\boldsymbol{F}(\boldsymbol{\theta})$, is given by the negative of the expectation of the second derivative of the likelihood function with respect to $\boldsymbol{\theta}$,

$$\boldsymbol{F}(\boldsymbol{\theta}) = -E\left[\frac{\partial^2 \mathfrak{L}}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right], \tag{23}$$

and, by the Cramer-Rao inequality, the inverse of the Fisher information is a lower bound on the variance the MLE. When $h(\boldsymbol{\theta})$ is differentiable in θ and if the asymptotic normality holds, then

$$\sqrt{\text{sample size}}\left[h(\hat{\boldsymbol{\theta}}) - h(\boldsymbol{\theta}^*)\right] \xrightarrow{dist} N\left(0, h'(\boldsymbol{\theta}^*)^T \overline{\boldsymbol{F}}^{-1} h'(\boldsymbol{\theta}^*)\right), \tag{24}$$

where $\boldsymbol{\theta}^*$ is the vector of true reliabilities for the subsystems and $\overline{\boldsymbol{F}}$ is the limit of the mean information matrix. Hence,

$$h(\hat{\boldsymbol{\theta}}) \sim N\left(h(\overline{\boldsymbol{\theta}}), h'(\overline{\boldsymbol{\theta}})^T \boldsymbol{F}(\overline{\boldsymbol{\theta}})^{-1} h'(\overline{\boldsymbol{\theta}})\right), \tag{25}$$

for $\overline{\boldsymbol{\theta}}$ close to $\boldsymbol{\theta}^*$ when the sample size is reasonably large [8]. In practice, $\overline{\boldsymbol{\theta}}$ is often set to $\hat{\boldsymbol{\theta}}$ in the mean and variance expressions of Eqn. (25). Thus, $\hat{\rho} = h(\hat{\boldsymbol{\theta}})$ has a normal distribution with an approximate variance given by the variance in Eqn. (25) evaluated at $\hat{\boldsymbol{\theta}}$. And so, the "variance of estimate" term of the MSE for the general MLE of system reliability is approximated with the asymptotic variance of the MLE. Further, assuming that the estimator is asymptotically unbiased and that there is no modeling error, the "bias of estimate" term is zero.

4.1 MSE With Model Error

If the system reliability model is incorrect or a model is chosen that does not adequately represent the system's behavior, then the true system reliability ρ is *not* uniquely determined by the model for the system, $h(\cdot)$. Given a model error, the estimate of system reliability from only subsystem test data is biased by the model error ε; that is

$$\rho = h(\boldsymbol{\theta}^*) + \varepsilon. \tag{26}$$

where, in general, $-\rho \leq \varepsilon \leq (1-\rho)$ to ensure $0 < \rho < 1$ (recall that $\boldsymbol{\theta}^*$ is the vector of true reliabilities for the subsystems). If the subsystem reliabilities are estimated individually and the model, h, is used to compute the system reliability, then the model error contributes a pure bias, ε to the estimate of full system reliability and to the MSE (as an example consider the estimator $\hat{\rho}_i = X_i/n_i$ for $i = 1, \ldots, p$). However, if the estimate of the subsystem reliabilities depends on the model h, (as in the general MLE) then the estimate, $\hat{\boldsymbol{\theta}}$, is influenced by the modeling error and the model error contribution to the bias is not straightforward.

To determine the bias in the general MLE for system reliability, $\hat{\boldsymbol{\theta}}$, due to a small error in the model for the system reliability, assume that the model, h, is in error as described in the preceding paragraph. Thus, the constraint in Eqn. (19) becomes $\rho \equiv h_\varepsilon(\boldsymbol{\theta}, \varepsilon) \equiv h(\boldsymbol{\theta}) + \varepsilon$, where ε is an error in the model (a true bias), which does not depend on $\boldsymbol{\theta}$. The addition of the modeling error will *not* change the log likelihood function. However, the relationship between ρ and the ρ_j differs, and so, the MLE of $\boldsymbol{\theta}$ differs. Let the function $g_{\mathrm{MLE}}(\boldsymbol{\theta}, \varepsilon)$ be the value of the score function in Eqn. (22), where the function h is replaced with h_ε. Given the model error, ε, the MLE is determined by finding a root of $g_{\mathrm{MLE}}(\boldsymbol{\theta}, \varepsilon)$.

Since the model error is unknown before estimation and its presumed value may change after the vector $\hat{\boldsymbol{\theta}}$ has been estimated, the new MLE of $\boldsymbol{\theta}$ is formulated as first-order function of ε via Taylor series expansion. The function that produces the MLE, Eqn. (19), is written:

$$\hat{\boldsymbol{\theta}}_\varepsilon = \hat{\boldsymbol{\theta}}(\boldsymbol{Y}) + \left.\frac{\partial\hat{\boldsymbol{\theta}}}{\partial\varepsilon}\right|_{\boldsymbol{Y},\varepsilon=0} \varepsilon + O(\varepsilon^2), \tag{27}$$

where $\hat{\boldsymbol{\theta}}_\varepsilon$ is the MLE given that the maximization of the likelihood function is subject to the constraint $\rho = h_\varepsilon(\boldsymbol{\theta}, \varepsilon)$. Assuming that the model error, ε is small, the $O(\varepsilon^2)$ and higher-order terms of the Taylor Series expansion can be dropped to form a good approximation of $\hat{\boldsymbol{\theta}}_\varepsilon$.

To obtain the derivative, $\partial\hat{\boldsymbol{\theta}}/\partial\varepsilon$, the conditions necessary to obtain a unique function $\hat{\boldsymbol{\theta}}(\varepsilon)$ relating ε to $\boldsymbol{\theta}$ must be established.

LEMMA *1. Suppose that the constraint in Eqn.* (19) *is rewritten* $\rho = h(\boldsymbol{\theta}) + \varepsilon$ *and the solution to the maximization problem in the presence of the model error* ε *is determined by finding the root of* $g_{MLE}(\boldsymbol{\theta}, \varepsilon)$ *with* g_{MLE} *being a continuously differentiable function in* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. *For* $\varepsilon = 0$, *suppose* $\partial g_{MLE}(\boldsymbol{\theta}, \varepsilon)/\partial\boldsymbol{\theta}$ *is invertible and* $\det[\partial g_{MLE}(\boldsymbol{\theta}, \varepsilon)/\partial\boldsymbol{\theta}] \neq \boldsymbol{0}$ *at* $\hat{\boldsymbol{\theta}}$ *such that* $g_{MLE}(\hat{\boldsymbol{\theta}}, \varepsilon) = \boldsymbol{0}$. *Then there exists an open neighborhood about* $\varepsilon = 0$ *and a unique continuous differentiable function* $\hat{\boldsymbol{\theta}}(\cdot)$ *such that for all* ε *in this neighborhood,* $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}(\varepsilon)$ *and*

$$\frac{\partial\hat{\boldsymbol{\theta}}(\varepsilon)}{\partial\varepsilon} = -\left(\frac{\partial g_{MLE}(\boldsymbol{\theta}, \varepsilon)}{\partial\boldsymbol{\theta}}\right)^{-1} \frac{\partial g_{MLE}(\boldsymbol{\theta}, \varepsilon)}{\partial\varepsilon}. \tag{28}$$

PROOF. By definition, the MLE satisfies, $g_{\mathrm{MLE}}(\boldsymbol{\theta}, \varepsilon) = \boldsymbol{0}$. Also, the derivative of the score vector is the Hessian,

$$\frac{\partial g_{\mathrm{MLE}}(\boldsymbol{\theta}, \varepsilon)}{\partial\boldsymbol{\theta}} = \frac{\partial^2 \mathfrak{L}(\boldsymbol{\theta}, \varepsilon)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T},$$

which is a negative definite matrix when evaluated at the solution to the maximum likelihood problem. By the Implicit Function Theorem [see 1, Section 13.4], there exists an open neighborhood about $\varepsilon = 0$ and one, and only one, continuously differentiable function $\hat{\boldsymbol{\theta}}(\cdot)$ such that for all ε in this neighborhood, $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}(\varepsilon)$. Further, $\partial\hat{\boldsymbol{\theta}}/\partial\varepsilon$ is as shown in Eqn. (28) in this neighborhood [see 10, p. 483]. □

Lemma 1 establishes the existence of the derivative, $\partial\hat{\boldsymbol{\theta}}/\partial\varepsilon$, which is found explicitly via Eqn. (28). However, the derivative in Eqn. (27) is evaluated with $\varepsilon = 0$, and so, the terms in Eqn. (28) must also be evaluated with $\varepsilon = 0$. The first term in Eqn. (28) is found by taking the second derivative of the score function, Eqn. (22), where the function h is replaced with h_ε. When evaluated with $\varepsilon = 0$, the term simplifies to the inverse of the Hessian for the log likelihood, as follows,

$$\left(\left[\frac{\partial g_{\mathrm{MLE}}(\boldsymbol{\theta}, \varepsilon)}{\partial\boldsymbol{\theta}}\right]_{\varepsilon=0}\right)^{-1} = \left(\frac{\partial^2 \mathfrak{L}}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right)^{-1}. \tag{29}$$

The second term in Eqn. (28) is found by taking the second derivative of the score function, Eqn. (22), where the function h is replaced with h_ε. The elements of the vector,

$\partial g_{\mathrm{MLE}}(\boldsymbol{\theta},\varepsilon)/\partial\varepsilon$, are,

$$\begin{aligned}\frac{\partial^2\mathfrak{L}(\boldsymbol{\theta},\varepsilon)}{\partial\theta_j\partial\varepsilon} &= \frac{Y}{h_\varepsilon(\boldsymbol{\theta},\varepsilon)}\frac{\partial^2 h_\varepsilon(\boldsymbol{\theta},\varepsilon)}{\partial\theta_j\partial\varepsilon} - \frac{n-Y}{1-h_\varepsilon(\boldsymbol{\theta},\varepsilon)}\frac{\partial^2 h_\varepsilon(\boldsymbol{\theta},\varepsilon)}{\partial\theta_j\partial\varepsilon}\\ &-\left\{\left(\frac{\partial h_\varepsilon(\boldsymbol{\theta},\varepsilon)}{\partial\theta_j}\frac{\partial h_\varepsilon(\boldsymbol{\theta},\varepsilon)}{\partial\varepsilon}\right)\left(\frac{Y}{h_\varepsilon(\boldsymbol{\theta},\varepsilon)^2}+\frac{n-Y}{(1-h_\varepsilon(\boldsymbol{\theta},\varepsilon))^2}\right)\right\},\\ &= -\left(\frac{\partial h_\varepsilon(\boldsymbol{\theta},\varepsilon)}{\partial\theta_j}\frac{\partial h_\varepsilon(\boldsymbol{\theta},\varepsilon)}{\partial\varepsilon}\right)\left(\frac{Y}{h_\varepsilon(\boldsymbol{\theta},\varepsilon)^2}+\frac{n-Y}{(1-h_\varepsilon(\boldsymbol{\theta},\varepsilon))^2}\right),\end{aligned} \tag{30}$$

for $j=1,\ldots,p$, because, given the definition of h_ε, the second derivative $\partial^2 h_\varepsilon(\boldsymbol{\theta},\varepsilon)/\partial\theta_j\partial\varepsilon = 0$. The terms in Eqn. (30) simplify further when used in Eqn. (27) because the derivative is evaluated with $\varepsilon = 0$.

Eqns. (27)–(30) comprise the mathematical machinery necessary for determining the bias in the general MLE of $\boldsymbol{\theta}$ due to a small error in the model. The difference, $\hat{\boldsymbol{\theta}}_\varepsilon - \hat{\boldsymbol{\theta}}$, is the sensitivity of the MLE to the model error ε. To determine the bias in the full system estimate, $\hat{\rho}$, a Taylor series expansion is used to find the MLE of ρ, given a deterministic error, ε, in the model:

$$\hat{\rho}_\varepsilon = h_\varepsilon\left(\hat{\boldsymbol{\theta}}\right) + \left.\frac{\partial h_\varepsilon}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\left(\hat{\boldsymbol{\theta}}_\varepsilon - \hat{\boldsymbol{\theta}}\right) + O\left(\left(\hat{\boldsymbol{\theta}}_\varepsilon - \hat{\boldsymbol{\theta}}\right)^2\right). \tag{31}$$

For series, parallel, and combination series/parallel systems the second order and greater terms are zero. The remaining terms of Eqn. (31) are supplied by Eqns. (27)–(30) to give an explicit formula for the MLE of ρ, given a deterministic error, ε:

$$\begin{aligned}\hat{\rho}_\varepsilon &\approx h\left(\hat{\boldsymbol{\theta}}\right) + \varepsilon + \left.\frac{\partial h_\varepsilon}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\cdot\left.\frac{\partial\hat{\boldsymbol{\theta}}}{\partial\varepsilon}\right|_{\boldsymbol{y},\varepsilon=0}\varepsilon,\\ &\approx \hat{\rho} + \varepsilon - \left.\frac{\partial h_\varepsilon}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\left(\frac{\partial^2\mathfrak{L}(\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right)^{-1}\left[\frac{\partial g_{\mathrm{MLE}}(\hat{\boldsymbol{\theta}},\varepsilon)}{\partial\varepsilon}\right]_{\varepsilon=0}\varepsilon.\end{aligned} \tag{32}$$

The quantity, $|\hat{\rho_\varepsilon} - \hat{\rho}|$, is the magnitude of the change in the estimate of full system reliability due to the modeling error, ε. It is a measure of the sensitivity of the general MLE of full system reliability to a given modeling error, and it is the bias in the MLE due to the modeling error, ε.

Practically, the outcome of the tests are unknown before a testing regime must be developed. Thus, the expectation of the quantitiy $|\hat{\rho_\varepsilon} - \hat{\rho}|$ is useful for test sizing and evaluating estimator accuracy. The expression for the expectation is,

$$E\left[\hat{\rho_\varepsilon} - \hat{\rho}\right] \approx \varepsilon + \left.\frac{\partial h_\varepsilon}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\left(\boldsymbol{F}(\hat{\boldsymbol{\theta}})\right)^{-1} E\left[\frac{g_{\mathrm{MLE}}(\hat{\boldsymbol{\theta}},\varepsilon)}{\partial\varepsilon}\right]_{\varepsilon=0}\varepsilon. \tag{33}$$

Note that the Hessian is replaced with its expectation, which is the negative of the Fisher Information, $\boldsymbol{F}(\boldsymbol{\theta})$, for the general maximum likelihood estimator. The expectation is a bias in the MLE estimate due to model error.

Given the model error, ε, the MSE of the general maximum likelihood estimator is composed of the asymptotic variance of the estimate from and the approximate expected bias of the estimate from Eqn. (33). From Eqn. (1), the expression for the MSE is,

$$E\left[\left(h\left(\hat{\boldsymbol{\theta}}\right) - \rho\right)^2\right] \approx h'(\hat{\boldsymbol{\theta}})^T\boldsymbol{F}(\hat{\boldsymbol{\theta}})^{-1}h'(\hat{\boldsymbol{\theta}}) + (E\left[\hat{\rho_\varepsilon} - \hat{\rho}\right])^2. \tag{34}$$

Note that $E\left[\hat{\rho_\varepsilon} - \hat{\rho}\right] = \varepsilon$ when full system tests are not performed and that the quantity ε^2 is the penalty to the MSE of the estimator for not performing any full system tests. The bias, ε, is a subjective input into any analysis performed using the MSE in Eqn. (34) (e.g. performing a test sizing study to minimize MSE); it is interpreted at the maximum error in the model for the system reliability h.

4.2 Special Case: Fully Series System

From Eqn. (19), the MLE in the series-subsystem case is found according to

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\mathfrak{L}(\boldsymbol{\theta})$$

$$\text{subject to } \rho = \prod_{j=1}^{p}\rho_j, \tag{35}$$

It is straightforward to determine the score vector taking partial derivatives of Eqn. (21). Making the substitution $\rho = \prod_{j=1}^{p}\rho_j$ in Eqn. (21) and taking the derivative, the $j = 1,2,\ldots,p$ elements of the score vector for the series case are:

$$\frac{\partial\mathfrak{L}}{\partial\rho_j} = \frac{Y+X_j}{\rho_j} - \frac{(n-Y)\rho}{(1-\rho)\rho_j} - \frac{(n_j - X_j)}{1-\rho_j}, \tag{36}$$

[9]. Because $\Theta = 0 < \rho_j < 1$ for all j, it is known that the Hessian matrix is continuous and, consequently, symmetric. From Eqn. (36), the elements of the Hessian for the series case are:

$$\frac{\partial^2\mathfrak{L}}{\partial\rho_j\partial\rho_k} = \begin{cases} -\frac{Y+X_j}{\rho_j^2} - \frac{(n-Y)\rho^2}{(1-\rho)^2\rho_j^2} - \frac{(n_j-X_j)}{(1-\rho_j)^2} & \text{when } j=k,\\ -\frac{(n-Y)\rho}{(1-\rho)^2\rho_j\rho_k} & \text{when } j\neq k,\end{cases} \tag{37}$$

[9].The Fisher information matrix, $\boldsymbol{F}(\boldsymbol{\theta})$, for the series case is the negative expectation of the Hessian. The corresponding elements of the information matrix $\boldsymbol{F}(\boldsymbol{\theta}) = \left[F_{jk}(\theta)\right]$ for the series case are:

$$F_{jk}(\theta) = \begin{cases} \frac{n\rho+n_j\rho_j}{\rho_j^2} + \frac{n\rho^2}{(1-\rho)\rho_j^2} + \frac{n_j}{(1-\rho_j)} & \text{when } j=k,\\ \frac{n\rho}{(1-\rho)\rho_j\rho_k} & \text{when } j\neq k,\end{cases} \tag{38}$$

[9]. The elements of the derivative $\partial h_\varepsilon/\partial\boldsymbol{\theta}$ for the series case are

$$\frac{\partial h_\varepsilon}{\partial\rho_i} = \prod_{j=1,j\neq i}^{p}\rho_j. \tag{39}$$

Finally, the derivative, $\partial g_{\mathrm{MLE}}(\rho_i,\varepsilon)/\partial\varepsilon$, is found by evaluating Eqn. (30) with the series model, this gives,

$$\frac{\partial g_{\mathrm{MLE}}(\rho_i,\varepsilon)}{\partial\varepsilon} = \frac{Y}{\rho_i\prod_{i=1}^{p}\rho_i} - \frac{(n-Y)\prod_{i=1}^{p}\rho_i}{\left(1-\prod_{i=1}^{p}\rho_i\right)^2\rho_i}, \tag{40}$$

and, the expected value of the derivative is

$$E\left[\frac{\partial g_{\mathrm{MLE}}(\rho_i,\varepsilon)}{\partial\varepsilon}\right] = \frac{n}{\rho_i} - \frac{n\prod_{i=1}^{p}\rho_i}{\left(1-\prod_{i=1}^{p}\rho_i\right)\rho_i}. \tag{41}$$

Eqns. (35), (38), (39), and (41) are the terms necessary for evaluating the MLE with the MSE (the asymptotic lower bound variance and bias) for a series system, given the modeling error ε.

5. OPTIMAL TEST PLANS

In this section, the optimal combination of system and subsystem tests, in terms of MSE and total test plan cost, is determined for a hypothetical series system using the methodology described in the previous section. The hypothetical system is an analogue for a system composed of five independent subsystems that are testable. Let the reliabilities of the five subsystems be as defined in Table 1. Given no modeling error, the full system reliability is 0.904, and performing a set of subsystem tests (a set of subsystem tests consists of one test for each subsystem) reduces the MSE (estimation uncertainty) more than a full system test (see Section 3).

Table 1: Subsystem reliabilities for a hypothetical system with five subsystems in series.

	Reliability
Subsystem 1	0.990
Subsystem 2	0.985
Subsystem 3	0.980
Subsystem 4	0.975
Subsystem 5	0.970

The methodology described in the previous section allows a test planner to assume that the system reliability model may be incorrect (the function that relates the subsystem reliabilities to the full system reliability is incorrect). Among other reasons, model error may arise because some of the subsystems are dependent or because a component is left out of the subsystem definitions or test plan. The methodology allows the model error to contribute a bias to the MSE of the general maximum likelihood estimator based on the number of full system/subsystems tests. Loosely, full system tests contribute unbiased information to the general maximum likelihood estimator. Thus, as the number of full system tests increases relative to the number of sets of subsystem tests, the model error contributes less to the bias term of the MSE.

The contours of the MSE for the general maximum likelihood estimator of system reliability are computed for the hypothetical system from Eqn. (34) given no model error and given three different non-zero model errors: $\varepsilon = -0.025$, $\varepsilon = -0.050$, and $\varepsilon = -0.075$. (Hence, the maximum errors range from approximately 2.8 to 8.3 percent of the true reliability.) The contours are plotted in Figure 1. The X-axes of the plots is the number of sets of subsystem tests, and the Y-axes of the plots is the number of full system tests. Each contour is a Pareto frontier for achieving the specified MSE, given the modeling error, in terms of the number of full system tests and sets of subsystem tests. The subfigure for the case with no model error indicates that a set of subsystem tests reduces the MSE of the general MLE more than a full system test (although the difference in the reduction is small indicating that a set of subsystem test is worth about the same as a full system test in terms of reducing MSE). The sub-figures for the nonzero negative model errors indicate that the model for system reliability (a series model) produces overly confident estimates of system reliability when subsystem tests are predominant. The MSE

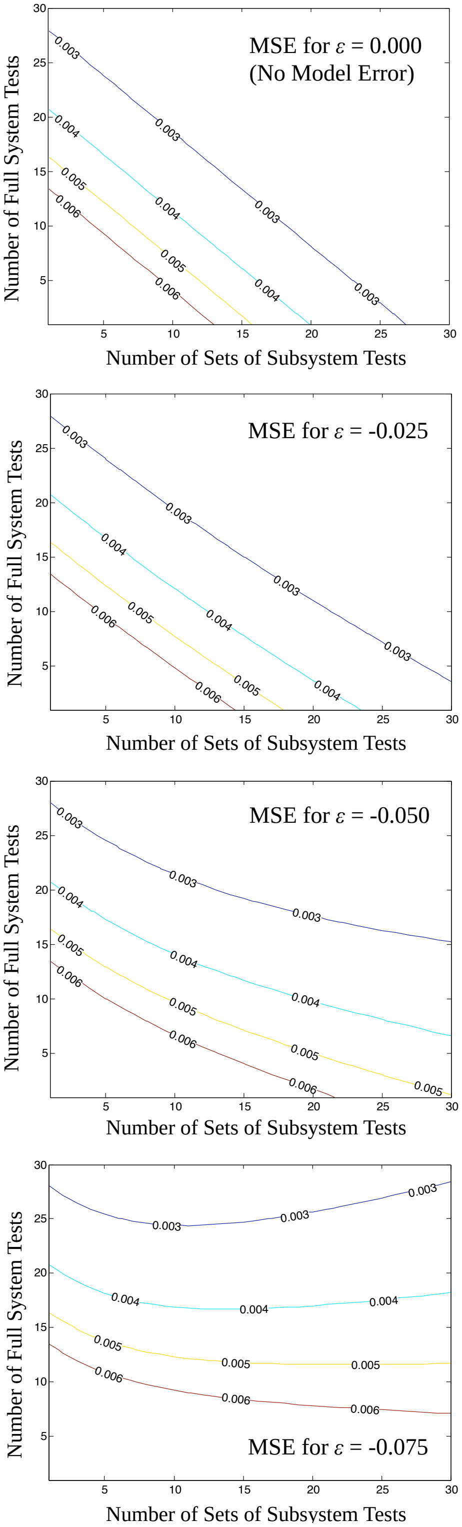


Figure 1: The contours of the MSE are computed for the hypothetical system, given no model error and given three different model errors: $\varepsilon = -0.025$, $\varepsilon = -0.050$, and $\varepsilon = -0.075$.

values are penalized by the square of the assumed modeling error when full system tests are not performed and less when some full system tests are performed. Assuming the model error $\varepsilon = -0.025$, $\varepsilon = -0.050$, or $\varepsilon = -0.075$, indicates that performing a full system test reduces the MSE more than a set of subsystem tests.

The design of a test plan should also account for the cost of the tests. To achieve an MSE of 0.005 or less (root mean squared error 0.07 or less) many different test plans can be devised. The total cost of the test plan depends on the number of each type of test: subsystem and full system. To illustrate the effect of cost on the test plan design, assume that a full system test of the hypothetical system (described in the previous paragraph) costs twice as much as a set of subsystem tests. If the modeling error is $\varepsilon = -0.025$, then the optimal test plan, in terms of cost, is to always perform sets of subsystem tests. However, if the modeling error is $\varepsilon = -0.050$ or $\varepsilon = -0.075$, then the optimal test plan is a mix of full system and sets of subsystem tests.

The cost benefit of performing a mixture of full system and subsystem test is is depicted in Figure 2. Four test plans are listed, each provide a MSE of 0.005, given a model error of $\varepsilon = -0.050$. The maximum cost test plan consists of performing only full system tests. The other three test plans, which consist of a mixture of full system and subsystem tests, are less costly (given that a set of subsystem tests is half the cost of a full system test). The potential cost reduction from performing one of these three test plans instead of performing only full system tests is plotted as a percentage in Figure 2. For $\varepsilon = -0.050$, the minimum cost test plan consists of nine sets of subsystem tests and ten full system tests. Several other test plans have the same total cost, for example, nine full system and eleven sets of subsystem tests or eight full system tests and thirteen sets of subsystem tests. However, if modeling error is a concern, then it is optimal to perform the maximum number of full system tests that can be performed while achieving the desired MSE for the least cost. The contours of the MSE for $\varepsilon = -0.075$ betray the optimal test plan in terms of cost for achieving a MSE of 0.005 (see Figure 1). The contour of 0.005 is almost level after a few sets of subsystem tests, and so, the optimal test plan in terms of cost is to perform fifteen system level tests and two sets of subsystem tests.

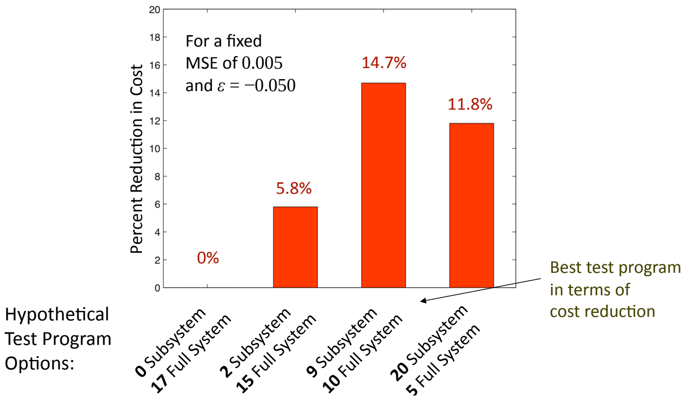


Figure 2: The potential cost reduction from performing a mixture of full system and subsystem tests instead of performing only full system tests.

6. CONCLUDING REMARKS

The main purpose of this paper was to develop a method for including the effect of modeling error in the MSE of a general MLE for system reliability. This was accomplished by decomposing the MSE of the MLE into the variance of the MLE and a bias from incorrectly specifying the model for the system reliability. The variance of the MLE was approximated with the Fisher Information. The bias was approximated by computing the sensitively of the system reliability MLE to a maximum modeling error, given a proposed test plan consisting of a mixture of full system and subsystem tests. The method showed that the bias penalty in the MSE diminished as the number of full system tests increased relative to the number of sets of subsystem tests and that the square of the model error was the bias term in the MSE when full system tests were not performed. The method enables optimum test plans to be developed for system reliability estimation involving trade-offs between the MSE (estimation accuracy), the degree of modeling error, and the cost of doing system and subsystem tests.

References

[1] T. M. Apostol. *Mathematical Analysis*. Addison-Wesley, Reading, MA, second edition, 1974.

[2] Y. Y. Haimes. *Risk Modeling, Assessment, and Management*. Wiley Interscience, New York, 1998.

[3] M. Henrion and M. G. Morgan. *Uncertainty: A Guide to Dealing with Uncertianty in Quantitative Risk and Policy Analysis*. Cambridge University Press, New York, 1990.

[4] R. Krzysztofowicz and K. S. Kelly. Hydrologic uncertainty processor for probabilistic river stage forecasting. *Water Resources Research*, 36(11):3265–3277, 2000.

[5] L. Ljung. *System Identification Theory for the User*. Prentice Hall, Uper Saddle River, NJ, second edition, 1999.

[6] C. J. Maranzano and R. Krzysztofowicz. Bayesian reanalysis of the Challenger O-ring data. *Risk Analysis*, 28(4):1053–1067, 2008.

[7] C. R. Rao. *Linear Statistical Inference and Its Applications*. Wiley Interscience, New York, second edition, 1973.

[8] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley Interscience, Hoboken, New Jersey, 2003.

[9] J. C. Spall. System reliability estimation and confidence regions from subsystem and full system tests. In *Proceedings of the American Control Conference*, number FrB14.1, St. Louis, MO, June 2009.

[10] W. F. Trench and B. Kolman. *Multivariable Calculus with Linear Algebra and Series*. Academic, New York, 1972.

[11] R. Zhang and S. Mahadevan. Model uncertainty and Bayesian updating in reliability-based inspection. *Structural Safety*, 22:145–160, 2000.

Unmanned and Autonomous Systems Mission Based Test and Evaluation

Philipp A. Djang, Ph.D.
Army Research Labs
Survivability/Lethality Analysis Directorate
Information & Electronic Protection Division
Attn: AMSRL-ARL-SL-EA
White Sands Missile Range, NM 88002
575-678-1550, philipp.djang@us.army.mil

Frank Lopez
Army Research Labs
Survivability/Lethality Analysis Directorate
Information & Electronic Protection Division
Attn: AMSRL-ARL-SL-EA
White Sands Missile Range, NM 88002
575-678-8316, frank.lopez@us.army.mil

ABSTRACT

We propose to apply principles from the Army Evaluation Center's Mission Based Test and Evaluation (MBT&E) to Unmanned and Autonomous Systems (UAS) Test and Evaluation (T&E) in order to conduct rigorous, real-world testing based on anticipated military missions. In order to understand MBT&E, we introduce and describe its parent, the Mission and Means Framework. Finally, we describe a vignette that incorporates autonomous systems in the context of a mission to illustrate these principles.

Categories and Subject Descriptors

General Terms

Management, Performance, Design, Experimentation, Standardization, Languages, Verification

Key Words

Mission and Means Framework, Mission Based Test and Evaluation, Measures of Performance, Measures of Effectiveness, Simulation Based Test and Evaluation, Unmanned and Autonomous System Test and Evaluation, Capability Based Evaluation

1. Mission and Means Framework

The two – sided missions and means framework provides a structured way to describe key elements of military operations that are essential to understand in order to successfully model and simulate those operations. The framework provides the necessary structure to support a disciplined, repeatable procedure to explicitly specify the mission and assess mission accomplishment [1].

The framework consists of seven levels and four operators that describe a military mission. Cast within a context and environment, each side executes a mission to achieve a specific purpose. A mission is decomposed into tasks which are the building blocks and based on authoritative sources such as the Uniform Joint Task List (UJTL) which are commonly accepted terminology and definitions. task lists such as the UJTL, etc. are deliberately designed to facilitate the ability to associate mission specific conditions and standards[2].

PerMIS'09, September 21-23, Gaithersburg, MD, USA, ACM 978-60558-747-9/09/09

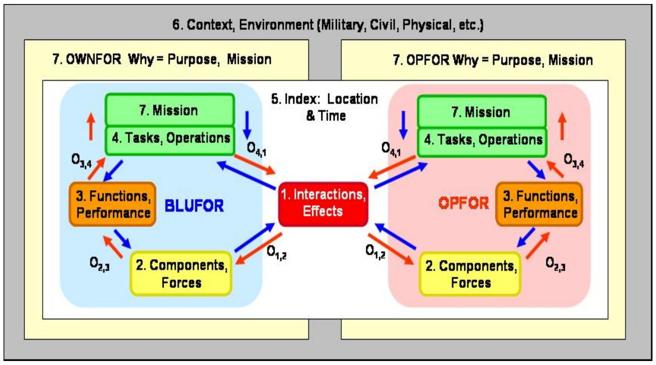


Figure 1: Mission Means Framework

Conditions and standards for specific tasks are established based war gaming and course of action (COA) mission planning. The same task may be iterated several times with different sets of conditions and standards based on when and where the task iteration is to occur within the concept of operations. The execution of the task (operations) may be structured to provide quantitative metrics in the form of Measures of Performance (MoP) which describe minimum acceptable levels of performance in terms of time, distance, accuracy, etc. Standards may also be structured to provide more qualitative metrics in the form of Measures of Effectiveness (MoE) which describe the desired end state or purpose of the task as specified by the commander in his intent statement. Functions and capabilities are based on physical systems (and systems of systems) and their components. When a physical system is employed in the context of a task, the degree to which the task can be successfully executed depends upon the capability and functionality afforded by the system. Components (the building blocks of systems) and forces (human and robotic) are directly affected by interactions and effects (kinetic, electromagnetic, etc.) of conflict.

The framework employs four operators to describe the dynamics of military planning and decision making as both the functions and capabilities of components and forces are affected by conflict. The blue operators (arrows) represent military planning and re-planning based on available components and forces. The red operators describe feedback dynamics of the conflict.

2. Mission Based Test & Evaluation (MBT&E)

In 2008, the Army Evaluation Center adopted and customized the Mission Means Framework as a touchstone for conducting testing. MBT&E [3] is a methodology that focuses T&E on the capabilities provided to the war fighter. MBT&E links the mission and associated tactics, tasks and performance standards with the capabilities of the system under test. The Center shifted their focus from component testing (meets a standard) to answer a more challenging and relevant question: "Does the system provide what the war fighter needs to accomplish the mission?" with respect to the test domain (performance, quality, reliability, etc). MBT&E provides a framework, procedure and complexity constraint strategies to 1) link capabilities to the attributes of the materiel system-of-systems, 2) develop evaluation measures that assess capabilities and attributes and 3) and link the evaluation measures to all available data sources.

MBT&E is focuses on a key concept: capability – which is defined as the ability to achieve a desired effect or result, outcome, or consequence of a task under specified standards and conditions, through a combination of means and ways to perform a set of tasks. Capabilities are the building blocks for this evaluation strategy.

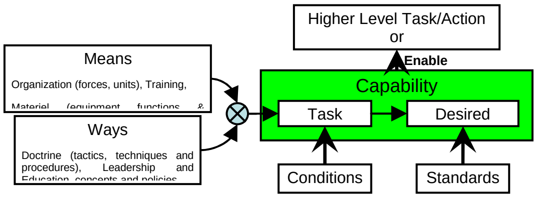

Higher Level Task/Action
or
Enable
Means
Organization (forces, units), Training,
Materiel (equipment functions &
Ways
Doctrine (tactics, techniques and procedures), Leadership and
Education concepts and policies
Capability
Task
Desired
Conditions
Standards

Figure 2: MBT&W Capability

System performance is measured by systems engineering metrics are integrated to determine system performance functionality in the context of doctrine, tactics, leadership and policies. They are integrated into system-of-system task performance and compared against MOPs and MOEs.

MBT&E specifies five major purpose areas with a total of 19 steps. The five major purpose areas are: 1) understand the mission, 2) understand the system, 3) design the test & evaluation, 4) determine the results and 5) report the results. MBT&E is an evolving process and a number of innovative proof-of-principle pilot studies are underway.

3. Application of MBT&E to UAS

For the purpose of modeling mission – task – function – component decomposition, we present the following vignette which was a developed as part of a larger scenario. This vignette was created with a minimalist intent to allow researchers to isolate the impact of component failure upon functions, task performance and mission accomplishment.

Per the MBT&E guidelines, we specify the following environmental context: The area of operation is Southwest Asia (desert environment); the context for the scenario is that the national government has been overthrown and a combination of terrorists and militants will take possession of a nuclear weapons facility. The mission for blue forces are to overcome local security forces if required, emplace failsafe devices on the nuclear weapons and secure the facility for follow-on forces. For the vignette, the mission of the blue force is to conduct a reconnaissance by fire. The mission of the red force is to set up a hasty defense and prevent blue forces from traversing the road. The force composition of this vignette is simple: two red tanks versus a blue autonomous aerial surveillance drone (UAV) and an autonomous armed ground robot (UGV).

Our mission-task decomposition follows standard doctrine until autonomous system tasks are assigned. Autonomous system tasks do not currently exist and were adapted from currently existing tasks. In particular, our UAS tasks incorporate surveillance, the ability to infer enemy intent and re-plan a navigation course that will take advantage of terrain features to provide tactical surprise. Our mission to task relationship:

- "SN 3 Employ Forces": represents the decision and action taken at the national level to use the military element of national power in response to a crisis caused by the actions of an external opposing faction.
- "ST 1.3.4 Integrate Direct Action in Theater": represents the planning and coordination actions taken by the geographic combatant commander to secure the nuclear warheads.
- "OP 1.2.4.7 Conduct Direct Action in the Joint Operations Area (JOA)": represents the planning and execution actions being taken by the Joint Task Force Commander responsible for the JOA, to neutralize opposing forces in order to secure the nuclear warheads.
- "ART 8.1.2 Conduct an Attack": represents the mission given to the blue forces (humans, traditional systems and robotic systems) to support the Joint Task Force's action to neutralize opposing forces. The purpose of the attack is to destroy opposing forces and occupy positions on key terrain in

order to prevent those forces from capturing the nuclear warheads.

- “UAS 2.3 Perform Intelligence, Surveillance and Reconnaissance”; represents the activity performed by unmanned sensors (i.e. UAV’s) to monitor, detect, identify and report enemy activity in areas of interest.
- “UAS 2.3.1 Provide aerial tactical intelligence overwatch”: represents activity performed by a UAV to detect, communicate enemy location and movement in areas of interest.
- “UAS 3.3 Employ Fires”: represents the means by which the robotic equipped company intends to engage opposing faction forces who might interfere with their maneuver to and occupation of the key terrain.
- “UAS 3.3.3.1 Conduct route reconnaissance”: by robotic assets based on terrain database movement and tactical intelligence
- “UAS 3.3.5.1 Exploit terrain to expedite tactical movements”: – based on mobility constraints, enemy location and terrain features, compute optimal path
- “UAS 3.3.8 Conduct Lethal Direct Fire Attack”: – apply direct fire to neutralize identified enemy.

We adapted the UAS specific tasks from FM 7-15 Army Universal Task List for both the UAV and the UGV. The UAV is responsible for UAS 2.3 Perform Intelligence, Surveillance and Reconnaissance and it’s subordinate task: UAS 2.3.1 Provide tactical intelligence over watch. The tasks assigned to the UGV are UAS 3.3.3.1 Conduct route reconnaissance, UAS 3.3.5.1 Exploit terrain to expedite tactical movements and UAS 3.3.8 Conduct Lethal Direct Fire Attack. Inferring enemy intent is a joint intelligence task UAS 2.1 Collaborative Situational Decision Making.

During the initial phase of the operation, the UAV performs tasks UAS 2.3 Perform Intelligence, Surveillance and Reconnaissance and subordinate task UAS 2.3.1 Provide aerial tactical intelligence over watch. During the execution of this task, the UAV identifies two enemy tanks that are located on the road. This information is communicated to the UGV. The UAV determines that the enemy tanks are in a fortified position and is blocking the route. The UAV infers that the mission of the enemy is to prevent friendly vehicles from passing their fortified position.

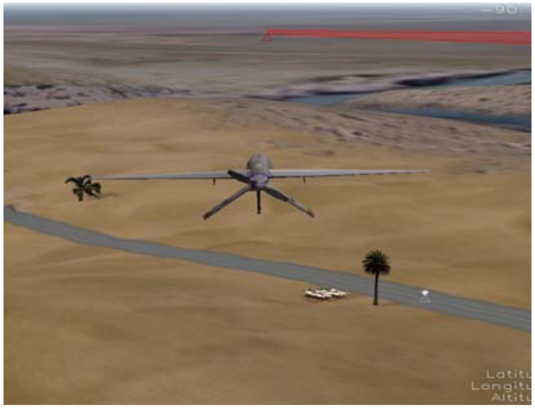


Figure 3: Vignette – UAV performs aerial tactical intelligence over watch

Once the enemy’s intent is determined and passed to the UGV, the UGV performs UAS 3.3.3.1 conduct route reconnaissance and determines that the original course of action – to follow the road – is no longer viable. The UGV switches to task UAS 3.3.5.1 Exploit terrain to expedite tactical movements and re-computes a new course of action based on the terrain elevation and mobility. The UGV computes an alternative route dynamically by evaluating off-road mobility and selecting terrain features to mask movement in order to maximize the element of surprise.

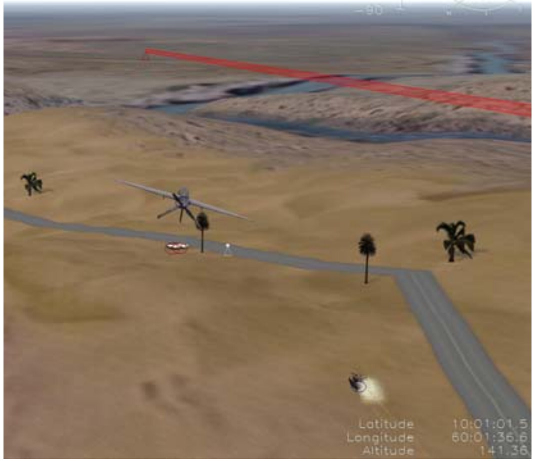


Figure 4: Vignette – UGV exploits terrain to expedite tactical movement

The new route exploits the hilly nature of the terrain. The UGV computes an off road route that requires more fuel expenditure because of decreased mobility associated with sand but provides a tactical advantage. The enemy forces expect the main attack from the direction of the road. Once the UGV crests the hill, task UAS 3.3.8 Conduct Lethal Direct Fire Attack is executed and the enemy tanks are destroyed.

Figure 5: UGV Conducts Lethal Direct Fire Attack

4. Component Failure and Loss of Functionality

While the outcome of the vignette favors the blue side, the value of mission based testing is illustrated when a component or components fail due to stress or are destroyed during ballistic interactions.

We examine a key subsystem of the weapon system that the UGV will use to destroy the opposing enemy tanks. This subsystem is critical to accomplishing UAS task 3.3.8 Conduct Lethal Direct Fire.

The projectile tracking sub-system of the UGV weapon system is an electromechanical system that is composed of multiple components and subordinate subsystems. The purpose of the projectile tracking sub-system is to measure the ballistic trajectory of the projectile (bullet) and determine if the projectile reached a desired aim point. The components of the projectile tracking system (PTS) shown in figure 6 are:

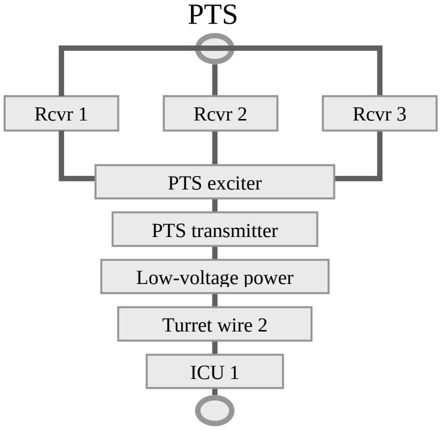


Figure 6: Projectile tracking system (PTS)

There are three parallel sensors connected to four components. The integrated circuit unit (ICU 1) computes the ballistic trajectory and determines if the system has achieved the desired aim point. While one sensor may fail, the other two sensors can provide the required information. But suppose one of the components in the serial chain fails before the ICU can process this information? Clearly, the system will not be able to accurately track the ballistic trajectory of the projectile.

Failure of one of the components in the PTS is critical because for the UGV, the PTS provides critical feedback to the UGV primary computing system. The relationship between the PTS and other subsystems is shown in Figure 7.

The PTS is linked to the vehicle mobility subsystems: drive sprocket, tracks, left and right traction motors and backup braking. Because the PTS is linked to the mobility subsystems, it failure will prevent the UGV from correctly adjusting it position so that future projectiles can be accurately aimed

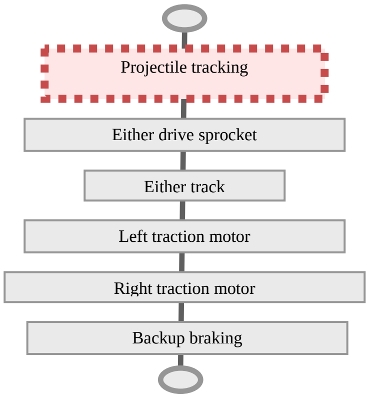


Figure 7: PTS and Mobility Subsystems

.

If the PTS failed, the UGV would be unable to accomplish UAS task 3.3.8 Conduct Lethal Direct Fire. In turn, the enemy tanks in their reinforced blocking position on the only road would prevent blue force access. Denying access to the road would degrade or delay the ability of the blue forces to accomplish a higher order task: ART 8.1.2 Conduct an Attack.

From a graph theoretic viewpoint, the relationship between the mission, tasks, functions or capabilities and components or forces can be viewed as a rooted tree. The mission is of course the root of the tree with branches for each task, as well as for each function or capability and with terminal leaves as components or forces. The role of Mission Based Test and Evaluation is to determine the impact of component, subsystem and subsystem failures and map these failures to this directed graph in order to find a minimum cut set that prevents the successful accomplish of the mission. Identifying the maximum likelihood of such cut sets could lead to more robust mission based metrics.

5. Conclusion

The Mission and Means Framework is responsible for stimulating innovative concepts and applications across the Department of Defense. One of these applications has been MBT&E.

MBT&E is a new philosophy developed by the Army Evaluation Center for testing the efficacy of new systems within a mission context. The purpose of MBT&E is to answer the question: “Does the system provide what the war fighter needs to accomplish the mission?” By refocusing the purpose of test and evaluation on the needs of the war fighter, test and evaluation gains relevance to real world conflict and increases confidence that newly fielded systems can perform in operational environments.

We propose to apply MBT&E concepts to testing unmanned and autonomous systems in order to demonstrate their relevance to the warfighter. By integrating postulated tasks to notional components, critical mission essential tasks, functions and components can be identified for hardening and reinforcement.

6. REFERENCES

[1] Sheehan, J., Deitz, P., Bray, B., Harris, B., Wong, A. The Military Missions and Means Framework, Interservice/Industry Training, Simulation and Education Conference, 2003.

[2] Cavaleri, Zehr,Minchew, Kearley, and Smits, Testing in a Joint Environment: A Mission and Means Framework Application Case Study, presentation to Office of Secretary of Defense, 2003.

[3] Wilcox, Christopher, Mission Based Test and Evaluation Tutorial, 25th Annual National Defense Industrial Association, Test and Evaluation Conference, March 2009.

Modeling and Simulation for Unmanned and Autonomous System Test and Evaluation

Mauricio Castillo-Effen
General Electric Global Research
1 Research Circle
Niskayuna, NY 12309, USA
Castillo-Effen@GE.com

Nikita Visnevski
General Electric Global Research
1 Research Circle
Niskayuna, NY 12309, USA
Nikita.Visnevski@GE.com

Raj Subbu
General Electric Global Research
1 Research Circle
Niskayuna, NY 12309, USA
Subbu@GE.com

ABSTRACT

Test and evaluation may be viewed as a technology enabler for the successful deployment of unmanned vehicles and robots in all their envisioned applications. It is however a challenging endeavor, considering that roboticists and developers are not used to thinking of test and evaluation as an integral component of robot development. Moreover, the community who has conducted test and evaluation up to this date does not possess the tools to cope with the growing complexity of unmanned and autonomous systems. This paper proposes a solution to one of the hardest problems in testing and evaluation of robots: test planning. The approach put forward relies on constructive simulation tools and on new techniques for searching in high dimensional spaces. The goal of the test planner is to generate a set of tests that make highly efficient use of resources to unveil weaknesses of the system under test. A secondary objective of the paper is to create reciprocal awareness between test and evaluation and robotics communities, who could benefit significantly from each other.

Keywords

Robotics, Unmanned and Autonomous Systems, Modeling and Simulation, Test and Evaluation

1. INTRODUCTION

Several technology roadmaps that have been published lately, e.g.: [8], [14], [15], etc. foresee a more predominant role of robotics[1] in several aspects of human society in the coming years. Furthermore, most of these reports agree on the fact that robotics represents a significant and growing commercial market. The main goal of these roadmaps is to guide policy makers and to focus research efforts. However, it is almost perplexing, how key technologies and technology enablers are easily overlooked. The "challenges" in autonomous driving sponsored by DARPA have been regarded as major successes. However, it is unclear when and how these technologies will be used in common day driving. According to the Computing Community Conssortium's (CCC) robotics roadmap [8], within the next 5-year time frame, autonomous vehicles will demonstrate driving safety "comparable to a human driver." The authors cautiously add: "with clearly lit and marked roads." This exemplifies the gaps and ambiguities that plague robotics. First, to the best of knowledge of the authors, a well-defined methodology that will allow for assessing "driving safety comparable to a human driver" in autonomous systems is nonexistent today. Furthermore, the disclaimer specifying the conditions under which the system will work, creates ambiguity that confounds someone who may want to acquire a vehicle with autonomous driving capability.

The example above illustrates well current challenges that stand before acquisition organizations procuring Unmanned and Autonomous Systems (UAS) for the government, particularly in the military domain. On the one hand, UAS are becoming key assets in modern military operations. There is significant pull from the warfighter requiring technology that provides relief from "dangerous, dirty, and dull" tasks. On the other hand, developers are trying to satisfy those needs with unmanned vehicles that are increasingly capable and sophisticated. Between these two parties stands the Defense Acquisition System with the Test and Evaluation (T&E) community supporting it. They have the mission of expediting the transfer of those technologies to the field, while making sure that the acquired assets fulfill the needs of the end-user. Higher levels of autonomy, emergent behavior, heterogeneous forms and evolving levels of cognition are some of the new challenges that need to be tackled by the T&E community when dealing with UAS. These are all new challenges for a community who had established procedures, facilities and methodologies for testing against well-defined and very specific requirements. The paper by Macias expounds the problem of UAS T&E (UAST) in full length [9].

Limitations of current T&E practices do not nullify its validity. T&E represents the feedback element that provides knowledge for timely risk mitigation during the development cycle of any system [6]. Hence, the development of robots could benefit from such knowledge. The authors of this pa-

[1] The term "robots" in the context of this paper refers to a variety of autonomous agents with the ability to interact with their environment, with humans, and eventually with other robots.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
Copyright 2009 ACM 978-1-60558-747-9/09/09...$10.00

per argue that T&E represents a key, highly underrated, and neglected technology enabler. Particularly, in robotics, there is the need for metrics to quantify robot capabilities and effectiveness. Then, complementary to metrics, there is the need for principled approaches to the generation of procedures altogether with facilities and personnel requirements for Test and Evaluation of robotic capabilities. Improved T&E capability will be followed by a smoother transition of robots from the developers' laboratories to the field. As a consequence, society will start benefitting from robotics technology much sooner, rather than waiting for its maturation, or worse, being afraid of it.

Recently, a growing tendency has emerged promoting the use of modeling and simulation (M&S) to support the development of robot software. Tools such as Microsoft Robotics Developer Studio, Player/Stage/Gazebo, Webots, USARSim, etc. provide simulated environments where developers test their control algorithms as they are developed. The advantages of this approach to robot software development are evident, among them:

- There is no risk that the robot will cause damage to people, to its environment, or to itself.
- This approach is highly efficient in terms of time and resources.
- Minimal requirements in terms of facilities and test equipment.
- Code can be used with almost no modifications in the real platform.

Simulations consist of experiments created by the developers to verify proper function of their code. However, a visible deficiency of these tools is the lack of a connection to concepts from dependable computing, verification and validation, formal methods, etc. These are all key aspects to consider when robots are viewed as safety critical systems. The military community has gathered significant experience and knowledge in these areas, which could be assimilated by the robotics community in order to generate more systematical approaches to the development of robots. Conversely, the military community may benefit from the knowledge stemming from the robotics community, because roboticists are the ones who advance technology underlying unmanned and autonomous vehicles used in the battlefield.

Test planning is one of the key challenges that need to be addressed by the UAST and robotics communities. More "intelligent" test planning capability may lead to a more agile T&E process and to a more efficient utilization of resources. The main goal of this paper is to introduce an architecture that leverages upon M&S for automated test planning. This confers its users the ability to use M&S tools to fully understand the main factors that affect the measures of effectiveness and to identify the most relevant tests which may be used in different phases of the T&E process.

Section 2 introduces important terminology, concepts and principles related to M&S and T&E. Section 3 addresses the need for metrics, and it presents the mission-based capability-driven T&E and the mission and means framework as useful tools for finding metrics and measures of effectiveness. Section 4 elaborates on the process of test planning in general. Section 3 presents a concrete instance of an architecture for automated test planning with its major components. Section 6 explains the main rationale guiding the decisions to select an appropriate modeling and simulation tool. Finally, Section 7 summarizes the main ideas introduced in this paper, and it lists a series of conclusions.

2. MODELING AND SIMULATION IN UAST

One of the objectives of this section is to establish a clear distinction between the concepts of modeling and simulation and test and evaluation, as well as their relationships. A number of definitions are introduced in this section to establish proper terminology usage and the conceptual background underlying the work presented in this paper. These definitions have been extracted textually or adapted from [6], [5], and [4].

Test and Evaluation (T&E). "Process by which a system or components are exercised and results analyzed to provide performance-related information. T&E enables an assessment of the attainment of technical performance, specifications, and system maturity to determine whether systems are operationally effective, suitable and survivable for intended use, and/or lethal."

Developmental T&E (DT&E). Encompasses all T&E activities that take place while the system is still being developed.

Operational T&E (OT&E). It "is conducted to evaluate system operational effectiveness, suitability and survivability in support of the full-rate production decision review."

Modeling and Simulation (M&S). The processes by which simplified representations of reality (*models*) are used to predict how systems might perform or survive under various conditions or environments.

Live Simulation. "A simulation involving real people operating real systems." This definition assumes that "people operate systems," which is not general enough to accommodate autonomous systems. This category could consider mock operations where real autonomous systems are used. This type of simulation is the most demanding in terms of resources, range safety, instrumentation, etc. Although the simulation results may be considered realistic because they involve the actual physical system, the simulated operations are designed with many constraints. As stated by a test pilot: "in live exercises we simulate; in simulation, we actually do things."

Virtual Simulation. "A simulation involving real people operating simulated systems." For instance, this category applies to different forms of flight simulators operated by real pilots for training purposes. Extending this definition to the autonomous domain, it could be interpreted as a mode of operation where autonomous systems operate in a virtual world, that is, a form of hardware-in-the-loop simulation. Here, the control computer of a robot could be interfaced to a virtual world. The robot gets synthetic stimuli from the simulated environment. Conversely, signals generated by the robot controller affect the simulated environment.

Constructive Simulation. "Models and simulations that involve simulated people operating simulated systems. Real people stimulate (make inputs) to such simulations, but are not involved in determining the outcomes." In the autonomous case, complete systems are simulated in the virtual environment. This is the main simulation type addressed in this paper.

Accreditation. "The official certification that a model, simulation, or federation of models and simulations and its associated data are acceptable for use for a specific purpose."

Verification. "The process of determining that a model implementation and its associated data accurately represents the developer's conceptual description and specifications."

Validation. "The process of determining the degree to which a model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model."

VV&A. Verification, validation and accreditation. All three are necessary for models to have relevance if used in the context of T&E.

The Defense Acquisition Guidebook (DAG) [6] acknowledges the value of M&S in T&E recognizing it as an "essential tool in achieving both: an effective and efficient T&E program." The DAG also points to the pitfalls, particularly to the limitations of M&S when dealing with systems that are not understood well. The DAG recommends a philosophy of interaction between T&E and M&S. According to this philosophy, M&S provides predictions of system performance, effectiveness, suitability, and survivability. On the other hand, T&E provides empirical data to confirm those predictions. Empirical data is used to refine the models. Then, the cycle is repeated, as depicted in Figure 1. The DAG also highlights the need for accreditation of all all M&S by the intended user. Accreditation can only be achieved through a robust verification, validation, and accreditation (VV&A) process.

The main conclusion that may be drawn from previous analysis is that M&S and T&E are two different complementary concepts. Pure physical T&E does not lead to efficient use of resources. On the other hand, pure M&S lacks relevance. Models can be validated and accredited through T&E. Hence, there is a continuum of simulation categories between purely physical (live simulation) and purely virtual (constructive simulation), with different "shades" of mixed categories (virtual simulation) in between. T&E is the enabler that allows shifting focus from live simulations which are most demanding in terms of resources towards constructive simulations. The most complex tests at the "Systems of Systems" (SoS) level are basically impossible to be executed as live simulations. However, complex constructive simulations can only be validated if smaller subsystems have been validated and accredited through T&E.

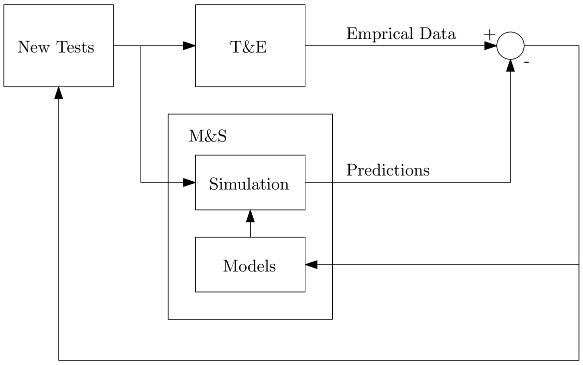


Figure 1: Model-test-fix-model philosophy [6]

While useful, the DAG recommendations and concepts are not specific and practical enough to answer key questions that arise in UAST and robotics in general. Some of them are: how to design tests?, how to analyze empirical data?, finding appropriate metrics, etc. This paper proposes practical alternatives to fill some of those gaps.

3. MISSION-BASED CAPABILITY-DRIVEN UAST

The main question addressed in this section is the one pertaining to metrics. Real-life cases have demonstrated that the traditional approach to T&E based on the verification of performance requirements do not work properly for complex systems and that a paradigm shift is necessary. An example that is often cited to support this notion is the Predator MQ-1 UAS, which failed operational T&E but proved extremely useful on the battlefield [9]. The case of the Predator proves that metrics for complex systems need to be tied to measures of effectiveness (MoE) and not necessarily to measures of performance (MoP). Frameworks such as the Mission and Means framework (M&M) can help in establishing a hierarchical relationship between mission effectiveness, tasks, capabilities and system components [7]. By using this framework it is also possible to trace back mission success or failure to specific tasks, to capabilities, and to components.

The RoboCup Virtual Rescue competition represents a good test-case, where mission-based capability driven UAST could be applied [1]. Although the scenario is simulated, metrics such as number of victims found within certain time or energy constraints are related directly to measures of effectiveness. The mission is composed of a number of tasks such as: exploring, searching for victims, reporting victims, etc. Similarly, tasks may be performed only if certain capabilities are present, for instance: an appropriate locomotion system, collaborative search capability, localization, navigation, etc. Another specific example illustrating the application of the M&M framework is presented in [7].

4. TEST PLANNING

It was mentioned before that the block "New Tests" in Figure 1 is not elucidated fully in [6]. The process of devising tests, called test planning, is one of the main challenges of UAST and robotics in general. Traditional approaches based on knowledge and experience are riddled with limitations, particularly, considering the degree complexity of UAS. To guarantee that tests are truly relevant and efficient, a formal approach to test planning is necessary. Statistical techniques such as Design of Experiment (DoE) have been already proposed as viable tools [3]. The authors adhere to that notion and to the view of T&E of robots as a DoE problem. The main objective is to obtain a minimal set of experiments which yields the maximum information with respect to the hypothesis that need to be tested. To maximize the efficiency of physical tests in terms of time and resources, test planning is crucial. The only way to avoid the curse of dimensionality in designing experiments with a large number of independent variables is by including as much knowledge of the process as possible. This is where modeling and simulation is most valuable. M&S can be seen as the main vehicle for incorporating knowledge about the system.

M&S shall be used as a tool for the generation of a set of experiments. Each experiment corresponds to a specific selection of independent variables X_i. Two kinds of independent variables may be distinguished:

Intrinsic variables This refers to parameters of the UAS or team of UAS. Some examples of this type of variables are:

- Physical properties of the UAS or UAS team, such as: sensor accuracy, turning radius, actuator power, number of team members, maximum speed, etc.
- Behavioral properties and their parameters, such as: localization algorithm, software architecture (reactive, deliberative, hierarchical, etc.), navigation algorithm, world model representation, etc.

Extrinsic variables These are all external factors that affect UAS behavior including initial conditions. Some examples of external factors are temperature, humidity, wind, lighting conditions, initial state (e.g.: position), terrain type, obstacles, etc.

As seen from the examples, independent variables may be continuous or discrete. The outcome of an experiment may be measured through metrics $\mathcal{M}_j(E)$, which correspond to measures of effectiveness, as explained previously. Given the stochastic nature of the problem, dependent variables are actually probabilities of the form $P[\mathcal{M}_j(E) > \mathcal{M}_{\text{Th}}]$, where $\mathcal{M}_{\text{Th}}$ is a threshold value used to measure mission failure (or success). An experiment E_k may be considered an n-tuple, which is a combination of independent variables X_i

$$E_k \equiv \langle X_1, X_2, \ldots, X_N \rangle; E_k \in \mathbb{R}^N \tag{1}$$

Hence, the goal of the test planner is to search in the N-dimensional space of experiments for those experiments that have a specific probability with respect to a metric $\mathcal{M}(E)$. The search could be also extended to consider multiple simultaneous metrics. The search is conducted with the help of a simulation engine, which has encoded a number of rules of interaction among entities and between entities and the environment, both defined by models. Thus, the simulation engine is used to determine the specific outcomes $\mathcal{M}(E_k)$. The actual determination of the sought experiments E_k^* is done through a search technique $\mathcal{S}$, which determines the set of experiments fulfilling the conditions corresponding to the probability of a specific metric having a concrete threshold value.

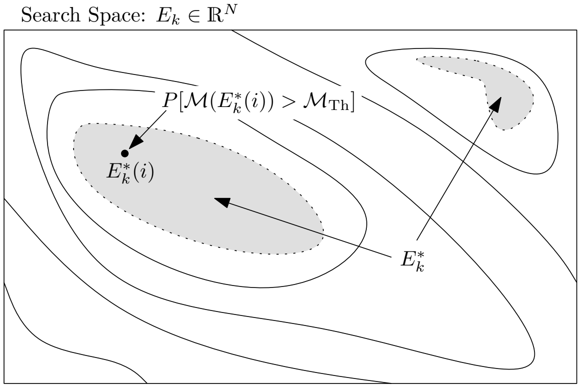

Search Space: $E_k \in \mathbb{R}^N$
$P[\mathcal{M}(E_k^*(i)) > \mathcal{M}_{\text{Th}}]$
$E_k^*(i)$
E_k^*

Figure 2: Search space and sought set of experiments

$$E_k^* = \{E_k : P[\mathcal{M}(E_k) > \mathcal{M}_{\text{Th}}]\} \tag{2}$$

The search processes may be better understood through Figure 2. The rectangle represents the whole search space. Gray zones surrounded by dotted lines represent the sought set of experiments E_k^*. The other continuous lines represent boundaries of zones where the probability for a particular value of the metric $\mathcal{M}(E_k)$ has a specific value. In practice, an analytical solution describing E_k^* cannot be obtained. Therefore, the final E_k^* will be a sample of this set.

5. AUTOMATED TEST PLANNER

This section proposes a concrete instance of an architecture for test planning according to the principles presented in previous section. The main elements of the planning process are depicted in Figure 3. The prime objective of this system is to find the set of experiments E_k^* in an iterative process viewed as an automated test planning process or, also as an automated DoE generator. Currently, concrete implementations of each of the functions shown in Figure 3 are being developed and integrated in the Cognitive Autonomous Systems Testing (CAST) project sponsored by the DoD Test Resource Management Center's Unmanned and Autonomous System Test (UAST) focus group, in collaboration with the White Sands Missile Range and the Army Research Lab.

Two main groups of blocks may be distinguished: group (1), consisting of simulation engine with model libraries (white blocks), and group (2) with what could be viewed as the planner itself (gray blocks). Details about group (1) are

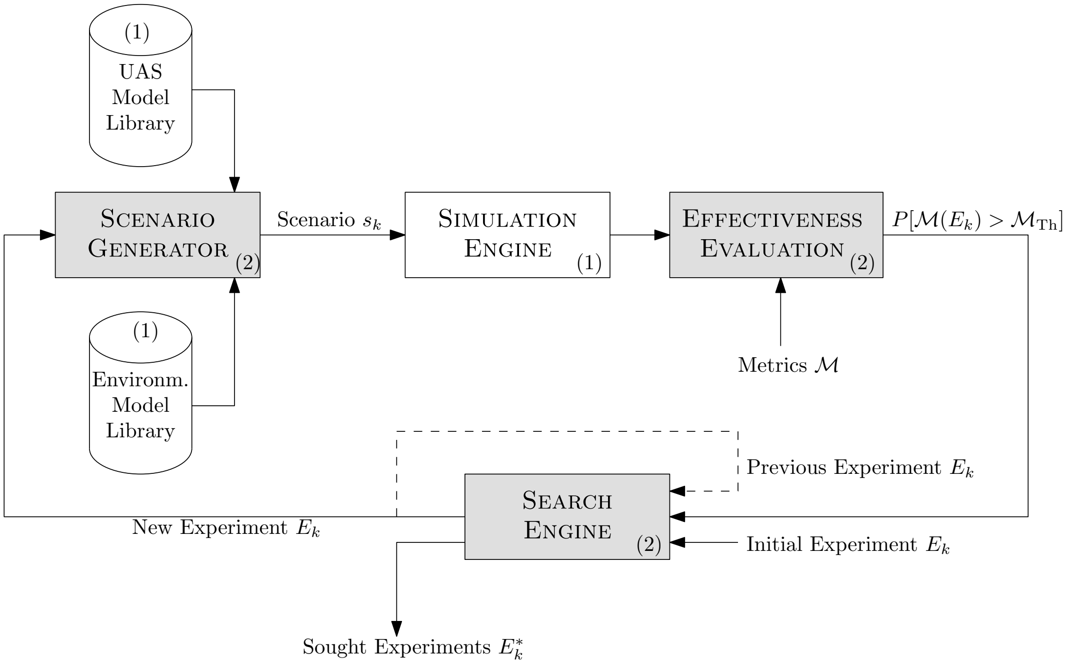


Figure 3: Automated planner architecture

provided in the following section. The planner, i.e.: group (2), consists of three main elements:

Scenario generator. This component accesses libraries of models, which should be verified, validated, and accredited. There are two libraries of such models: one library with models of UAS and models of other entities relevant to the missions being simulated, and another library with models of the environment. Experiments generated by the search engine altogether with models are used to generate so-called *scenarios*. Scenarios are understood by the simulation engine, which is able to execute them.

Effectiveness evaluator. The effectiveness evaluator uses metrics defined by the testers to evaluate the probability of mission success/failure. Since the simulation engine is stochastic, the outcomes for a certain experiment may vary for different iterations. Thus, the effectiveness evaluator needs to run one experiment several times to obtain probabilistic measures. The number of iterations depends on the computational capability available to the automated planner.

Search engine. The search engine may be considered the core of the automated test planner. The search may start from a set of randomized experiments. The main function of the search engine is to generate a new set of experiments using previous experiments and their outcomes.

The overall function of the three components is to generate scenarios with increasing difficulty for the systems under test. Hence, only extrinsic independent variables may be manipulated. The resulting experiments E_k^* correspond to scenarios yielding a high probability of mission failure. In CAST, evolutionary computation techniques are used to implement the search engine. They are explained in further detail in [16]. All three functions are implemented in MATLAB, which interfaces with the simulation engine through an Application Programming Interface (API). This API is a customization of the API provided with the simulation engine by the vendor.

6. SIMULATION ENGINE

In this section, the rationale are introduced for selecting a simulation engine suitable for test planning according to the general principles explained in Section 4, and to the architecture illustrated in Section 5. What are the main requirements that need to be considered when selecting a simulation engine suitable for designing robot tests?. Here is a non-exhaustive list of key features to consider:

Modeling flexibility. This refers to the ability to incorporate models with varying levels of fidelity. This is useful for covering all phases of the T&E process (DT&E, OT&E, etc.). When the system is under development, some subsystems are still being developed; therefore, they do not have corresponding models. This is the case when abstract generic surrogates need to be used. For example, in case SLAM algorithms to allow for localization and mapping have not yet been implemented, knowledge of position may be supplied directly from the actual state of the entity by the simulation engine. At a later point, when SLAM algorithms need to be tested, the surrogate localization capability may be replaced with the actual SLAM solution. The engine should also allow for instantiating several forms of the same functionality.

Built-in functionality. M&S-based test planning may quickly become a major software development effort. To avoid this, it is important that the engine already incorporates simple models of interaction and perception, which could be used as surrogates in the initial phases of DT&E. This also applies to robot and behavior of other entities. For instance, if a robot algorithm is tested for roaming in crowded spaces, it is important that the simulation engine has default realistic behaviors for the people who act as obstacles to the robot. The effort of developing test scenarios for robot navigation should not become a major effort in developing the dynamic obstacle behavior.

Handling large-scale simulations. Major challenges in T&E stem form the fact that it is envisioned that UAS may be more effective when operating as cooperative teams. Hence, simulations may involve significant numbers of UAS. Consider for instance the current vision of employing swarms of micro aerial vehicles. Simulations can quickly become a serious computational challenge to any centralized simulation engine. Therefore, the ability to perform distributed simulation through the use of distributed computing techniques should be considered.

Fast-time simulation support. Many simulators are designed only for real-time simulation. They are not very useful for search techniques that need to run hundreds or thousands of these simulations in the shortest time possible. Hence, the ability to run scenarios in faster-than-real-time is essential for automated T&E planning.

Visualization. When performing search, the simulator should be used in fast-time mode and only a few status messages may be displayed to the user. However, when the search engine converges to a specific set of scenarios, the tester may want to perform a qualitative assessment of their validity. This is done best using some form of realistic visualization, where the scenarios are "played" in a 3-D visualization environment with interactive viewing control. Sometimes, to have a proper overview of the test scenario, 2-D visualizations are also extremely helpful.

Extensibility. Since simulation engines have been developed for different uses, they lack functionality needed specifically for T&E. Therefore, the engine needs to be extendable by the user with elements that are needed in specific domains. For instance, aerial vehicles may need high-fidelity aerodynamic behavior, which is not part of the basic engine functionality. Similarly, simulation of RF-propagation is another feature, which may not be part of the simulation engine but which should be addable by the user.

Interfacing to other applications. Since the user may already have tools developed in other environments, it is also important that the simulation engine has the capability to interface easily with other applications. For example, in the architecture presented in previous section, the search engine, performance evaluation and scenario generation modules are implemented in MATLAB. Seamless interfacing between the simulation engine and MATLAB is necessary. Interfacing to other applications may be simplified when the simulation engine is provided with a well-documented API.

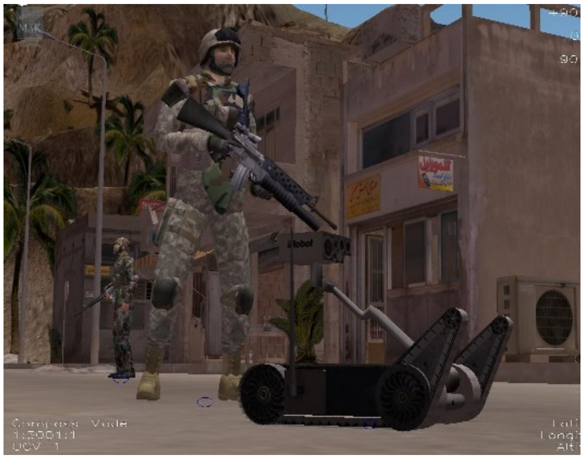


Figure 4: 3-D interface to simulation engine

Reliability. Although stochastic variability is a desired feature, it needs to be under control of the user. For example, simulations may be triggered to exhibit stochasticity by changing seeds of random generators in the different scenario executions. However, if seeds are kept the same, the simulation engine should always yield the same results.

The simulation engine selected for the CAST project is VR-Forces from MAK technologies [13]. VR-Forces complied with most of the requirements listed above. VR-Forces itself is the simulation engine ("back-end"), which has been developed with the main objective of enabling distributed simulation by making use of standard distributed simulation protocols such as DIS and HLA. This feature is ideal for implementing large scenarios with many participating entities distributed across several networked computers. For visualization ("front-end"), there are two types of GUIs which could be selected, one with 2-D representation and one with 3-D representation, as seen in Figure 4.

Out-of-the-box functionality of VR-Forces is relatively sophisticated, and it includes a number of features, such as: models of interaction between entities, basic models of perception, rules of engagement, ability to incorporate plans, path planning capability and sophisticated individual and group behaviors enabled through the add-on B-Have [11]. Currently, the authors are engaging with researchers from the Army Research Lab (ARL) with the purpose of obtaining validated models of real UAS. Meanwhile, the automated test planning capability is developed using models that are well-known in the robotics community [10].

The VR-Forces API has been used extensively to allow for seamless connectivity between MATLAB and the simulation back-end. As mentioned before, MATLAB is used as the main prototyping tool for algorithms related to test planning capability. The effort of interfacing these two elements has two main work areas: enabling functionality for configuring scenarios, and enabling functionality for metric evaluation by obtaining simulation information from the so-called "state repository" [12].

7. CONCLUSIONS

The robotics field stands at a crossroads. The exploratory phase with some of its successes has created great expectations. But it seems that it has also reached a critical point. It could be even said that, to a certain extent, the robotics field has outpaced itself. The time has arrived when it is necessary to assess what has been accomplished and the path forward. Sound scientific principles need to be brought back to the practice of the robotics discipline to solidify its foundations. This fact has been acknowledged in several circles in the robotics community. It has motivated the organization of special interest groups and workshops such as "Good Experimental Methodology in Robotics" [2], "Performance Metrics of Intelligent Systems," etc.

Society needs robotics technology, but it cannot take the risk of accepting systems that do not offer any guarantees with respect to their regular operation and safety. Testing and evaluation offers an opportunity to think about effectiveness, safety, reliability , etc. of robotic systems. Furthermore, it also bring a wealth of knowledge and experience accumulated over the years in the practice of T&E of military systems. It is in the hand of roboticists to use this experience and to contribute to its further development.

This paper has presented some concepts on using M&S as an efficient way to plan and generate tests of UAS. Beyond that, it has also put forward specific alternatives on how to put those concepts into practice. While there is still a long path to travel before reaching a seamless synergy between M&S and T&E, this work could be viewed as an initial effort towards that goal. It is expected that the concepts and tools proposed here will be refined and expanded gradually with the demands of the users of robotics technologies, and with the efforts of robot developers from all backgrounds: academic, industrial, military, etc.

8. ACKNOWLEDGMENTS

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. This work is funded by the T&E/S&T Program through the White Sands Missile Range, NM, contract W9124Q-08-C-0562.

9. REFERENCES

[1] S. Balakirsky, C. Scrapper, S. Carpin, and M. Lewis. USARSim: Providing a Framework for Multi-robot Performance Evaluation. In *Performance Metrics of Intelligent Systems PerMIS'06*, 2006.

[2] F. Bonsignorio, A. D. Pobil, and J. H. South. Defining the requisites of a replicable robotics experiment. In *Workshop on Good Experimental Methodology In Robotics, at RSS 2009*, Seattle, WA, June 2009.

[3] M. L. Cohen, J. E. Rolph, , and D. L. Steffey, editors. *Statistics, Testing, and Defense Acquisition: New Approaches and Methodological Improvements*. National Academy Press, 1998.

[4] Defense Acquisition University. *Glossary of Defense Acquisition Acronyms & Terms*. Defense Acquisition University Press, 12th edition, July 2005.

[5] Department of Defense. Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A). DoD Instruction 5000.61, May 2003.

[6] Department of Defense. *Interim Defense Acquisition Guidebook*. https://acc.dau.mil/dag, June 2009.

[7] P. A. Djang and F. Lopez. Unmanned and Autonomous Systems Mission Based Test and Evaluation. In *Performance Metrics of Intelligent Systems PerMIS'09*, Gaithersburg, MD, September 2009. NIST.

[8] Henrik I. Christensen et al. A Roadmap for US Robotics – From Internet to Robotics. Technical report, The Computing Community Consortium – Computing Research Association, http://www.us-robotics.us, May 2009.

[9] F. Macias. The Test and Evaluation of Unmanned and Autonomous Systems. *ITEA Journal*, 29(4):388–395, December 2008.

[10] R. Madhavan, E. R. Messina, and J. S. Albus, editors. *Intelligent Vehicle Systems: A 4D/RCS Approach*. Nova Science Publishers, 2007.

[11] MAK Technologies – VT Systems, Cambridge, MA 02138 USA. *B-HAVE Module for VR-Forces, User's Guide*, revision bhv-1.2-1-080214 edition, 2008.

[12] MAK Technologies – VT Systems, Cambridge, MA 02138 USA. *VR-Forces – Developer's Guide*, revision vrf-3.11-2-080103 edition, 2008.

[13] MAK Technologies – VT Systems, Cambridge, MA 02138 USA. *VR-Forces User's Guide*, revision vrf-3.11-1-080125 edition, 2008.

[14] Maurits Butter et al. Robotics for Healthcare – Final Report. Technical report, European Commission, DG Information Society, October 2008.

[15] Office of the Secretary of Defense. FY2009-2034 Unmanned Systems Integrated Roadmap. Technical report, Department of Defense, United States of America, April 2009.

[16] R. Subbu, N. A. Visnevski, and P. A. Djang. Evolutionary Framework for Test of Autonomous Systems. In *Performance Metrics of Intelligent Systems PerMIS'09*, Gaithersburg, MD, September 2009. NIST.

Evolutionary Framework for Test of Autonomous Systems

Raj Subbu
General Electric Global Research
1 Research Circle
Niskayuna, NY 12309
(518) 387-6457
subbu@research.ge.com

Nikita Visnevski
General Electric Global Research
1 Research Circle
Niskayuna, NY 12309
(518) 387-4385
visnevsk@research.ge.com

Philipp Djang
Army Research Lab
ARL SLAD IEPD
White Sands MR, NM 88002
(575) 678-1550
djang@arl.army.mil

ABSTRACT

A DoD mission and challenge is to enable a high percentage of autonomous vehicles in the warfighter fleet by 2015. These systems will need to display a high degree of autonomous capabilities. The capabilities of these autonomous systems must be acceptable to the warfighter and his/her logistical support structure. Autonomous systems of the future will need to be tested so their mission capabilities and robustness are predictable to the warfighter. The principal challenge therefore is the set of test strategies for these future autonomous systems. The goal of the test community is that these autonomous systems be broadly accepted to seamlessly operate either independently or as part of a human-in-the-loop system. Our goal is to develop an efficient intelligent test process that will enable the rapid introduction of autonomous systems on the battlefield. We propose a novel war game simulation-based multi-objective evolutionary test framework that combines the elements of testing an autonomous system's mission execution capabilities as a function of its innate capabilities and evolutionary computation.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search — Heuristic methods; I.2.9 [**Artificial Intelligence**]: Robotics — Autonomous vehicles; I.6.7 [**Simulation and Modeling**]: Simulation Support Systems — Environments; J.7 [**Computers in Other Systems**]: Military.

General Terms

Algorithms, Measurement, Performance, Experimentation, Verification.

Keywords

Autonomous Systems, Test and Evaluation, Evolutionary Algorithms, Multi-objective Optimization, Tradeoff frontier, War-game Simulation.

1. INTRODUCTION

A key DoD mission is to enable a high percentage of autonomous vehicles in the warfighter fleet by 2015. This is being driven principally to reduce risk exposure to combat troops in hazardous conditions, and for advance reconnaissance and threat neutralization. These vehicles will need to display a high degree of autonomous capabilities. Further, their capabilities must be acceptable to the warfighter and his/her logistical support structure. However, current DoD test and evaluation capabilities and methodologies while sufficient for tightly tethered human-in-the-loop systems are insufficient for the mission certification of complex autonomous systems operating in non-deterministic environments [1]. Autonomous systems of the future will need to be tested so their mission capabilities, robustness, and failure modes are predictable to the warfighter. The principal challenge therefore is the set of scalable test strategies for these future autonomous systems. The goal of the test community is that these autonomous systems be broadly accepted to seamlessly operate either independently or as part of a human-in-the-loop system and scale from small to large deployments. Our goal is to develop an efficient intelligent test process that will enable the rapid introduction of autonomous systems on the battlefield.

We propose a novel war game simulation-based test framework that utilizes evolutionary algorithms for identifying the mission failure modes. While the traditional application of evolutionary methods is for efficient synthesis or design, we propose the use of these methods for the efficient identification of failure scenarios from a mission satisfaction perspective. This approach combines the elements of testing an autonomous system's mission execution capabilities as a function of its innate capabilities and evolutionary computation. In this paper, we present the evolutionary test framework, preliminary experimental results based on a limited scale war game, and ideas for developing this work into a deployable mission based test and evaluation framework [2].

This research effort is being conducted under the auspices of the DoD Test Resource Management Center's Unmanned and Autonomous System Test (UAST) focus group, and in collaboration with the White Sands Missile Range and the Army Research Lab.

2. RELATED WORK

In this section, we present background and briefly review the literature in complex adaptive systems, evolutionary multi-objective optimization, and search-based systems test.

2.1 Complex Adaptive Systems

Complex adaptive systems constitute a dynamic network of diverse and adaptive systems. The paradigm originally coined at the Santa Fe Institute has been used to model disease propagation, financial markets, and economic networks [3].

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.

The complex adaptive systems paradigm has been implemented in a number of military simulations in the form of agent based models. Structurally, a set of combat agents interact with one another in a simulated battlefield. A combat agent is an autonomous computational entity with an internal state and associated decision-making process implemented as a set of rules governing tactical behavior [4]. The agent's state is usually represented as a dynamic vector describing metrics such as agent position, identity, current functionality, and so on. Combat agents can interact with one another by passing messages between themselves, which can represent communication, cooperative actions, or conflict. Given these elements, a military-domain agent-based model is then a collection of interacting combat agents instantiated within a virtual "artificial world" that contains a terrain-based environment within which the agents function as well as contend with other hostile combat agents. Existing agent-based models of land warfare such as Irreducible Semi-Autonomous Adaptive Combat (ISAAC) [5], Simulation of Information in Battlefield Decisions (SinBaD) [6], and AgentKit [7] all address the emergent behavior of combat units of interacting Blue and Red agents.

2.2 Evolutionary Multi-objective Optimization

Evolutionary Algorithms (EAs) have received a lot of attention for use in optimization and learning applications, and have been applied to various practical problems [8], [9]. In recent years, the area of evolutionary multi-objective optimization has grown considerably [10], starting with the pioneering work of Schaffer [11].

Most real-world optimization problems have several, often conflicting objectives. Therefore, the optimum for a multi-objective problem is typically not a single solution—it is a set of solutions that trade-off between objectives. The Italian economist Vilfredo Pareto first generally formulated this concept in 1896 [12], and it bears his name today. A solution is Pareto optimal if (for a maximization problem) no increase in any criterion can be made without a simultaneous decrease in any other criterion. The set of all Pareto optimal points is known as the *Pareto frontier* or alternatively as the *efficient frontier*. In the absence of further information, each such solution is as good as the others are when all objectives are jointly considered. Each solution on the Pareto frontier is not dominated by any other solution. Formally, given an n-dimensional measurable space whose elements can be partially ordered, a vector in this space $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is considered non-dominated if there exists no other vector $\mathbf{z}$ such that $x_i \leq z_i$ for all i, and $x_k < z_k$ for at least one $1 \leq k \leq n$. The symbol ≤ may be interpreted as "the right-hand-side of it *is as good as or better than* its left-hand-side" without loss of generality.

A review of mathematical programming-based optimization methods for multi-objective problems is presented in [13]. These techniques generally require multiple executions to identify the Pareto frontier, and may in several cases be highly susceptible to the shape or continuity of the Pareto frontier, restricting their wide practical applicability. An evolutionary multi-objective optimizer works by systematically searching, memorizing, and improving populations of vectors (solutions), and performs multi-objective search via the evolution of populations of test solutions in an effort to attain the true Pareto frontier. This characteristic allows finding an entire set of Pareto optimal solutions in a single execution of the algorithm. Traditionally, multi-objective optimization has been pursued via the application of single-objective optimizers to linearly (or nonlinearly) weighted and aggregated objectives, and repeating the optimization for multiple weight combinations. While this traditional approach appears satisfactory in practice, the method is unable to identify non-convex regions of the Pareto frontier [14]. This problem is more pronounced when the underlying models that represent mappings to multiple mutually competing output objectives are nonlinear.

Practical evolutionary search schemes do not guarantee convergence to the global optimum in a predetermined finite time, but they are often capable of finding very good and consistent approximate solutions. However, they are shown (theoretically and practically) to asymptotically converge under mild conditions [15].

2.3 Search-based Systems Test

Mission based testing involves the automated generation of a large number of realistic missions within a high-fidelity simulation environment to help identify scenarios that induce system failure. This concept is not only being adopted by the DoD and presented in this paper, but also by NASA for test and evaluation of autonomous deep space systems [16], [17]. We briefly but chronologically review the literature on the use of evolutionary algorithms for systems test.

Schultz et al. [18], [19] use a genetic algorithm for testing the controller of an autonomous system. Their approach subjects a controller coupled to a vehicle simulator to a combination of fault scenarios generated by a genetic algorithm. The genetic algorithm searches for those combinations of faults that produce degraded system performance. Corno et al. [20] use a genetic algorithm to generate test patterns for sequential digital circuits. Their genetic algorithm generates a sequence of values to be applied to the input pins such that the outputs of a fault-free circuit will be different from the corresponding ones of a faulty circuit. A generally similar approach to dynamic test pattern generation for software programs is presented by Michael et al. [21]. Harman et al. present a multi-objective genetic algorithm that can identify a "branch-adequate" test set for software programs. A branch-adequate test set includes at least one test case that can trigger the execution of at least one branch, and covers every possible branch in the program flow. Windisch et al. [23] present a particle swarm optimization approach that generates test cases for software programs. Nguyen et al. [24] present an evolutionary test method for autonomous distributed software systems such as web crawlers. Terrile and Guillaume [25] use evolutionary algorithms to search a space of possible behaviors to identify emergent behaviors that are unexpected or detrimental in spacecraft systems.

3. EVOLUTIONARY MISSION BASED TEST AND EVALUATION

The evolutionary mission based test and evaluation framework is based on the use of a multi-objective evolutionary algorithm (NSGA-II [26]) and MÄK VR-Forces [27], a simulation toolkit for generating and executing battlefield scenarios. In this simulation environment, various types of Blue (friendly) and Red (enemy) force compositions can be created and allowed to engage according to high-level directions. The engaging entities (such as tanks, infantry, and UAVs etc.) have configurable behaviors, and they behave as distributed autonomous agents in the battlefield scenario. A vignette of a battlefield and forces engagement scenario involving tanks is shown in Figure 1.

Figure 1: A vignette of a battlefield and forces engagement scenario involving tanks.

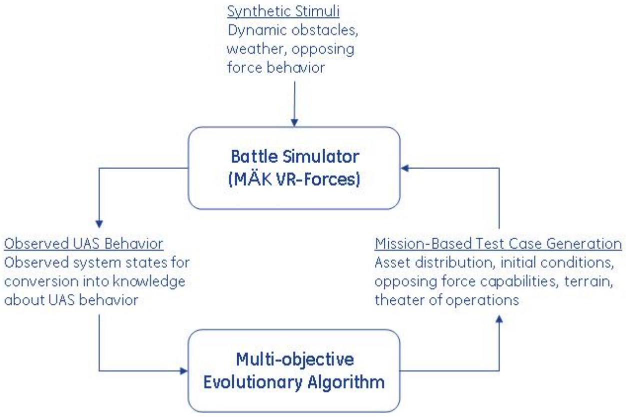

Synthetic Stimuli
Dynamic obstacles, weather, opposing force behavior
Battle Simulator (MÄK VR-Forces)
Observed UAS Behavior
Observed system states for conversion into knowledge about UAS behavior
Mission-Based Test Case Generation
Asset distribution, initial conditions, opposing force capabilities, terrain, theater of operations
Multi-objective Evolutionary Algorithm

Figure 2: Evolutionary framework for test of autonomous systems.

The architecture of the test and evaluation framework is shown in Figure 2. For a given mission, the multi-objective evolutionary algorithm intelligently generates a series of test cases that will drive the *mission to fail*. This approach of driving mission failure rather than mission success allows the identification of an autonomous system's failure modes. Each test case that the evolutionary algorithm generates varies the Blue and Red force asset distributions, initial conditions, opposing force capabilities, terrain, and theater of engagement, constituting a configuration. This configuration is executed to completion in the simulation environment. At the conclusion of each execution the system states and outcomes are observed to generate metrics feedback to the evolutionary algorithm. The simulation execution and outcome are influenced by factors such as dynamic obstacles, and behavior of opposing forces. The metrics of interest computed at the conclusion of each execution are *Loss Exchange Ratio* (LER)[1] and *Percentage of Healthy Enemy Forces* (PHEF)[2]. The evolutionary algorithm seeks to minimize both these metrics. Minimization of the LER metric rewards friendly losses, and minimization of the PHEF metric rewards enemy losses. Both these metrics oppose one another, and an optimal tradeoff frontier is identified. Such a tradeoff frontier allows the decision-maker to evaluate the conditions and outcomes associated with each of the tradeoff scenarios optimal from a mission dissatisfaction perspective.

4. EXPERIMENTAL RESULTS

In this section, we present preliminary experimental results based on a limited scale war game involving a set of M1A1 Blue tanks and T80 Red tanks. While the M1A1 tanks are more superior and powerful to the T80 tanks, the evolutionary algorithm drives the search to identify weak engagement scenarios for the M1A1 tanks.

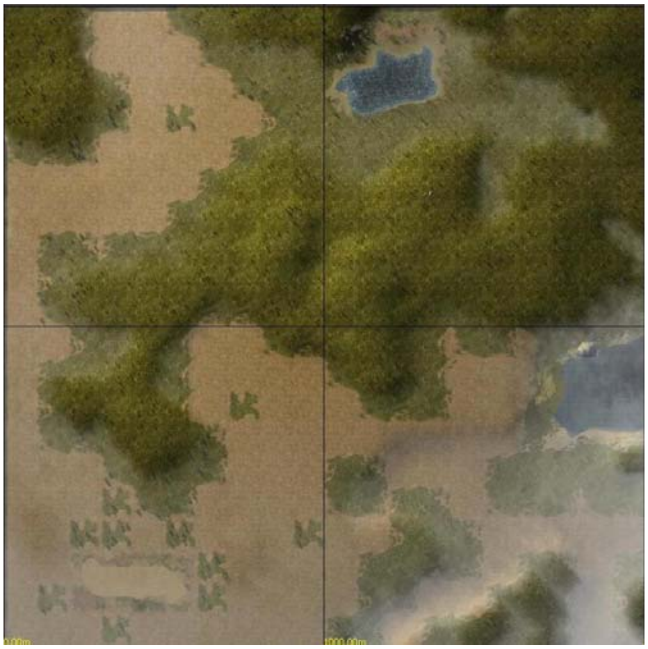


Figure 3: The simulated 2000 meter x 2000 meter battlefield terrain.

The simulated 2000 meter x 2000 meter terrain where the battle ensues is shown in Figure 3. The highest point on this undulating terrain with significant foliage (elevated green areas) is 66.5 meters with respect to a zero altitude (brown areas) baseline. A three dimensional view of a portion of this terrain is shown in Figure 4 for perspective. The layout of the forces engagement theater is shown in Figure 5. In this layout, the Blue and Red forces may originate from one of the four corners, but the Blue and Red forces may not originate from the same corner. The task for each opposing force is to reach and take a target point within the Blue-Red forces engagement zone. For these

[1] Ratio of number of damaged enemy assets to number of damaged friendly assets.

[2] Ratio of number of healthy enemy assets to total number of enemy assets.

preliminary experiments we lower-left biased the engagement theater due to the dense foliage in the upper right of the terrain and further due to the two lakes present. This way, all engagement takes places only on solid ground.

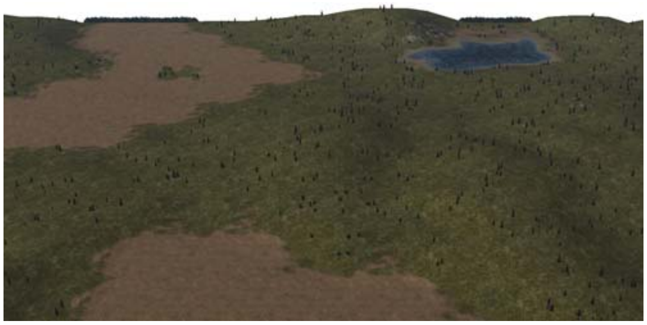


Figure 4: A 3D view of the battlefield terrain.

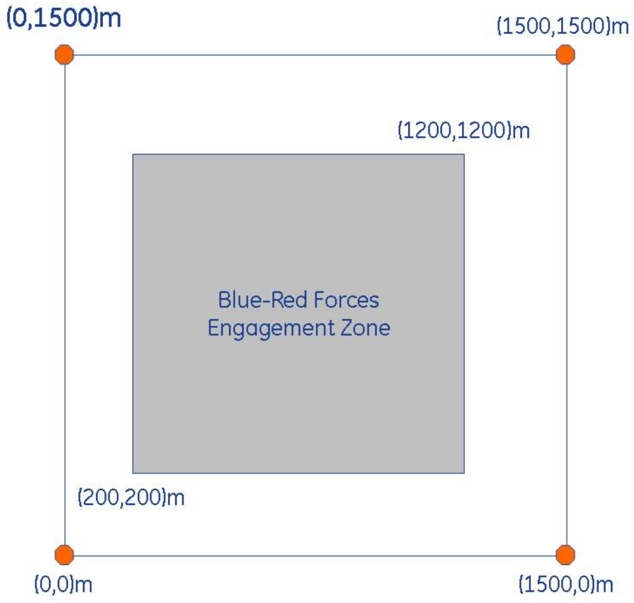


Figure 5: Layout of the forces engagement theater.

The multi-objective evolutionary algorithm varies the following parameters: Blue force size from 50% to 200% of the Red force size, which is fixed at ten T80 tanks; X-Y coordinates of the target point within the Blue-Red forces engagement zone; and the start locations of the Blue and Red forces with the constraint that Blue and Red forces cannot start from the same location. We use a population size of 20, and a total of 30 generations, for a total of 600 simulations. Each simulation executes for 3 minutes, resulting in a total execution time of 30 hours. The best tradeoff frontier at the conclusion of the evolutionary algorithm execution is shown in Figure 6. Each of the scenarios on the tradeoff frontier is shown superimposed on the terrain in Figure 7. The green circles correspond to the target points; the red squares show the starting location of the Red force; the blue square shows the starting location of the Blue force. Further, the Blue force size is at least 80% and up to 200% of the Red force size for each of the tradeoff scenarios.

The interesting observation is that as the overall goal is mission dissatisfaction, the Blue force is always selected to originate from the undulating upper right of the terrain, while the Red force is predisposed to originating from the more even lower left or right of the terrain. Further, there is a cluster of targets at the transition point from the higher to low altitudes which allows the Red force to take advantage due to the line of sight visibility constraints for the Blue force as they descend to the lower altitude terrain.

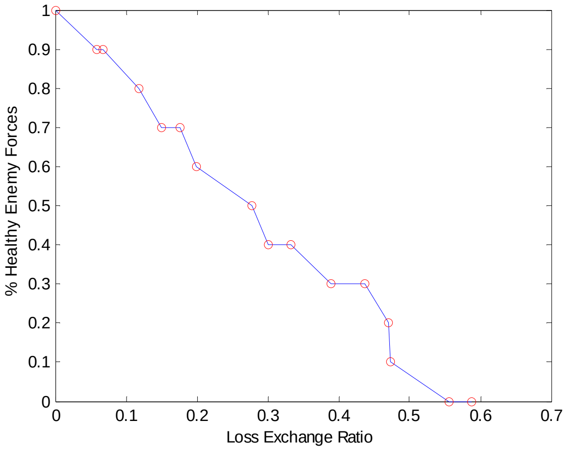


Figure 6: Best tradeoff frontier identified at the conclusion of the evolutionary algorithm execution.

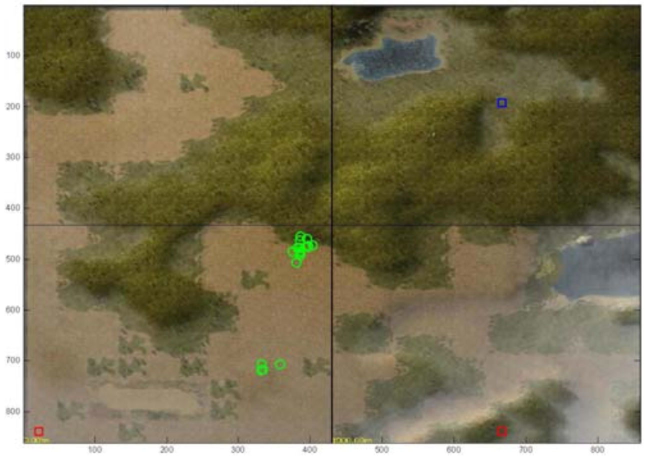


Figure 7: Best tradeoff frontier points shown superimposed on the terrain.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed and presented a basic framework for the intelligent test and evaluation of autonomous systems. This framework is based on the use of autonomous systems simulation tools and multi-objective evolutionary algorithms. The evolutionary algorithm identifies failure modes through intelligent search. We have presented preliminary experimental results based on a limited scale war game involving a set of M1A1 Blue tanks and T80 Red tanks.

Our goal is to develop an efficient and scalable test process that will enable the rapid introduction of autonomous systems on the battlefield. The basic framework presented in this paper in an advanced deployable form is expected to support the DoD mission and challenge to enable a high percentage of mission certified autonomous vehicles in the warfighter fleet by 2015. To achieve this goal several objectives need to be met. A first objective is experimentation with more complex battlefield scenarios involving force mixtures (tanks, infantry, aircraft etc.) and terrain. The next objective is speeding up the overall simulation execution time through high-performance hardware. The next objective is distributing the computation and leveraging a coevolutionary computational framework [28] as shown in Figure 8 to enable a scalable network-efficient search over large-scale battlefield scenarios. The long-term vision is to empower a decision-maker tasked with test of complex autonomous systems with a fast virtual system so these autonomous systems can be robustly mission certified to be broadly accepted and seamlessly operate either independently or as part of a human-in-the-loop system.

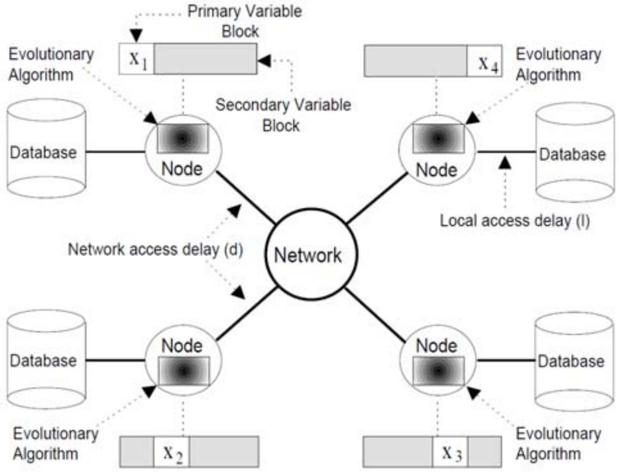


Figure 8: Distributed coevolutionary computation.

6. ACKNOWLEDGMENTS

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. This work is funded by the T&E/S&T Program through the White Sands Missile Range, NM, contract W9124Q-08-C-0562.

This research effort is being conducted in collaboration with the Army Research Lab.

7. REFERENCES

[1] F. Macias, “The Test and Evaluation of Unmanned and Autonomous Systems,” International Test and Evaluation Association Journal, 29: 388–395, 2008.

[2] P. Djang, and F. Lopez, “Unmanned and Autonomous Systems Mission Based Test and Evaluation,” Proc. of PerMIS'09, Gaithersburg, MD, USA, September 21-23, 2009.

[3] http://www.santafe.edu/

[4] P. Maes, editor, Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back. MIT Press, 1990.

[5] A. Ilachinski, “Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Combat,” Military Operations Research, 5(3): 29-46, 2000.

[6] R. B. Hencke, “An Agent-Based Approach to Analyzing Information and Coordination in Combat,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, September 1998.

[7] R. F. A. Woodaman, “Agent-Based Simulation of Military Operations Other Than WarSmall Unit Combat,” Master’s Thesis, Naval Postgraduate School, Monterey, CA, September 2000.

[8] T. Bäck, Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.

[9] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Massachusetts, 1989.

[10] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, “Evolutionary Algorithm MOP Approaches”, Evolutionary Algorithms for Solving Multi-Objective Problems, pp. 59—99. Kluwer Academic, 2002.

[11] J. D. Schaffer, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, PhD thesis, Vanderbilt Univ., Nashville, TN, 1984.

[12] V. Tarascio, Pareto's Methodological Approach to Economics, University of North Carolina Press, Chapel Hill, VA, 1968.

[13] K. M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic, Boston, MA, 1998.

[14] I. Das and J. Dennis, “A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems”, Structural Optimization, 14(1):63—69, 1997.

[15] R. Subbu and A. C. Sanderson, “Modeling and Convergence Analysis of Distributed Coevolutionary Algorithms”, IEEE Transactions on Systems, Man, and Cybernetics (Part-B), 34(2): 806-822, 2004.

[16] K. Reinholtz, and K. Patel, “Testing Autonomous Systems for Deep Space Exploration,” IEEE A&E Systems Magazine, September: 22-27, 2008.

[17] K. J. Barltrop, K. H. Friberg, and G. A. Horvath, “Automated Generation and Assessment of Autonomous Systems Test Cases,” IEEE Aerospace Conference, Big Sky, Montana, March 1 - 8, 2008.

[18] A. C. Schultz, J. J. Grefenstette, and K. A. De Jong, “Adaptive Testing of Controllers for Autonomous Vehicles,” Proc. of the IEEE Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, June 1992.

[19] A. C. Schultz, J. J. Grefenstette, and K. A. De Jong, “Test and Evaluation by Genetic Algorithms,” IEEE Expert: Intelligent Systems and Their Applications, 8(5): 9 – 14, 1993.

[20] F. Corno, M. Rebaudengo, and M. S. Reorda, “Experiences in the Use of Evolutionary Techniques for Testing Digital

Circuits," SPIE proceedings series - Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA, July 20-22, 1998.

[21] C. C. Michael, G. McGraw, and M. A. Schatz, "Generating Software Test Data by Evolution," IEEE Transactions on Software Engineering, 27(12): 1085 – 1110, 2001.

[22] M. Harman, K. Lakhotia, and P. McMinn "A Multi–Objective Approach to Search–Based Test Data Generation," Proc. of GECCO'07, London, England, United Kingdom, July 7–11, 2007.

[23] A. Windisch. S. Wappler, and J. Wegener, "Applying Particle Swarm Optimization to Software Testing," Proc. of GECCO'07, London, England, United Kingdom, July 7–11, 2007.

[24] C. D. Nguyen, A. Perini and P. Tonella, "Constraint-based Evolutionary Testing of Autonomous Distributed Systems," IEEE International Conference on Software Testing Verification and Validation Workshop, Lillehammer, Norway, April 9-11, 2008.

[25] R. J. Terrile, and A. Guillaume, "Evolutionary Computation for the Identification of Emergent Behavior in Autonomous Systems," IEEE Aerospace Conference, Big Sky, Montana, March 7 - 14, 2009.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, "IEEE Transactions on Evolutionary Computation," 6(2): 182 – 197, 2002.

[27] http://www.mak.com/products/vrforces.php

[28] R. Subbu and A. C. Sanderson, "Modeling and Convergence Analysis of Distributed Coevolutionary Algorithms," IEEE Transactions on Systems, Man, and Cybernetics (Part-B), 34(2): 1276 – 1283, 2004.

Metrics for Co-evolving Autonomous Systems

Jack Ring
Educe LLC
442 N Sage Ln.
Gilbert, AZ 85234
1+480-488-4615
jring@amug.org

1. ABSTRACT

Autonomous system innovations have overrun the test and evaluation capability to find problems before they become expensive to fix --- or lethal. The autonomy paradigm demands that an equivalent test and evaluation system be conceived, architected and engineered, operated and evolved. This in turn demands an autonomous test and evaluation enterprise, staffed with competent systemists, as the enabling agent. This paper outlines the metrics and key capabilities for realizing such an enterprise. It features a game-theoretic basis, a model-based systems engineering approach and a four part strategic framework. This paper focuses on the unclassified situation in the U.S. Dept. of Defense. However, these ideas will apply to other domains of autonomy in both the public and private sectors.

Categories and Subject Descriptors

C.3 [Computer Systems Organization]: Special Purpose and Application-based Systems – *process control*.

General Terms

Management, Measurement, Design, Economics, Experimentation, Human Factors, Languages, Verification.

Keywords

Ontology, Agility, Reconfigurable, Repurposed, Systemics, Persistent Integrity Assurance

2. INTRODUCTION

Independent, objective test and evaluation, T&E, is a crucial part of all projects, especially for systems of higher complexity and autonomy. DoD-sponsored autonomous systems, both manned and unmanned, are not benefitting from T&E. T&E is equivalent to C4ISR where the 'enemy' is the extent, variety and ambiguity of the Unmanned Autonomous Systems, UAS's, to be tested. DoD plans to produce more than 1,000 new kinds of UAS's costing billions of dollars in the next few years. However, a necessary, sufficient and efficient T&E capability is not being pursued with equivalent alacrity and vigor.

The T&E community must learn to create an autonomous T&E enterprise that can design and generate an agile infrastructure and plug-in modules thereby enabling the rapid instantiation of situation-specific T&E systems whenever and wherever needed. Failure to do this will result either in bypassing independent and objective testing by taking UAS's from development directly to the warfighter, or in over-reliance on modeling and simulation that does not indicate the possible error in its results. Neither of these alternatives are prudent.

This is about T&E of more than an autonomous vehicle. This is about T&E of whole systems composed of multiple, heterogeneous vehicles, networked to C4ISR enabling joint military missions. Such systems are configured in the field in a matter of hours, supported by world-wide production and supply chains and operated by humans, often without benefit of specific training on the system just created. Also, this is about the design, engineering and construction of collateral T&E systems in less than one tenth the cycle time typical of past practices in order to keep pace with the evolutionary acquisition of UAS's. In fact, this challenge reaches clear back into the way practitioners think, formulate, experience and learn about systems[14]. Current standards, guides, and handbooks, regarding the processes and practices of systems engineering, system of systems engineering and family of systems engineering under conceptualize the praxis required to formulate necessary, sufficient and efficient autonomous T&E systems.

Metrics are key. Metrics define the problem. Metrics drive system design/architecting, engineering/construction/ and deployment/evaluation. Hypothesizing solutions sans metrics is malpractice. Accordingly, this paper focuses on whole system metrics. A subsequent paper will focus on whole systems realization.

The balance of this paper presents a description of the metrics that will quantify the problematic situation then a summary of key capabilities that can respond at the rate UAS's evolve and warfighter situations demand.

In addition to military value the advancements described herein are expected to benefit government, industrial and commercial domains ranging from global-scale intelligent transportation to endo-human nanomachines and neurons interfaced to silicon.

3. RELEVANT SEMIOTICS

3.1 Systemics

The concepts that will be used throughout this paper are shown in Figure 1. On the left side, reading from top to bottom, 'e' signifies entity. '─' signifies a relation and 'e─e' signifies a system. Most systems consist of more than two entities and one relationship. This notion is illustrated as three 'e' and three r's. When all are observable this is called an explicit system. Importantly, [18] adds that there can be relationships among relationships. This is called an implicit system. Also, [3] and others point out that if any entities or relationships are not observable or predictable then the notion of soft system applies.

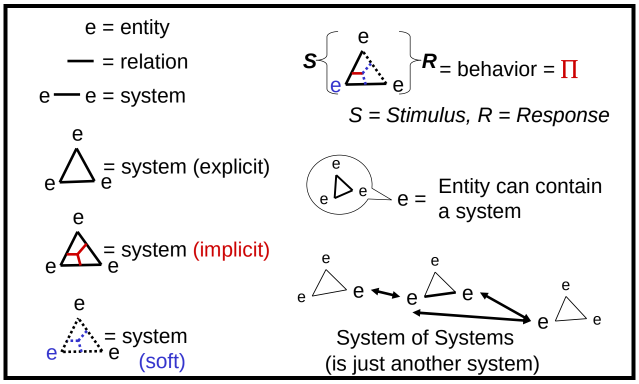


Figure 1. Systemics Concepts

Figure 1, upper right, shows a whole system of entities and relations exhibiting a behavior labeled ∏. This notion is also called Stimulus<>Response by some and Transfer Function by others. Figure 1, center right, shows the notion that any entity can be a system, containing other entities and relationships. Bottom right shows that one or more entities in one system may also participate as entities in another system. Some people label this situation System of Systems. Note, however, that the bottom relationship in the middle system is different when the e's are participating in the larger system than if the middle system is not interoperating with the other systems. Relevant, here, is the [7] notion of holon, a system that simultaneously can be a component of another system.

All systems are man made. Some by descriptive modeling of "natural' systems, e.g., the human body, and others by prescriptive modeling of systems intended to successfully intervene in problematic situations. The essence of a systemics praxis adapted from [15]. Is shown in Figure 2. Key concepts are Context, Content Structure. Prudent praxis involves descriptive modeling of the Context, designated as [1] in Figure 2, especially the underlying problem system [2], then nominating System Content [3], the capabilities that are intended to moderate the problem system. The several hypotheses are then systemized by nominating structure, the pattern of interrelations among them [5] that is expected to produce the intended responses when encountering the various stimuli. The arrangement of content and relationships comprise the system architecture. In modern systems engineering parlance this is called the Effects and Capabilities approach to systems.

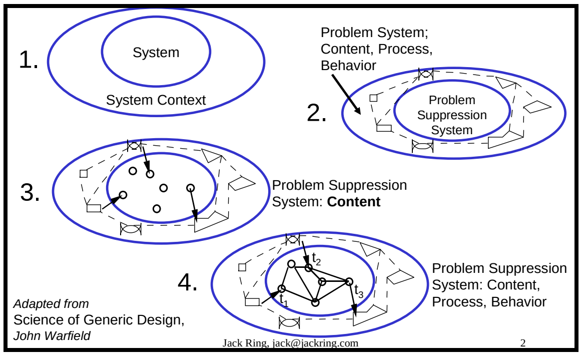


Figure 2. Systems Praxis

When this prescriptive model is implemented the behavior of the system becomes undeniably evident and often surprising. This spawned the notion of POSIWID, the purpose of a system is what it does regardless of designer intent [2] which leads to the notion that T&E must measure not only system viability in specified usages (contexts) but also system limits of stability and integrity.

Figure 3 shows the notion of a system (in the center ellipse) exhibiting its behavior labeled ∏. A system exists in a Situation Space, is influenced by the Problem Space and influences the Value Space. System behavior results from several influences.

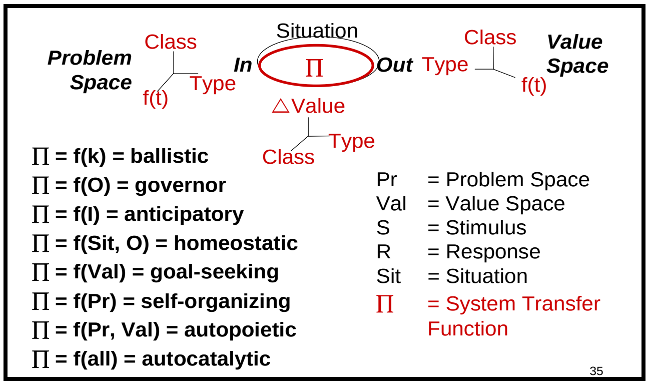


Figure 3. Categories of System Behavior

The Problem Space and Value Space artifacts can be categorized by Class, Type and temporal existence. The Situation Space attributes are the Δ Value that the ∏ must contribute as well as Class and Type. Labels for the various kinds of system behaviors are shown at the lower left of Figure 3. These indicate the spectrum of system behaviors that will be encountered in T&E of autonomous systems, manned or unmanned.

System behavior ordains system worth. In all but trivial cases and especially in autonomous situations system behavior manifests in multiple modes, each with perhaps multiple states within each mode. We use 'coverage' to signify the degree to which T&E reveals all of these.

3.2 Field of Discourse

Figure 4 presents a concept map of our Field of Discourse regarding unmanned autonomous systems test and evaluation.

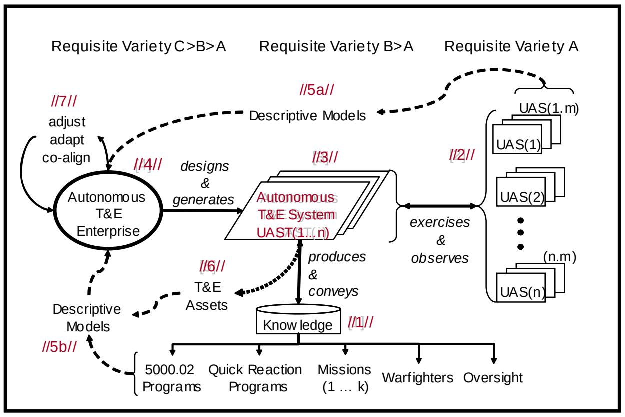


Figure 4. UAST Field of Discourse.

Knowledge //1// serves the needs of various stakeholders regarding respective UAS's //2//. Members of a set of autonomous T&E Systems, UAST(1…n) //3// produce and convey relevant knowledge as each exercises and observes selected UAS's. An Autonomous T&E Enterprise, ATEE //4// designs and generates instances, UAST(i), of the generic Autonomous T&E System.

Requisite Variety, RV, [1] ramifications are noted across the top of Figure 4. A UAS must exhibit a certain degree of RV to accomplish its mission. This, in turn requires a higher degree of RV by the UAST(i) for testing the UAS. This, in turn, requires a yet higher degree of RV by the ATEE, that produces UAST's.

Three more concepts are shown in Figure 4. One concerns descriptive models of UAS's //5a// and Stakeholders //5b//. A second concerns descriptive models of T&E Assets //7// that are re-used to formulate UAST's. Descriptive models are executable formal ontologies that reveal expected system behaviors. The third recognizes that the Autonomous T&E Enterprise continuously adjusts its own gradients, adapts its pattern of relationships and co-aligns its content relative to its context [8].

3.3 Whole System Viewpoint

Figure 5 depicts a whole system anatomy signifying the several aspects of a whole system that must be measured across a range of contextual situations. The cloud on the left represents the problematic situation, containing the problem system from which stimuli emanate. The primary mission system is called the Problem Suppression System. It is enabled by the Operational Availability System (the logistics, maintenance, etc., that keep the Problem Suppression System running). The Operator Preparation System prepares the war fighters. The Production System produces multiple copies. The Test System(s) report on the readiness of all the others and of 'itself."

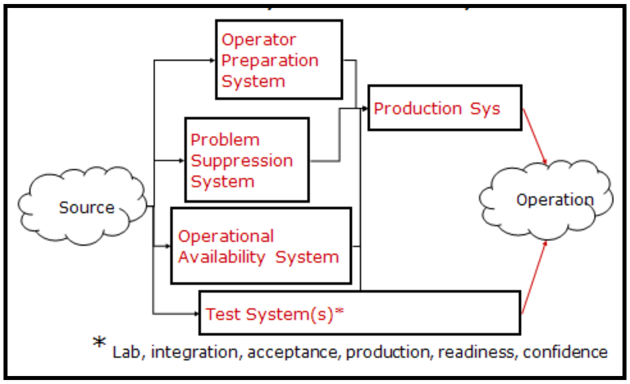

Operator Preparation System
Production Sys
Problem Suppression System
Source
Operation
Operational Availability System
Test System(s)*
* Lab, integration, acceptance, production, readiness, confidence

Figure 5. Whole System Anatomy

It is important to note and remember our use of UAS signifies the whole system, including the personnel, as do UAST and ATEE, respectively.

4. METRICS

4.1 General System Metrics

Useful metrics for a general system are Quality, Parsimony and Beauty. Quality in the [4] sense of Conformance to Requirements, a binary Yes or No, not the fuzzy 'high' or 'low' quality. Parsimony in the sense that no other system exhibits required quality at less cost of ownership. Beauty in the machine sense as articulated by [6] and in the human sense as articulated by R. Buckminster Fuller, "When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong."

4.2 Knowledge Metrics

Figure 6, adapted from [10], clarifies the knowledge metric. Kinds of knowledge ranging from concepts to theory are listed on the left side of Figure 6.

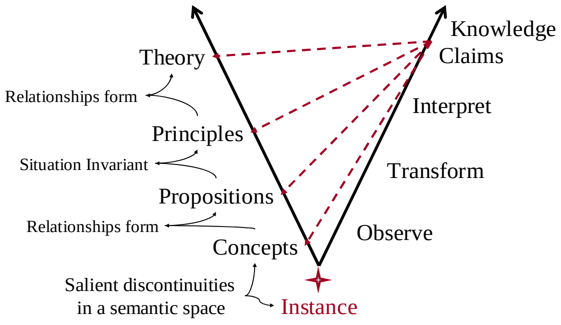


Figure 6. Sources and Kinds of Knowledge Claims

A theory consists of a set of interrelated Principles which, in turn are a set of propositions that transcend specific situations. Propositions are interrelated Concepts. Concepts are the fundamental building blocks and are simply meaningful discontinuities in a semantic space. A T&E activity fits on the right side of Figure 6. T&E Acknowledges UAS events, Observes the characteristics, Interprets the findings and Produces knowledge claims (as indicated by the dashed lines). Knowledge claims can pertain to any of the four constructs on the left side of the Vee.

Quality of knowledge claims is measured by adequacy, accuracy and timeliness of the claims. Adequacy connotes the spectrum of knowledge that occurs across the several kinds of interested parties. Accuracy (of knowledge claims) connotes not only the quality of observations but also the assessment of the likelihood of error in evaluations. Timeliness connotes whether the latency from time of observation to time of conveyance of the knowledge claim is/was consistent with stakeholder intended usage.

Parsimony concerns not only the cost of producing the claims but also how well new claims are conveyed. In light of Ausabel's theory of meaningful learning [10] this entails relating new knowledge claims to stakeholders' existing knowledge. Of course this implies that the Evaluation side of T&E be familiar with the 'audience' of stakeholders and their knowledge states even though neither the UAST nor Autonomous T&E Enterprise gets to select the stakeholders.

4.3 UAS Metrics

The U.S. Dept. of Defense planning horizon anticipates more than 1,000 kinds of UA vehicles spanning Space, Air, Ground, Marine and Undersea. Current examples range from 40 ton ground-based monsters to 2.3 gram airborne platforms that carry video cameras. In addition to these in physical space we can expect many other kinds of UAS's in cyberspace.

Stakeholders want to know about UAS Safety, Suitability, Effectiveness and Survivability. These describe UAS characteristics and properties (see Appendix 1) not only in a specified, nominal scenario but also across a various operating modes such as degraded, diagnostic, training, maintenance and re-purposing. Further, UAS dynamic and integrity limits must be determined by actual test or by estimation techniques.

UAS Testability is a key metric. Inevitably, stakeholders want to know 'why' a UAS system did what it did (and want to predict what it may or will do in a future scenario). This is a challenge for the Evaluation side of T&E. It gives rise to a metric regarding the degree to which a UAS Whole System as shown in Figure 5 self-identifies and reports its current configuration self assessment of its readiness. For example, no testing should begin if the UAS is rife with bugs. If the UAS does not have these capabilities then the UAST must have the capability to inspect and assess the UAS.

4.4 UAST Metrics

The Autonomous T&E System, shown at [3] in Figure 4, covers all members of the set, UAS, as well as relevant members of the sets, Missions, and Stakeholder Interests. The Autonomous T&E System is a set of UAST(I), each member being sufficient yet parsimonious with respect to specific UAS-Mission-Stakeholder triples. A framework for UAST's is described in (8).

The knowledge needs of the several diverse stakeholders suggest that tens of test episodes will be needed for each kind of system. Parsimony factors such as budgets constrain the T&E community to far less than thousands of unique UAST installations. However one UAST would be far too large and complex to build, let alone schedule and operate. The answer lies somewhere in between. We can be reasonably confident of the concept of a set of UAST(i), each using a mix of multi-use and situation-specific components specifically configured to stimulate and observe a part of or a whole UA system then produce and convey knowledge claims. However, range of 'i' remains to be discovered because it depends on the maximization of the whole system shown in Figure 4.

Considering the degree of autonomy that can be exhibited by a UAS or a mission-oriented group of UAS's a UAST(i) must exhibit even greater autonomy (essentially Ashby's Requisite Variety, RV). This metric has been obvious in Range Safety systems at large test ranges. Now, a UAST(i) must have the requisite variety to morph its operations and even its configuration during UAS operation not only for Range Safety and fratricide avoidance but also for contriving stimulations and observations as well as producing and conveying unforeseen knowledge claims.

In essence, the UAST(i) must be able to prevail as the Angel in a formal game with UAS(s) as Demons while the UAS(s) are prevailing as Angels in a formal game with their Mission context as Demons. [Pizzarello, A., OntoPilot LLC, private communication].

Because of its autonomy each UAST(i) must include the capability of persistent viability assurance. In Joint testing episodes a UAST(i) may interoperate with other operational and test systems. Persistent viability of the associations (as Demons) must be measured.

Meanwhile parsimony demands that UAST's must be able to determine readiness for test of both UAS and UAST systems. Relevant metrics are incidence of test aborts (ideally zero).

Other metrics reveal cost of UAST generation (including proactive, facilitated reuse of assets), operation and recovery. Underneath, asset turnover is a key metric along with metrics for characterizing each asset as reuseful, reusable and reused.

4.5 ATEE Metrics

The Autonomous T&E Enterprise, ATEE, designs and generates the Autonomous T&E System as UAST(1…n) instances in response to UAS–mission-stakeholder triples. The extent, variety and ambiguity presented by the UAS set and the desired UAST set demands that the ATEE operate as an Intelligent Enterprise. Figure 7 indicates the key context, content, structure and behavior of an intelligent enterprise [12].

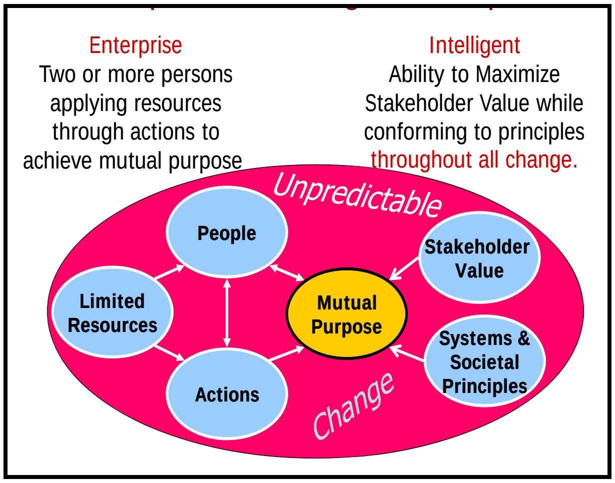


Figure 7. Concept of Intelligent Enterprise

Two or more persons applying resources through actions to achieve mutual purpose regarding both stakeholder value and systems and societal principles, all in a context of unpredictable change.

Key capabilities are [8] a Goal, Triggers, Energy, Competence, Situation Assessment and Gap Closure. Intelligent system quality metrics include accuracy in achieving mutual purpose, response time to external and internal changes, and dynamic limits with respect to rate of change. Parsimony metrics are cost to achieve goal and the cost of disergy, having more responsiveness than is needed. Beauty metrics are [5] Market Standing, Productivity, Innovation and Liquidity.

Unfortunately, the Drucker metrics are lagging indicators. Leading indicators involve the orchestration of change because an enterprise evolves in many steps across many aspects. Because a system can be viewed as content and structure (relationships) and because its behavior depends on the interrelationship gradients, an intelligent enterprise has three modalities of change, notably, adjusting gradients, adapting the pattern of interrelationships and co-evolving its content. All three must be measured so that the enterprise changes gracefully rather than chaotically. Gracefully recognizes well known thermodynamics constraints, notably, conservation of mass, momentum and energy. In the intelligent enterprise graceful change also acknowledges other constraints stemming from other factors comprising an enterprise, notably, Informatics (data, information and knowledge), Teleonomics (skills mix, rate of learning, and rate of invention), Human social dynamics (trust, enthusiasm, co-evolution), Economic (investment, ROI, liquidity), and Ecology (tbd).

Graceful evolution of the ATEE is highly important. Once change appears incoherent to the incumbents a variety of depressing and destructive behaviors can arise. Enterprises exhibit an etiology, exemplified in Figure 8. The quality of information available to the participants is shown on the left. Important metrics regarding personnel attitudes are shown on the right, aligned with each kind of information.

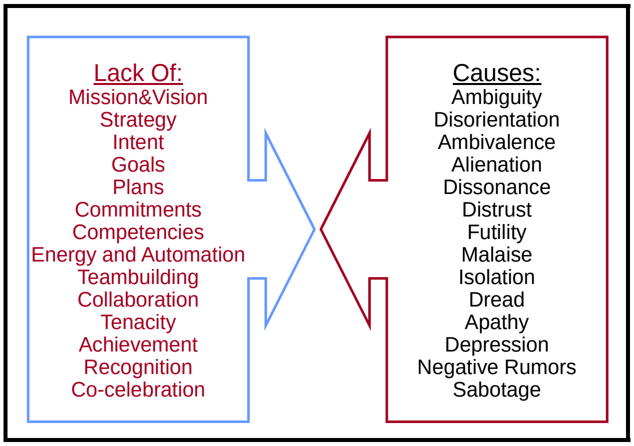


Figure 8. Enterprise Etiology
The lack of each item on the left causes the personnel situation on the right.

As in the UAS and UAST cases, ATEE Testability is a key metric. Inevitably, stakeholders want to know ‘why’ the ATEE did what it did (and want to predict what it may or will do in a future scenario). An autonomous ATEE must contain a model of ‘itself.’ It gives rise to a metric regarding the fidelity of the model and challenges the Evaluation side of T&E to quantify it.

5. DISCUSSION

The myriad metrics identified in the foregoing (undoubtedly readers can nominate many more) indicates the mind-numbing extent, variety and ambiguity that the UAS T&E community must master. As [17] cautions, such situations of cognitive overload lead to underconceptualization of solutions. Fortunately, ways of coping with these problematic situations have been demonstrated recently though perhaps not within the Department of Defense, DoD. The page limit in this paper does not allow more than pointers to the key ideas that merit further consideration. Four are summarized in this section. However, brevity does not imply arbitrary choice. All four must be pursued in order to formulate a sufficient cascade of requisite variety.

a) Autonomous T&E Enterprise: Commit to creating an Autonomous (intelligent) T&E Enterprise staffed with qualified DoD personnel. The DoD T&E capability must be distributed geographically and across Services but unified in strategy, objectivity and systemics. The new enterprise should be acknowledged as an intelligent system and conduct their practice reflectively (14). This may entail formulation of a Concept of Operations (usage), an intervention strategy, agile system design and evolutionary acquisition of personnel and assets. The Interactive Management process [16] is appropriate. Architecture-frameworks are not.

b) Enhance Systemics Praxis: Practitioner productivity and innovation must be increased approximately ten fold over current practices [13]. Evolve the Effects-Capabilities approach into model-based systems engineering [19] of whole systems.

   Evolve the DoD guidelines for systems engineering of system of systems. The current version adds seven new processes to sixteen processes from traditional systems engineering as described in ISO #15288. All ignore metrics. Further, improving traditional systems praxis will not suffice for the autonomy paradigm shift.

   Revise the JTEM approach to separate the systems engineering of the operational capability from the systems engineering of the T&E capability.

c) UAST Generator: Develop an ontology-based configuration generation and behavior estimation tool encompassing thermodynamics, informatics, biomatics, teleonomics, human social dynamics, economics and ecologics.

   Generate prescriptive models and UAST(i) build scripts that enable engineering, construction and readiness verification within the UAS development cycle times and Field Command cycle times.

   Leverage new technologies to assess UAS and UAST correctness in seconds instead of hours of test and retest [11].

d) Proactive Interchange of Models: Demand that UAS development projects provide an executable model of the mission profile for which the UAS was designed. Likewise from field commanders who use JTEM and net-centric configurators.

6. CONCLUSIONS

Current practices for exercising and observing UAS’s in various stages of development are inadequate, partly because of the complexity (actually the extent, variety and ambiguity) of the situation but also because most participants under conceptualize the systemics involved [17].

Adequate, accurate and timely knowledge about UAS’s will not be garnered by simply understanding and categorizing UAS concepts. UAS’s must be exercised and observed by simulation, emulation and actual operation. In Live, Virtual and Constructive approaches the fidelity of contrived context is a key metric. Expansion of evaluation techniques and tools for *in situ* testing should be pursued.

A necessary and sufficient response to the problematic situation requires that the whole field of discourse be treated as one system, that the cascade of requisite variety be necessary, sufficient and efficient and that the whole system be treated as an implicit, soft system.

Metrics define the problem. Hypothesizing solutions sans metrics that describe the intended effect on the problem is malpractice.

Presuming the T&E community converges on a set of metrics, then evolutionary acquisition of the ATEE will be prudent and urgent. A concept of operations (usage) of the whole system depicted in Figure 5 must be prepared and vetted along with an Intervention Strategy, and a framework of ATEE capabilities.

Proven practices exist for accomplishing these objectives. They should be identified, adopted and systematized.

Because an autonomous system includes a model of ‘itself’ two kinds of metrics are key. One set of metrics is concerned with the state of the system while the other set of metrics is concerned with the state of the model, especially the viability of a proposed scenario of change.

Because any system ‘as is’ rarely matches the system presumed then establishing the identity of a system (the capability of a

UAST to ‘know’ the UAS and of a ATEE to ‘know’ a UAST) is fundamental [11].

7. ACKNOWLEDGMENTS

Fil Macias, Ph.D., Executive for the UAST Focus Group, TRMC, DoD, and members Thomas Tenorio, Ph.D., Frank Lopez, Ph.D., and Philipp Djang, Ph.D. stimulated ideas on this subject. Thanks to PreMIS track organizers, Nikita Visnevski, Ph.D. and Mauricio Castillo-Effen, Ph.D., both of GE.

8. REFERENCES

[1] Ashby, W.R., 1956 Introduction to Cybernetics, Chapman and Hall.

[2] Beer, S. 2002 "What is cybernetics?" Kybernetes (MCB UP Ltd) 31 (2): 209–219.

[3] Checkland, P, and Scholes, J. 1990 Soft Systems Methodology in Action. (Wiley) ISBN 0-471-92768-6

[4] Crosby, P., 1995 Quality Without Tears: The Art of Hassle-Free Management, McGraw Hill

[5] Drucker, P. 1954 The Practice of Management, Harper & Brothers

[6] Gelertner, D., 1998 Machine Beauty, Basic Books

[7] Koestler, A. 1967. The Ghost in the Machine Penguin Group (1990 reprint edition)..

[8] Livingston III, W. 1990 Friends In High Places, F.E.S. Limited Publishing

[9] Macias, F. 2008 *Test and Evaluation of Unmanned and Autonomous Systems*, ITEA Journal, Vol 29 No 4, pg 392.

[10] Novak, J., 1998 Learning, Creating, & Using knowledge, Lawrence Erlbaum Assoc.

[11] Ring, J. and Pizzarello, A., System of Systems Viability Assessment Capability, Presentation at 3rd Annual System of Systems Conference, NIST, Gaithersburg, MD, December, 2008, available from author, jring@amug.org

[12] Ring, J. ed. 2007 About Intelligent Enterprises. A Collection of Knowledge Claims. INCOSE 2007

[13] Ring, J. 2009 Discovering a Strategy for Evolving to Whole System Modeling, IEEE Systems Conference.

[14] Schon, D.,1987 Educating the Reflective Practitioner, Jossey-Bass.

[15] Warfield, J. 1990 Science of Generic Design, Intersystem Publications.

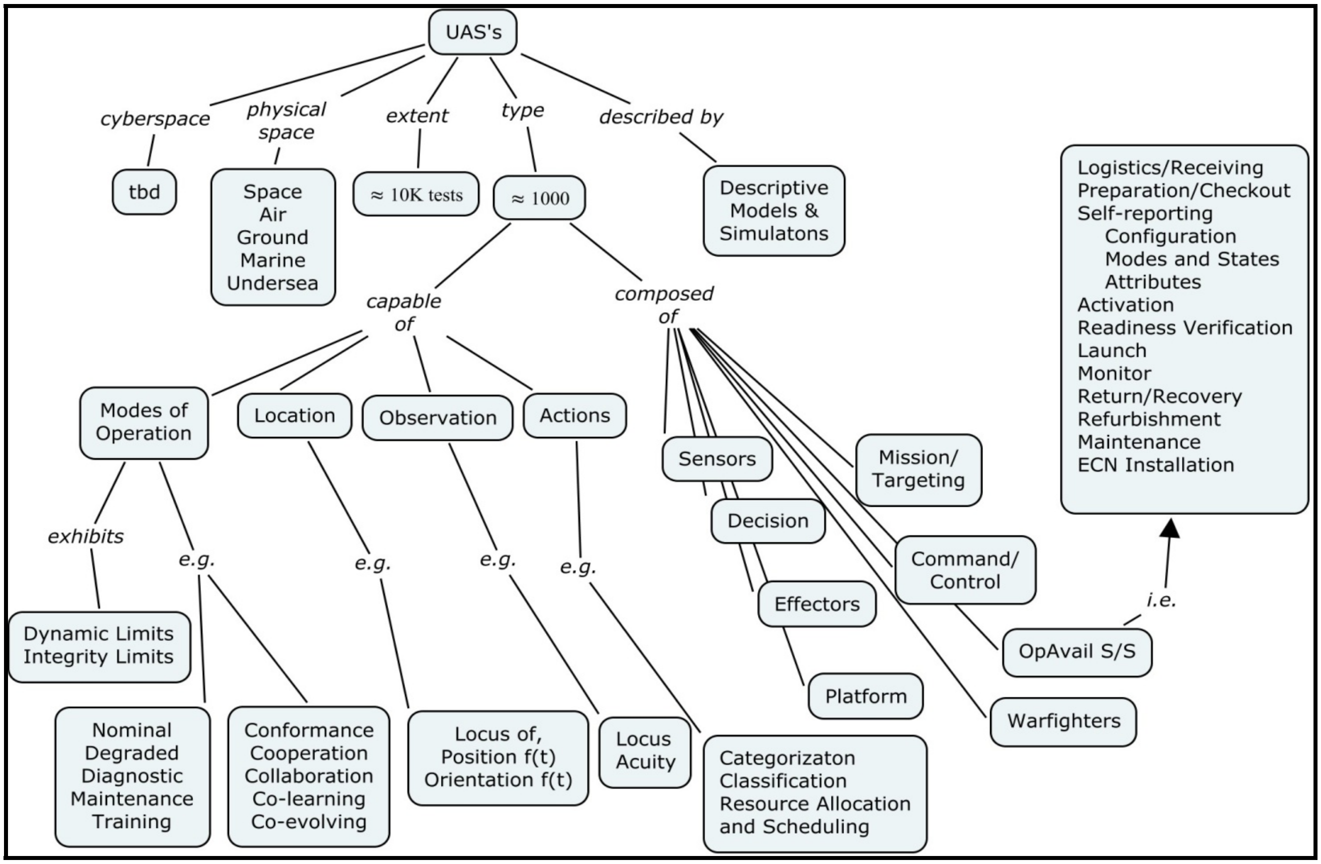
[16] Warfield, J. 1994 Handbook for Interactive Management, Iowa State University Press.

[17] Warfield, J. 2002 Understanding Complexity: Thought and Behavior, Ajar Publishing.

[18] Weinberg, G. 2001 An introduction to general systems thinking, Dorset House Publishing Co., Inc.

[19] Wymore, A. 1993 Model-based Systems Engineering, CRC Press.

Appendix 1. Example UAS Characteristics and Properties



The Role of Competitions in Advancing Intelligent Systems: A Practitioner's Perspective

Elena Messina
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230
+1 (301) 975-3510
elena.messina@nist.gov

Raj Madhavan
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230
+1 (301) 975-2865
raj.madhavan@nist.gov

Stephen Balakirsky
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230
+1 (301) 975-4791
stephen.balakirsky@nist.gov

ABSTRACT

In recent years, the number of competitions in the robotic domain has increased tremendously. This growth is spurred by the appealing nature of robots, the flexibility that they afford in competition themes due to the practically unlimited applications, and by the recognition that competitions can yield advances in a technological domain that is immature. The National Institute of Standards and Technology has supported a variety of competitions to help stimulate innovation in certain critical technologies and capabilities needed in robotics. In this paper we present some general views of competitions and discuss the experiences NIST has had with robotics competitions as catalysts for advancing the state of intelligent systems. This paper is a lead-in to others in a special session organized at the 2010 Performance Metrics for Intelligent Systems (PerMIS) workshop that describe in more detail how competitions are used to advance intelligent systems.

Categories and Subject Descriptors

F.2.3 [**Theory of Computation**]: Analysis of Algorithms and Problem Features – *Tradeoffs among complexity measures.*

General Terms

Measurement, Performance Evaluation, Performance Metrics, Intelligent Systems, Robotics

Keywords

Robotics, Competitions

1. INTRODUCTION

Robot competitions are becoming increasingly popular, serving a number of goals that range from primary education to stimulating technological advancements for real-world applications. Competitions can be categorized in many ways. In this paper, we view them through a prism of three dimensions: motivation, objectives, and evaluation techniques. Note that these usually have some form of interdependence.

The spectrum of motivations for robot competitions ranges widely. On one end, robot competitions are pure entertainment. There have been several popular television shows featuring robot combat, for instance, such as Robot Wars[1] and BattleBots [1] [2]. These types of competitions exercise the creativity of the contestants, but are not designed to specifically advance the state of the art. Education is another popular motivation for competition. The For Inspiration and Recognitions of Science and Technology (FIRST) competition was specifically created to inspire young peoples' interest and participation in science and technology [3]. A third major motivator for competitions is to stimulate progress in the technology itself. One-time "grand challenges" are an example of this. Some competitions blend education and technological advancement. One instance of this is the RoboCup array of competitions. In this paper, we briefly present an overview of the spectrum of objectives and methodologies employed in robotics competitions. Certainly, all forms of competitions involve performance measurements, making them useful to examine in the context of performance metrics for intelligent systems.

2. THE BEGINNINGS: MICRO-MOUSE

It has been over three decades since the first mobile robot competition made its debut. The Micro-mouse maze contest, announced in 1977, was first conducted in 1979 and is considered the first such competition [4]. The initial task was simple: a robot mouse was to drive from start to goal through a maze in the least time. Over the years, the rules evolved to create greater challenges to the intelligence of the "mice." In the first contest, the robots were simple wall-huggers. An early rule change was to have the mouse start in a corner of the maze and end up in the center. This change forced more intelligence in the path planning. Another later rule change required the mouse to explore the entire maze and then compute the shortest path. Even this earliest competition illustrates the basic premise of this paper: that carefully crafted competitions (and their rules, which should evolve) can steer research advancements. According to Braünl [5], by 1999 the electronics, sensors, and software problems of the micromouse were solved, with only mechanical improvements still possible.

[1] Certain commercial equipment, instruments, or materials are identified in this document. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology nor does it imply that the products identified are necessarily the best available for the purpose.

3. SPECTRUM OF COMPETITIONS

Since the micro-mouse beginnings, robotics competitions have flourished, for a variety of purposes and reasons. Robots are inherently appealing to youngsters. Therefore robot-centered competitions are useful for attracting students to science, technology, engineering, and math (STEM). Competitions can expose students to many aspects of STEM and encourage them to pursue studies in these disciplines. Notable examples of STEM-oriented competitions are FIRST and BotBall [6].

At the college level, a number of competitions are designed to bolster engineering education by providing a systems design and integration challenge. For example, the Association for Unmanned Vehicle Systems International (AUVSI) has a wide range of robotics competitions spanning ground, aerial, and aquatic domains [7]. The challenges posed are representative of missions that robots currently cannot complete in the military or commercial world, but the emphasis is on education. A recent AUVSI International Aerial Robotics Competition (IARC) had fairly detailed scoring, allotting points for effectiveness measures, such as avoiding all obstacles without collision, and for specific mission tasks such as retrieving a specific object, as well as subjective measures, such as elegance of design and safety of design to bystanders. There are even points allotted for the quality of a journal paper submitted by the team and for best tee shirt design. The mission design for each competition builds on the prior ones, increasing in difficulty with each year.

High risk, high payoff competitions have been staged to advance the state of the art in targeted applications. This “grand challenge” model is used to introduce a community to a compelling and major technological goal that can only be attained by concentrated and often collaborative efforts. A recent example is that of the United States’ (U.S.) Defense Advanced Research Projects Agency (DARPA) competitions for driverless vehicles. These competitions have offered prizes of a million dollars and above to teams that successfully complete a course autonomously. The impetus for this competition is the U.S. military’s stated goal of having one third of ground military vehicles autonomous by 2015. The first challenge was off-road, and none of the robots completed more than 11.78 km out of the 20 km length course in the first year. By the second year, all but one of the contestants went beyond 11.78 km and five teams completed the entire course [8]. For the third year, the competition turned its focus to urban driving environments and required the vehicles to follow traffic laws. The competition featured multiple vehicles on the course simultaneously. Six teams completed this challenge [9]. Another dimension of competitions is the set of objectives for victory. Yanco proposed a taxonomy for determining competition outcomes that includes ranked competition with subjective scoring, ranked competition with objective scoring, and non-ranked competition with technical awards [10]. In this paper, we augment Yanco’s perspective. Some robotic competitions may require the contestants to complete a successful task. The Association for the Advancement of Artificial Intelligence (AAAI) has held several competitions at its annual conference, many of which have been credited with fostering advancements in robotics. For the “Hors d’ Oeuvres Anyone?” competition, robots served food to attendees at the conference’s banquet. The scoring for this competition included an audience vote component, along with the successful completion of the task of serving food (including restocking the serving tray). The Hors d’ Oeuvres event drove research in manipulation, navigation through dynamic worlds, and human-robot interaction [11].

Some competitions are based on team contests. Various leagues within the RoboCup Soccer organization pit two teams against each other on a range of soccer fields. The RoboCup initiative’s goal is to “foster artificial intelligence and robotics research by providing a standard problem where a wide range of technologies can be examined and integrated” [12]. The soccer competitions were begun in 1997, with several leagues designed to challenge different aspects of the overall robotics problem. For example, the small-size (below 18 cm diameter) robot league focuses on the issues of multi-agent cooperation with a hybrid centralized/distributed system, whereas the humanoid league encourages mechanical and electronic advances in physical bipedal robots, as well as in the planning and perception software. The abilities of the robots have steadily – even dramatically – improved over the years. Robots can detect the ball, goal, and opponents as well as teammates much more quickly than in the early years and can apply strategy and adaptive techniques. The sensing and planning have evolved tremendously [13]. On the hardware front, there has been significant progress for humanoid robots. They have increased bipedal stability and can move with greater agility each year. The progress is evidence that an ongoing, well-defined set of challenges can inspire innovation.

Other competitions use performance-based models, pitting the robots against a baseline measure. Such is the case with RoboCup Rescue. In 2001, the RoboCup organizers expanded their competitions to include disaster rescue [14]. Viewed as an important challenge in robotics, wherein large numbers of heterogeneous agents collaborate within hostile environments, there are multiple competitions and leagues in this application area [15]. The multi-faceted goals of RoboCup Rescue are “to promote research and development in this significant domain by involving multi-agent team work coordination, physical robotic agents for search and rescue, information infrastructures, personal digital assistants, standard simulator and decision support systems, evaluation benchmarks for rescue strategies and robotic systems that are all integrated into a comprehensive system in the future” [16]. The competitions in the physical robot league pit a robot or team of robots against a disaster environment, called the arena. The robot is to explore the space, map it, and identify victims within a fixed time period. There are many mobility challenges, and areas where fully autonomous operation is required (teleoperation is allowed in many parts of the arena). Robotic hardware designs, as well as software algorithms and sensors have shown tremendous progress in the past decade. Innovations introduced by teams that prove successful are quickly replicated by others, disseminating good designs and accelerating progress. Robots can tackle terrains that were deemed impossible a few years ago. They produce maps of better quality with each passing year. The competition’s rules and scoring metrics are revised each year in order to ensure that the challenges grow increasingly complex and more reflective of reality. For instance, there are areas of the arenas where robots must operate exclusively autonomously and new manipulation tasks (e.g., opening doors) are being introduced.

A virtual competition for rescue offers larger, more complex environments and stresses collaborative planning of teams of robots [17]. Teams are required to address elemental tasks that include the autonomous distribution of up to eight robots to form communication repeater networks, autonomous multi-vehicle mapping, and multi-vehicle tele-operation. These skills are then brought together in a simulated full rescue scenario. As in the physical rescue league, the tasks and rules are modified as teams become more capable. This competition also provides an open source coding environment and the requirement that a team's source code becomes open source at the conclusion of the event. This assures that new teams are able to quickly become competitive and that good ideas propagate throughout the community.

RoboCup Rescue has evolved over the years to directly tie the competition to performance standards being developed for response robots. In a NIST-led project funded by the Department of Homeland Security, individual test methods are being developed to measure how well a robot meets certain requirements, which have been defined by end users [18]. Examples of requirements entail mobility over terrains of varying difficulty and the ability to aim or direct cameras and other sensors in a purposeful way to identify victims or relevant items in the environment. Individual test method elements are incorporated within the physical competition arenas. Thus, the research community is presented with real-world challenges against which they can pit their ingenuity and thereby advance the state of the art in robotics.

The success in stimulating innovation in the rescue robotics community via the RoboCup competitions led NIST to establish competitions in other domains. A virtual manufacturing automation competition (VMAC) strives to promote advancements in robotic algorithms, especially in sensing and planning, for factory operations [19]. Recent advances in microelectromechanical systems have enabled the development of mobile microrobots that can autonomously navigate and manipulate. This technology is expected to be critical to numerous applications, including sensor networks, medical diagnosis and treatment, and micro-assembly. Since there are many challenges, such as in locomotion, NIST has organized performance-based competitions for mobile microrobots to help coalesce the research community. Both the VMAC and the microrobotics competitions have been adopted by the Institute of Electrical and Electronics Engineers' annual International Conference on Robotics and Automation. The RoboCup organization hosted microrobotics demonstrations for the first few years.

Recent advances in the design and fabrication of microelectromechanical systems (MEMS) have enabled the development of mobile microrobots that can autonomously navigate and manipulate in controlled environments. It is expected that this technology will be critical in applications as varied as intelligent sensor networks, in vivo medical diagnosis and treatment, and adaptive microelectronics.

However, many challenges remain, particularly with respect to locomotion, power storage, embedded intelligence, and motion measurement. As a result, NIST has organized performance-based competitions for mobile microrobots that are designed to: 1) motivate researchers to accelerate microrobot development, 2) reveal the most pressing technical challenges, and 3) evaluate the most successful methods for locomotion and manipulation at the microscale (e.g., actuation techniques for crawling).

4. CONCLUSIONS

We have discussed a sampling of robotics competitions and the various objectives possible. This is not meant to be an exhaustive list, as there are too many competitions (and the number is growing annually). For example, [20] gives several examples of how mobile robot competitions can foster advances on many fronts. Clearly, robotics competitions are useful mechanisms to serve many purposes, ranging from entertainment to education to stimulating innovations. According to Yanco, "Competitions often influence the direction of research in robotics, which can be used to great advantage" [10]. Incorporating ways of measuring performance in particular tasks or missions has proven to be a useful means of helping the research community better understand the problems to be solved. Having annual competitions with evolving challenges as technologies mature is an effective way of motivating the creativity of the international robotics community towards useful, real-world solutions and advancements in the technologies for robots.

5. REFERENCES

[1] http://www.marcthorpe.com/robot.html

[2] http://www.battlebots.com/BattleBots.com/Home/Home.html

[3] http://www.usfirst.org/

[4] http://micromouse.cannock.ac.uk/history.htm

[5] Braünl. T., Research Relevance of Mobile Robot Competitions, in IEEE Robotics and Automation Magazine, Vol. 6, no. 4, Dec 1999, pp 32-76.Balch T. and Yanco, H., Ten Years of the AAAI Mobile Robot Competition and Exhibition, AI Magazine, Vol. 23, No. 1 (Spring 2002).

[6] Miller, D., Using robotics to teach computer programming and AI concepts to engineering students. In Accessible Hands-on Artificial Intelligence and Robotics Education, AAAI Spring Symposium, 2005.

[7] http://www.auvsi.org/AUVSI/AUVSI/Events/AUVSIStudentCompetitions/

[8] Seetharaman, G. Lakhotia, A. Blasch, E.P., Unmanned Vehicles Come of Age: The DARPA Grand Challenge, Computer, Volume: 39, Issue: 12, December, 2006.

[9] Buehler, M., Iagnemma, K., and Singh, S., Editorial, Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part I, Volume 25 Issue 8.

[10] Yanco. H., "Designing Metrics for Comparing the Performance of Robotic Systems in Robot Competitions, " In Proceedings of the Workshop on Measuring Performance and Intelligence of Intelligent Systems (PerMIS), IEEE Conference on Control Applications, Mexico City, Mexico, September 2001.

[11] Balch T. and Yanco, H., "Ten Years of the AAAI Mobile Robot Competition and Exhibition," AI Magazine, Vol. 23, No. 1 (Spring 2002)

[12] Kitano, H., Kuniyoshi, Y., Noda, I., Asada, M., Matsubara, H., and Osawa, E., RoboCup: A challenge problem for AI. AI Magazine, 18(1):73–85, Spring 1997.

[13] E. Pagello, E. Menegatti, A. Bredenfel, P. Costa, T. Christaller, A. Jacoff, D. Polani, M. Riedmiller, A. Saffiotti, E, Sklar, and T. Tomoichi, "RoboCup-2003 New Scientific and Technical Advances", AI Magazine, Volume 25, No. 2, 2004.

[14] Jacoff, A., Messina, E., Evans, J., "A Standard Test Course for Urban Search and Rescue Robots," Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, NIST SP 970, August 2000.

[15] Kitano, H. and Tadokoro, S., "RoboCup Rescue: A Grand Challenge for Multiagent and Intelligent Systems," AI Magazine, Volume 22, No. 1, 2001.

[16] http://www.robocuprescue.org/

[17] S. Balakirsky, S. Carpin, and A. Visser , "Evaluating The RoboCup 2009 Virtual Robot Rescue Competition," Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, Gaithersburg, MD, September 2009.

[18] Messina, E., "Performance Standards for Urban Search & Rescue Robots: Enabling Deployment of New Tools for Responders," Defense Standardization Program Office Journal, July/December 2007, pp. 43-48.

[19] Balakirsky, S. and Madhavan, R., "Advancing Manufacturing Research Through Competitions," Proceedings of the SPIE Defense Security and Sensing, Orlando, FL, April 13-17, 2009.

[20] L. Almeida, J. Azevedo, C. Cardeira, P. Costa, P. Fonseca, P. Lima, F. Ribeiro, V. Santos, "Mobile Robot Competitions: Fostering Advances in Research, Development, and Education in Robotics," in Proceedings of CONTROLO'2000, the 4th Portuguese Conference on Automatic Control, Guimarães, 2000

Evaluating The RoboCup 2009 Virtual Robot Rescue Competition

Stephen Balakirsky
NIST
100 Bureau Drive
Gaithersburg, MD, USA
+1 (301) 975-4791
stephen@nist.gov

Stefano Carpin
University of California, Merced
5200 N Lake Rd
Merced, CA, USA
+1 (209) 228-4152
scarpin@ucmerced.edu

Arnoud Visser
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam, NL
+31 (20) 525-7532
A.Visser@uva.nll

ABSTRACT

The 2009 RoboCup Competitions took place in Graz, Austria in July of 2009. The Virtual Robot Rescue Competition included 11 competitors from 10 different countries. The main objective of this competition is to utilize teams of robots to perform an urban search and rescue (USAR) mission over both indoor and outdoor terrains. For the first time, elemental tests were performed in autonomously generated map quality, multi-vehicle tele-operation, and communication's system deployment. In addition, a comprehensive USAR scenario was performed. This year's competition featured new performance metrics and automatic scoring programs. This paper presents an overview of the metrics for the competitions and lessons learned from their application during a high-intensity international competition.

Categories and Subject Descriptors

F.2.3 [**Theory of Computation**]: Analysis of Algorithms and Problem Features – *Tradeoffs among complexity measures.*

General Terms

Measurement, Performance, Experimentation, Human Factors, Standardization.

Keywords

Robotics, Evaluation, Competition, Simulation, Performance Metrics, RoboCup

1. INTRODUCTION

July 2009 saw the fourth annual running of the RoboCup Rescue Virtual Robot Competition in Graz, Austria. The RoboCup competition [1] provided an international forum where approximately 400 teams, with 2000 participants from 35 countries came together to compete in the areas of robot soccer, rescue, service robotics, and junior leagues. For the Virtual Robot Competition (VRC), 11 teams from 10 countries (Austria, Brazil, China, England, Germany, Iran, Italy, the Netherlands, Spain and the USA) participated.

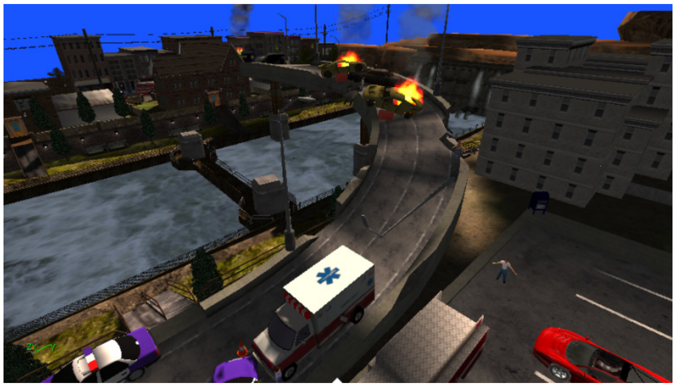


Figure 1: Example of the bridge accident scene from the outdoor environment used in the RoboCup07 competition.

In the past, the VRC was run as several rounds of Urban Search and Rescue (USAR) scenarios. Each scenario consisted of teams of robots striving to find as many victims as possible in an indoor or outdoor accident scene such as the one depicted in Figure 1. The scoring performance metrics were specifically designed to award research advances in the general areas of multi-agent cooperation, human-computer interfaces (HCI), and map building. Specific emphasis was placed on the formation of multi-agent communication networks, complex terrain navigation, and victim search and identification strategies. While certain aspects of the scoring were computed automatically, a significant part of the scoring metric was computed by hand by the technical committee of the competition. This scoring procedure was very time consuming and placed a large burden on the committee, thus limiting the number of teams that would be able to participate in the event. More information on the scoring metrics utilized in past competitions may be found in Balakirsky et al. [2].

While these performance metrics proved useful in determining the overall winner of the competition, it was not possible to get deep insight into why a team won. The individual components that constitute a team's capabilities were not evaluated; only the composite results. Since strength in several different areas is required to successfully carry out the mission, it may be stated that the team with the strongest weakest link would win the competition. The goal of this year's event was to change that. We wished to be able to determine which team had the strongest mapping, which had the best communications strategy, which had the best human-computer interface, AND which had the best overall system. To this end, we implemented the SCORE framework of evaluation [3] and evaluated three elemental tests as well as the overall USAR scenario. In addition, an effort was made to automate as much of the scoring procedure as possible. This allowed our three person technical committee to simultaneously work with four teams (two teams competing in an actual event and two teams setting up to compete in the next event).

The remainder of this paper is organized as follows: Section 2 provides an overview of the elemental tests and the final scenario. Section 3 describes the scoring metrics and the automated scoring tools that were utilized during the competition. Section 4 presents a summary of the results and lessons learned about the scoring metrics and Section 5 presents conclusions and future work.

2. COMPETITION OVERVIEW

The competition consisted of three preliminary events that were designed to test the individual team's capabilities in elemental skills followed by four comprehensive events that presented an overall search and rescue scenario. The points gained from the preliminary events (0 to 50 for each elemental test) were totaled to determine who would proceed to the semi-final round. The semi-final round presented two disaster scenarios (one indoor and one outdoor) to the remaining teams. Points were again summed to determine who would proceed to the final round. The final round was run in the same manner as the semi-finals. All of the environments used in the competition have been released to the public via our sourceforge website.[1]

An enhancement in the 2009 worlds included the use of elements that were directly borrowed from the RoboCup Rescue Physical Robot League. This league features real robots competing in physical arenas to provide maps of the environment and locate victims. For this year's competition the physical league's maze area was virtually constructed and replicated in order to fill the upper right room of the mapping challenge world (see Figure 2). Additional elements, such as step fields, appeared in all of the virtual competition worlds.

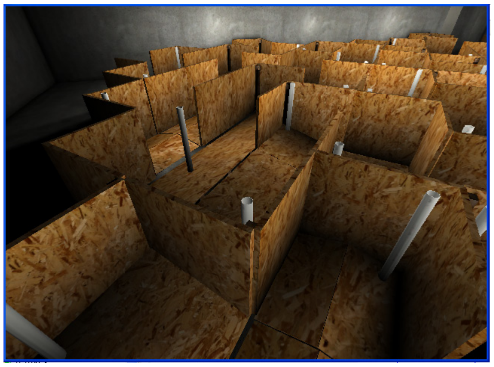


Figure 2: View on the physical league's maze inside the VRC mapping challenge environment.

2.1 Mapping Challenge

The first elemental skill test consisted of a mixed-autonomy (both tele-operation and autonomous operation) mapping challenge. The idea behind this event is that robots are given 20 minutes to map out the environment before the emergency responders enter the building. Once they enter, the emergency responders need to know the best routes to take to various newly discovered points of interest. No *a priori* data was provided to the teams for this event. Teams were allowed to use up to four robots to explore this indoor environment. During this exploration, the robots must communicate with each other through the use of a communications simulator and with the outside world through a communications station. The simulated sensors included a realistic noise model for both external (laser range finder and sonar) and internal (wheel encoders and IMU) sensors. The world was designed to stress the robot's sensors and algorithms by featuring both flat floored and sloped floor mazes, large featureless spaces, and various lighting conditions. The overall size of this world was 45m x 55m. No team was able to explore the entire environment with 4 robots during the 20 minute period.

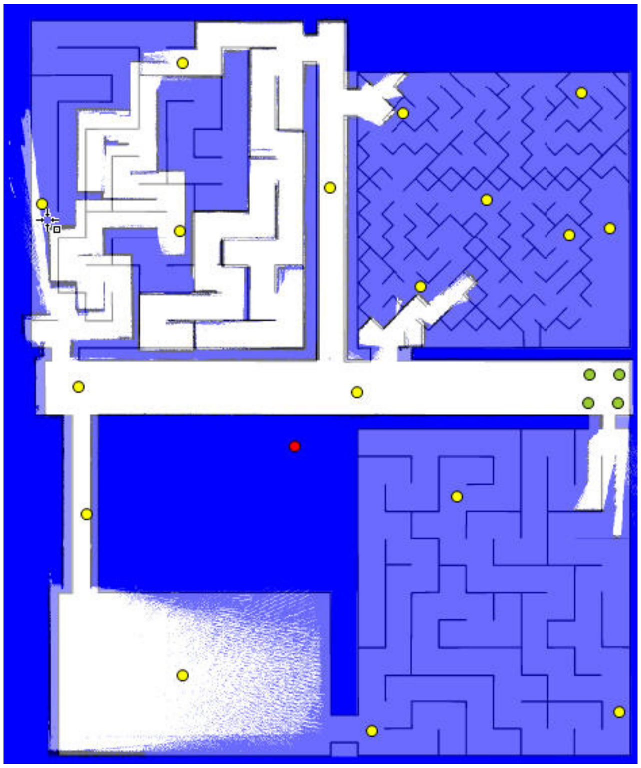


Figure 3: Team 1's mapping challenge map.

A sample of "Team 1's" map is shown in Figure 3. The green circles in the figure represent the starting locations of the four robots. Yellow circles are utilized for scoring and will be discussed in 3.1. The team's map (white) has been overlaid on the ground truth for the event. The upper-left and lower-right rooms consisted of flat floored mazes. The upper right room was an enlarged model of one of the physical robot's mapping mazes. The room on the lower-left was a large featureless and dark space. The robots were able to maintain a good connection with the

[1] http://sourceforge.net/projects/usarsim/files/Maps/3.31/RoboCup09.zip

communication's station (shown as the red dot) from anywhere in the world. Teams from this event also participated in the interleague challenge which featured real data collected from the physical leagues maze. For the interleague challenge, the teams ran their identical code from the simulation events on real data.

2.2 Deployment Challenge

The second elemental skill test consisted of a deployment challenge. Teams were allowed to use combinations of up to eight robots in this event. For this event, a radio propagation model was utilized to determine if robots were able to communicate with each other. This model computes radio signal attenuation based on a combination of the distance between the two robots and the number of objects (walls and obstacles) that the signal needs to pass through. A uniform signal loss was applied for each object penetration.

The idea behind this event was that emergency responders needed to enter and work in a building that was too large to have continuous communications coverage without repeaters. Each of the team's robots had a repeater mounted on it, and the teams were required to establish a communications network that covered as much of the building's interior as possible. Scoring for this event was based on the number of square meters of the building that had network connectivity with a communications base station. This measure was automatically generated.

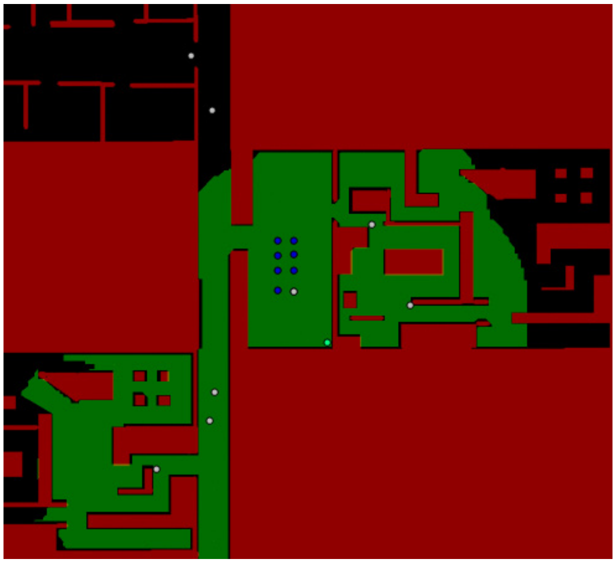


Figure 4: Section of Deployment map from Team 2.

Approaches to accomplish this task ranged from pre-planning the locations for the robots based on an estimate of the communication strength, to driving a robot until it reached the end of its communication range and then extending this range by driving a new robot into the frontier. No operator involvement was allowed during this test, so all robots had to navigate autonomously through the environment.

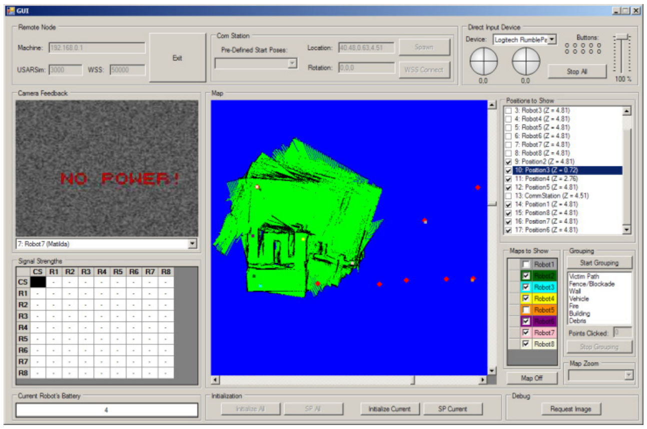
Figure 4 shows Team 2's map for this challenge. Team 2 pre-computed expected coverage and then autonomously navigated their robots to the computed locations. The blue dots in the figure represent the robot's starting locations and the white dots represent the robot's final locations. The green dot is where the communication station was located. The red areas are obstacles or outside of the building structures, while the green area represents the radio coverage. Teams were provided with *a priori* data for this event, but the data had several intentional errors with some rooms being blocked by collapses. Some teams miscomputed the radio coverage and drove their robots too far thus disconnecting them from the overall network. Team 2 was one of those teams. As may be seen in Figure 4, the two top robots are not connected to the overall network (i.e. there is no path from them to the communication's station) and thus did not generate any points for the team. In addition, not all of the robots were able to reach their desired ending locations. This was due to blockades and navigational challenges present in the terrain, which the robots had to overcome autonomously.

2.3 Tele-Operation Challenge

The final elemental skill test performed was a tele-operation challenge. For this challenge, teams were permitted to use 8 robots in order to reach 8 predetermined goal points. Each goal point was selected such that a particular class of robot was best suited to reach it. For example, there were elevated goal points, as well as goal points in small spaces. The teams did not have *a priori* data on which robot should attempt which goal location.

The idea behind this challenge is that emergency responders have knowledge of interesting locations that must be remotely examined. The teams need to reach these locations and provide feedback. In addition, past experience has shown that few teams were experimenting with novel robots and teaming arrangements. Therefore, another idea behind the tele-operation challenge was to introduce teams to a number of different platforms and to stress their human computer interfaces. Teams could use up to 8 robots, but could only use 2 of each robot class. This rule was put into effect in order to encourage teams to experiment with multiple types of robots and to form heterogeneous teaming arrangements.

An additional challenge for the teams was the lighting condition. All ground robots had to navigate through a small maze before they could reach the target points. In that maze it was quite dark, forcing teams which rely on visual feedback to fall back to the other sensors present on the robots.



Machine: 192.168.0.1

USARSim: 3000 WSS: 50000

Pre-Defined Start Poses:

Location: 40.48,0.63,4.51

Rotation: 0,0,0

Device: Logitech RumblePa

0,0 0,0 100 %

7: Robot7 (Matilda)

	CS	R1	R2	R3	R4	R5	R6	R7	R8
CS	■	-	-	-	-	-	-	-	-
R1	-	-	-	-	-	-	-	-	-
R2	-	-	-	-	-	-	-	-	-
R3	-	-	-	-	-	-	-	-	-
R4	-	-	-	-	-	-	-	-	-
R5	-	-	-	-	-	-	-	-	-
R6	-	-	-	-	-	-	-	-	-
R7	-	-	-	-	-	-	-	-	-
R8	-	-	-	-	-	-	-	-	-

4

- □ 3: Robot3 (Z = 4.81)
- □ 4: Robot4 (Z = 4.81)
- □ 5: Robot5 (Z = 4.81)
- □ 6: Robot6 (Z = 4.81)
- □ 7: Robot7 (Z = 4.81)
- □ 8: Robot8 (Z = 4.81)
- ☑ 9: Position2 (Z = 4.81)
- ☑ 10: Position3 (Z = 0.72)
- ☑ 11: Position4 (Z = 2.76)
- ☑ 12: Position5 (Z = 4.81)
- □ 13: CommStation (Z = 4.51)
- ☑ 14: Position1 (Z = 4.81)
- ☑ 15: Position8 (Z = 4.81)
- ☑ 16: Position7 (Z = 4.81)
- ☑ 17: Position6 (Z = 4.81)

- □ Robot1
- ☑ Robot2
- ☑ Robot3
- ☑ Robot4
- □ Robot5
- ☑ Robot6
- ☑ Robot7
- ☑ Robot8

Victim Path
Fence/Blockade
Wall
Vehicle
Fire
Building
Debris

Points Clicked: 0

Figure 5: Example of GUI from the Tele-operation challenge.

This challenge was automatically scored by a metric that evaluated the number of goals reached and the distance from the goal that the robot was able to achieve. A screen shot from a sample GUI is shown in Figure 5. This figure shows the map that the robots are generating as well as unreached goal locations (the red dots).

2.4 Semi-Finals

The semi-finals took the competition back to its roots of performing a tele-assisted multi-robot rescue mission. The top five teams ran through both an indoor and outdoor disaster environment. The environments were challenging with large areas of uneven terrain. The semi-finals incorporated a complex scoring metric that included automatic and hand-generated scores. The metric included the amount of area cleared by the robots (guaranteed to be victim free), victim scoring that included the number of victims found and various attributes of the victims (location of injuries, physical description, …), and map quality points.

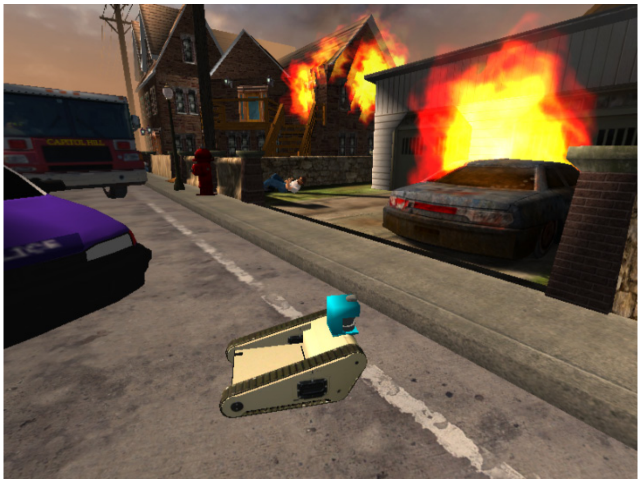


Figure 6: Example of the outdoor disaster environment.

The original idea behind the semi-finals was to award the 3rd place prize and have only the top two teams proceed to the finals. This idea was abandoned, because after the semi-finals there was only a one point difference between 3rd and 4th places Therefore, a decision was made to have a 3rd place runoff the next day.

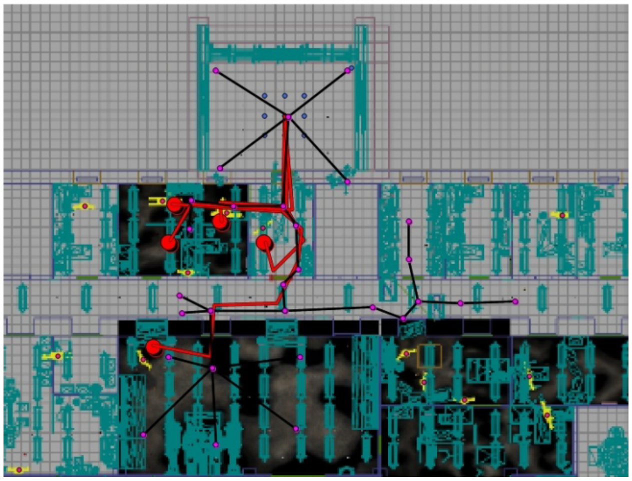


Figure 7: Vector components from final's indoor map.

2.5 Finals and Run-off

Another set of worlds was utilized for the finals and run-offs. These were the most complex worlds that have ever used since the start of the competition in 2006. The top two teams were able to provide both pixel maps (in our standard color scheme) and MapInfo Data Interchange Format (MIF) formatted vector skeletons. One such skeleton (with black arcs and purple nodes) is shown in Figure 7. This was a new feature of this year's competition and shows one way that we are encouraging innovation amongst the teams. Figure 7 also displays other innovative vector information such as responder paths to located victims (red lines) and detailed victim information (each red dot is tagged with information on victim location and vital signs). Raster map components included geo-referenced victim pictures that can be displayed on the map as well as a raster obstacle map.

3. AUTOMATIC SCORING

As previously stated, it was desired to provide automatic scoring programs whenever possible. During this competition, several new scoring techniques were examined with various degrees of success. These techniques are evaluated in the following sections.

3.1 Mapping Challenge

The evaluation of autonomously generated maps is an open question in the current literature. Many approaches tend to treat the map as an image and apply various image processing techniques to the map in order to judge its quality. One such technique is presented by Varsadan et al. in [5], where an image similarity metric is used to compare robot-produced maps against their ground truth equivalents. Past VRC events utilized a combined metric that had such a comparison at its heart. More information on previous year's map evaluation metrics may be found in [5]. A problem with such a metric is that it is likely to negatively affect maps with a single misalignment that propagates through the rest of the map. Collins et al. [6] augment a purely image-based evaluation approach by adding a measure that assess if a path generated on the robot-generated map would be valid on the ground truth map. This approach is useful if another robot will be utilizing the autonomously generated map as ground truth for planning its own routes.

It is said in [8] that any map assessment method should be intimately tied to the practical task for which the map will eventually be used. In the case of the VRC, this would be for a emergency responder to utilize the map to find a path to a point of interest. These points would be specified in terms of features (e.g. fourth office on the left) instead of geodetic coordinates. It was hoped that this evaluation technique would reduce the problem of misalignments propagating through the map and distorting the map's score. The actual approach that was implemented was to choose several random destinations from the environment as points of interest (POI). Paths to these POIs would then be computed on the team's maps and evaluated for their topological correctness.

The problem with this approach is that the team's maps are delivered as grid-based images and not as topological structures. Therefore, a technique for extracting the topological information needed to be developed. To solve this task, the POIs were manually mapped to the team's map and a standard path generating algorithm [9] was run to compute a path solution. The topological properties of this path were then evaluated and used to determine the map's score.

Unfortunately, this automatic procedure was not finished in time for the competition and topological paths were generated by hand on the competitors map. While not the ideal solution of having an automated scoring tool, this procedure was easy and quick to implement and provided valuable insight to the value of this scoring metric.

3.2 Deployment Challenge

The deployment challenge saw the first application of a fully automatic scoring technique applied to the competition. The Java scoring application may be found at sourceforge[2].

The program faithfully replicates the equations used by the Wireless Communication Server (WSS) server in order to determine the signal attenuation between a transmitter and a receiver located at arbitrary positions in the world. The attenuation considers both degradation due to increasing distance, and the presence of obstacles. The overall score is computed as follows. Once the challenge is over, final positions of all robots are retrieved from the log files. These log files are automatically generated by the simulation system and contain full ground truth of the simulation run. The position of the communication base station, instead, is fixed and known to all participants. A *connectivity graph* is then created. The connectivity graph is a graph whose vertices are the robots and the communication base station. An edge is added between two vertices if the corresponding elements can communicate with each other. In order to determine if two elements (robots or base station) can exchange data, the formerly mentioned equations are used to compute the signal attenuation between the two. If the signal strength is above a given threshold (-93 dB during the competition) an edge is added between the vertices. Once the graph is available, a breadth-first graph search is computed having the base station as the source vertex. All vertices reachable from the source are considered in communication with the base station. The rationale is that they can send information to the base station either directly or indirectly. Once the set of robots connected to the base station is known, the whole environment is sampled on a grid with a given resolution (specified by the user). A point in the grid is considered in communication range if it is connected to one or more robots reachable (directly or indirectly) from the base station. The overall score is the number of sampled points that are connected.

3.3 Tele-operation Challenge

The tele-operation challenge was also scored automatically by a program located at sourceforge[3] using the automatically generated log files. The goal of the challenge was to bring at least one robot to each target location in the allocated time.

A target location was considered reached if at least one robot was within *T* meters of location where *T* was a constant determined by the judges and known to the teams before the competition. For the 2009 event, this constant was set to 2 m. 50 points were awarded for each point reached. In addition, the program automatically summed the distance of the target location from the deployment site. This distance was an indication of the difficulty to reach this point. To be able to rank teams which reached exactly the same targets, the distance of the robots to those target locations was subtracted from the score.

The scores of all three challenges were normalized relative to the score of the best team, which gives each challenge an equal weight.

3.4 Semi-finals and Finals

The scoring of the full scenarios involved a combination of automatic scoring and hand scoring. Points for world exploration and victims were computed with automatic programs[4] while map quality assessments were performed by hand. For world exploration, a team's map was first trimmed to remove any out-of-bounds areas or poorly covered areas that were claimed to be explored. An example of a poorly covered area may be seen in the upper right corner of the lower left room of Figure 3. Here a team presents stripes of explored area mixed with stripes of unknown areas. An automatic program was run on the resulting image that computed the area of map that was explored (seen by the robot) and the area that was cleared (guaranteed to be victim free). Exploration points were normalized to a maximum of 50 points.

Victim points included points for correct victim localization and attribution, and subtractions for incorrect localizations and victims

[2] http://usarsim.cvs.sourceforge.net/viewvc/usarsim/usarsim/Tools/ScoreRadio/

[3] http://usarsim.cvs.sourceforge.net/viewvc/usarsim/usarsim/Tools/ScoreTeleOp/

[4] http://usarsim.cvs.sourceforge.net/viewvc/usarsim/usarsim/Tools/ScoreVictims/

that resided in "cleared" areas and were not detected. Victim localization was computed automatically while attribution needed to be hand computed. Victim points were again normalized to a maximum of 50 points.

The final area of scoring was in the computation of skeleton quality, metric quality, and attribution of team provided maps. These scores were computed entirely by hand following the procedure outlined in [2].

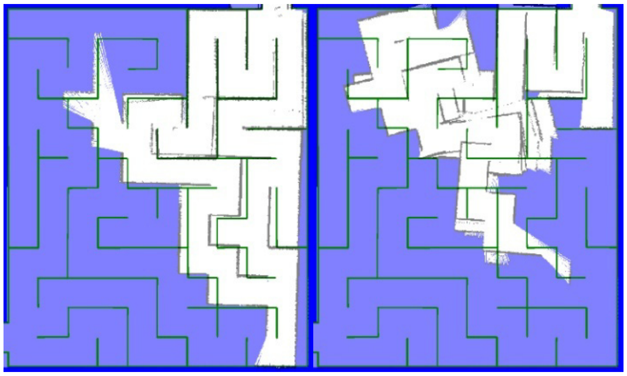


Figure 8: Image of two different team's maps for the same area of the mapping challenge world.

4. SUMMARY OF RESULTS

Overall, we were very happy with the results of the automatic scoring. However, issues did arise with the proposed automatic scoring of the mapping challenge. The intent of the scoring metric was to select several pseudorandom points in the environment and to then compute routes to these points. However, mapping errors made is difficult to place these points on some of the competitor's maps (even by hand). For example, Figure 8 shows the maps from two of the teams. While the map on the left shows slight misalignments, the map on the right presents several rotational errors and scan mis-matches that have caused extra walls to be added.

The problem becomes one of determining where to place our pseudorandom points in the right hand map. This determination must be made before any topological map calculations may be made. One possible solution for this problem is to allow the teams to know the locations of the points before the run. The teams will then need to mark the point locations in their maps and routes will then be generated from the starting location to their marked points. The topological properties of these routes may then be judged against routes created on the ground truth map.

5. FUTURE WORK

While the current automation performed well, there are still several areas that need automating. Techniques need to be developed (or the metrics modified) that will allow for the automatic generation of scores for the semi-final and final rounds. In addition, the mapping challenge scoring program needs to be created and validated.

6. REFERENCE LIST

[1] H. Kitano, RoboCup-97: Robot Soccer World Cup I, Berlin: Springer-Verlag, 1998.

[2] S. Balakirsky, C. Scrapper, and S. Carpin, "The Evolution of Performance Metrics in the RoboCup Rescue Virtual Robot Competition," Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, . Aug. 2007.

[3] C. Schlenoff, M. Steves, B. Weiss et al., "Applying SCORE to Field-based Performance Evaluations of Soldier Worn Sensor Technologies," Journal of Field Robotics, vol. 24, no. 8-9. pp.671-698, 2006.

[4] I. Varsadan, A. Birk, and M. Pfingsthorn, "Determining Map Quality Through An Image Similarity metric." In *RoboCup 2008: Robot Soccer World Cup XII.* Springer Berlin/Heidelberg. Lecture Notes in Artificial Intelligence, volume 5339, pp. 355-365. 2009.

[5] B. Balaguer, S. Balakirsky, S. Carpin et al., "Evaluating Maps Produced by Urban Search and Rescue Robots: Lesssons Learned from RoboCup," *Autonomous Robots 27*, 2009.

[6] T. Collins, J. Collins, and C. Ryan, "Occupancy Grid Mapping and Empirical Evaluation," Proceedings of the Mediterranean Conference on Control and Automation, pp. 1-6, 2007.

[7] G. Fontana, M. Matteucci, and D. G. Sorrenti, "The RAWSEED proposal for representation-independent benchmarking of SLAM," Workshop on Experimental Methodology and Benchmarking in Robotics Research (RSS 2008), 2008.

[8] J.C. Latombe, "Robot Motion Planning", Kluwer Academic Publishers, Boston, MA, 1991.

RoboCupRescue Interleague Challenge 2009: Bridging the Gap between Simulation and Reality

Alexander Kleiner
University of Freiburg
Freiburg, Germany
kleiner@informatik.uni-freiburg.de

Chris Scrapper
The MITRE Corporation
McLean, VA, USA
cscrapper@mitre.org

Adam Jacoff
National Institute of Standards and Technology
Gaithersburg, MD, USA
adam.jacoff@nist.gov

ABSTRACT

The RoboCupRescue initiative, represented by real-robot and simulation league, is designed to foster the research and development of innovative technologies and assistive capabilities to help responders mitigate an emergency response situation. This competition model employed by the RoboCupRescue community has proven to be a propitious model, not only for fostering the development of innovative technologies but in the development of test methods used to quantitatively evaluate the performance of these technologies. The Interleague Challenge has been initiated to evaluate real-world performance of algorithms developed in simulation, as well as to drive the development of a common interface to simplify the entry of newcomer teams to the robot league. This paper will discuss 1) the development of emerging test methods used to evaluate robotic-mapping, 2) the development of a common robotic platform, and 3) the development of a novel map evaluation methodology deployed during the RoboCupRescue competition 2009.

Keywords

SLAM, Performance Metric, Simulation, Mapping, Rescue

1. INTRODUCTION

Response robots refer to a broad class of mobile robots intended to assist emergency response personnel in a variety of application domains; such as Urban Search and Rescue (USAR), Explosive Ordnance Disposal (EOD), and Intelligence, Surveillance, and Reconnaissance (ISR). These platforms serve as an extension of the operator to improve remote situational awareness and to provide assistive capabilities that minimizes the risk to responders and maximize the effectiveness and efficiency of a response in a tactical environment. Although recent advancements in the technical capabilities of these robots have improved the flexibility, utility, and survivability of overall system, in large these systems have failed to achieve a technology readiness level suitable for fielded systems deployed in their respective operational domains.

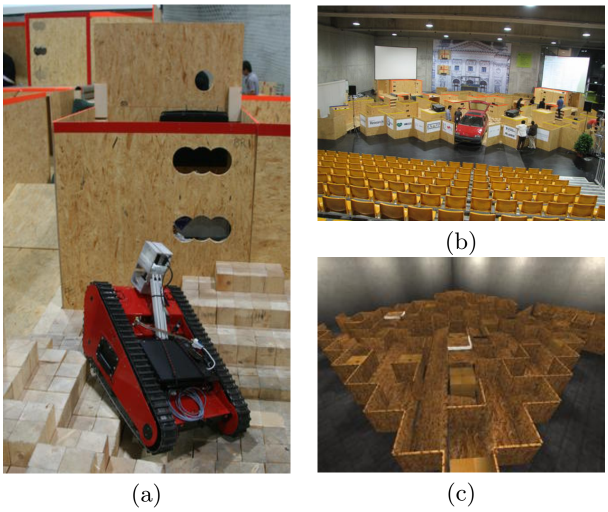


Figure 1: (a) Common robotic platform. (b) 2009 RoboCupRescue Maze in Graz. (c) Virtual maze in USARSim.

Test methods establish a confident connection between developers and consumers regarding the expectations and performance objectives of robotic technologies. This is a cardinal step in fostering innovation and assessing the maturity of evolving technologies. They consist of well-defined testing apparatuses, procedures, and objective evaluation methodologies that isolate particular aspects of a system in known testing conditions [ASTM, 2007]. The development of test methods start with a comprehensive analysis of the application domain to identify requirements with associated metrics and the range of performance, starting from a baseline threshold to the objective "best-case" performance. This analysis provides the basis for developing test methods and testing scenarios that are intentionally abstract so as to be repeatable across a statistically significant set of trials and reproducible by other interest parties. This also provides developers with a basis for understanding the objective performance of a system and allows consumers to confidently select systems that will meet their requirements.

Robotic competitions have also proven to be a propitious model for fostering innovation and assess the maturity emerging robotic technologies. Commonly, test methods provide the basis for evaluating the performance of robots in the competitions; setup as either elemental tests or embedded in operational environments to produce testing scenarios. For instance, Defense Advanced Research Project Agency (DARPA) and the European Space Agency have employed this model to assess autonomous ground vehicle in urban environments [Darpa, 2007] and in extraterrestrial environments, such as lunar landscapes or Mars exploration [ESA, 2008]. Not only do these competitions provide a means to assess the performance of emerging technologies, they also provide feedback as validity of the tests themselves.

The RoboCupRescue initiative also leverages the competition model to foster the research and development of the key capabilities to assist in the mitigation of an emergency response situation. This initiative partitions the emergency management problem into three competitions; the RoboCupRescue Robot, the RoboCupRescue Virtual Robot Simulation, and RoboCupRescue Agent competitions. Each of the competitions explore the partitioned problem space at different levels of abstraction, ranging from search and rescue of a single building to the development of an Incident Command System. The relevance of this initiative can be gauged according to two aspects: 1) the ability of the competitions to communally develop comprehensive systems capable of achieving an appropriate technology readiness level, and 2) the development of quantitative benchmarks and test methods capable to assess emerging technologies and assisting responders making deployment and purchasing decisions.

Therefore, the 2009 RoboCupRescue competitions sponsored an Interleague Challenge between the Robot competition and the Virtual Robot Simulation competition to demonstrate how well robotic algorithms developed in simulation can perform on a real robot. The challenge utilizes a common robotic platform and emerging standard test methods to explore the stability and accuracy of online mapping technologies, emphasizing the impact on an operator's ability to efficiently and completely map an unknown environment. This paper will discuss the development of emerging test methods used to evaluate robotic-mapping, the development of a common robotic platform, and the development of a novel map evaluation methodology deployed during the 2009 RoboCupRescue Interleague Challenge.

The remainder of this paper provides an overview of the Interleague Challenge and is structured as follows. Section 2 provides an overview of the test methods and robotic platform employed at the challenge. Section 3 will detail the 2D map evaluation framework used to evaluate the competing maps at the challenge. This map evaluation framework consists of a ground truth generation process and a map assessment tool. The evaluation results from the challenge are presented in Section 4, followed by the conclusion in Section 5.

2. THE INTERLEAGUE CHALLENGE

The evaluation of robot-generated maps is often based on qualitative approaches that do not take into account how specific environmental conditions, differing sensor configurations, and *in situ* decisions impact the performance of the system. While this type of analysis provides some indication of the overall performance, it does not allow researchers to identify problematic situations, analyze how errors propagate throughout the system, or compare the performance of one system with competing approaches.

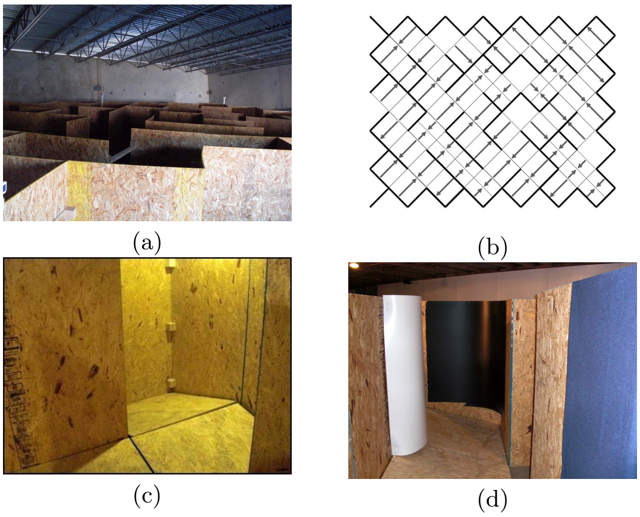

(a) (b) (c) (d)

Figure 2: (a) The Maze is a testing apparatus that limits complexities in the environment in order to evaluate the objective performance of particular mapping systems. (b) The configuration of the maze, whose dimension is 15 meters by 10 meters, with black lines representing the maze layout and the gray arrows showing the configuration of 15°pitch and roll ramp flooring, shown in (c). (d) The maze utilizes a variety of materials and shapes to produce additional mapping features, such as concave and convex surfaces.

To address these issues the Interleague Challenge leverages emerging test methods for robotic mapping and the development of a common robotic platform to enable repeatable and reproducible testing scenarios. This constrains the variability in the test to abet in the evaluation of mapping algorithms as they transition from a purely virtual world to the real world. The remainder of this section is dedicated to describing the testing method and common robotic platform used for this challenge.

2.1 A Test Method for Robotic Mapping

Associated Metrics

Arguably, the most common mapping paradigm employed for robotic navigation is the metric mapping paradigm. This intuitive mapping paradigm provides a representation where the spatial relationship between any two objects in the map is proportional to the spatial relationship of the corresponding objects in the actual environment. Therefore, assessing the quality of metric maps is based on the spatial consistency of features, such as walls and hallways, between the map produced by the robot and the ground truth map of the actual environment. Error propagation and sensitivity to performance singularities [Scrapper *et al.*, 2008] idiosyncratic to most robotic mapping systems suggests the associated metric needs to quantify the local (or regional) consistency of areas within the map, as well as the global consistency of the overall map.

Apparatus

The Maze testing apparatus, seen in Figure 2, is part of an emerging suite of test methods for characterizing the performance of the robotic mapping for response robots [ASTM E54.08, 2009]. The apparatus is essentially a random maze, whose overall dimension is 15 meters by 10 meters, and is constructed from oriented strand board (OSB) to form a series of hallways that are approximately 1.2 meters wide. The non-flat flooring, comprising of 15°ramps, makes the vehicle pitch and roll as it traverses the maze. The modularity of the apparatus enables the randomization of the maze configurations for repetitive testing. The use of OSB to construct this apparatus provides a surface friction similar to dust covered concrete and a cost-effective testing apparatus that is easy to replicate.

This apparatus was chosen for the Interleague Challenge because it provides a feature-rich scenario that limits environmental complexities. This should provide the best-case scenario, where mapping systems should perform optimally. The configuration of this apparatus generates a closed set of distinct mapping features and vertical walls that provide a mapping system with distinct observations throughout the apparatus. The presence of distinct features and the lack of occlusions found in this apparatus reduce the uncertainty in the mapping system when associating features, which increases the likelihood of determining valid correspondences. Additionally, surfaces perpendicular to the motion of the vehicle are present in almost every scan. This increases the ability of the system to make accurate measurements of the surrounding environment and accentuates the displacement of features between observations of the external world. Limiting the environmental complexities allows developers to tune their systems and establishes a baseline for comparison.

2.2 A Common Robotic Platform

A common robot with a fixed sensor configuration is used to challenge researchers and ensure compatibility. As mentioned above, this platform also support the entry to robotics of newcomer teams.

Robotic Platform

The base robotic platform used for the Interleague is designed to provide a cost-effective robotic platform capable maneuvering in the complex terrain. The dimensions of the common robotic platform is .533m x .304m x .762m (21 x 12 x 30 inches) , it weighs approximately 18 kg (40 lb), and it is equipped with 4 rechargeable 12 volt batteries allowing it to operate in for up to 10 hours. The platform is capable of operating in all weather conditions and a high degree of mobility allows it to navigate on rough terrain and to climb stairs up to 50° inclination. The motors are equipped with encoders for controlling set-velocities and computing the wheel odometry.

Sensor Configuration

The sensor configuration is defined by two statically mounted laser scanners (one horizontal scan and one vertical scan), and an inertial measurement unit (IMU). Both laser scanners deliver range readings with a field of view of 240° with an angular resolution of 0.36° and a range of approximately 4 meters. The IMU provides measurements of the three Euler angles; yaw, roll, and pitch. Although the IMU supports compass-based yaw measurements, i.e., orientation measurements relative to magnetic North, the current implementation does not utilize this information. Therefore, the orientation angles obtained from the IMU are based solely on measurements take from the gyroscopes and accelerometers. This is motivated by the fact that, particularly in harsh environments, magnetometer readings can strongly be perturbed.

Robotic Interface

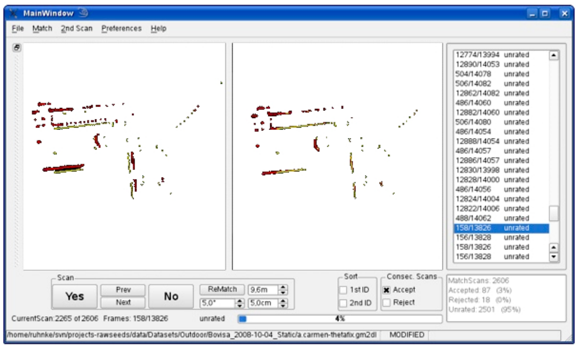
The platform provides two robotic interfaces: a low-level serial interface and a high-level robotic interface. The serial interface provides developers with direct access to the on-board controller, giving them direct access to motor commands and supporting the integration of both laser scanners and the IMU via an USB interface implementing a vendor specific protocol. In order to create a more user-friendly interface, the common robotic platform has implemented a high-level robotic interface. This interface is built on a wrapper application that automatically collects data from all the sensors and provides them via a TCP/IP server executed on-board the robot. The messaging protocol used by this process server is chosen according to the protocol of the USARSim simulation application [Wang and Balakirsky, 2009]. Thereby users can control the robot and access sensor data in the same way as if they would connect to the USARSim simulator used in the RoboCupRescue Virtual Robot Simulation competition. Hence, the migration of software developed in simulation towards execution on the real platform is simplified.

3. A 2D MAP EVALUATION FRAMEWORK

As mentioned in Section 2.1, the development of an evaluation framework for quantifying the performance of robotic mapping algorithms in the metric paradigm requires two key components. First is the ability to obtain or generate an accurate ground truth of the test method. Second is the ability to quantify the local consistency, incremental errors arising in bounded regions within the map, as well as the globally consistency, accumulation of errors throughout the entire map. The remainder of this section will describe a 2D map evaluation framework consisting of a generalized ground truth generation methodology (Section 3.1), and a methodology for quantifying the performance of the mapping algorithms. (Section 3.2).

3.1 A Ground Truth Generation Tool

The process developed for generating ground truth of the test methods at the Interleague Challenge, as described in this section, is an attempt to automate the production of ground truth with minimal *a prior* information about the environment. However, this ground truth generation process requires human intervention. Depending on the length of the runs and sampling frequencies employed, this process can be laborious. It is the belief of the authors that some level of human intervention is necessary when developing ground



12774/13994	unrated
12890/14053	unrated
504/14078	unrated
506/14082	unrated
12862/14082	unrated
486/14060	unrated
12882/14060	unrated
506/14080	unrated
486/14054	unrated
12888/14054	unrated
486/14057	unrated
12886/14057	unrated
12830/13998	unrated
12828/14000	unrated
486/14056	unrated
12824/14004	unrated
12822/14006	unrated
488/14062	unrated
158/13826	unrated
156/13828	unrated
158/13826	unrated
156/13828	unrated

9,6m
5,0°
5,0cm

□ 1st ID
□ 2nd ID

☒ Accept
□ Reject

MatchScans: 2606
Accepted: 87 (3%)
Rejected: 18 (0%)
Unrated: 2501 (95%)

CurrentScan:2265 of 2606 Frames: 158/13826 unrated 4%

/home/ruhnke/svn/projects-rawseeds/data/Datasets/Outdoor/Bovisa_2008-10-04_Static/a.carmen-thetafix.gm2dl MODIFIED

Figure 3: Manual verification of automated laser scan alignment for ground truth generation.

truth of test environments, especially environments where *a priori* information is not readily available.

The process used for generating ground truth during the Interleague Challenge is a two-step process. The first step uses a *Simultaneous Localization and Mapping* (SLAM) algorithm, supervised by a *subject-matter expert* (SME), that directly processes raw sensor streams to estimate a globally consistent trajectory and to derive initial displacement candidates $\delta_{i,j}$, referred to in this paper as *constraints*. During the second step the SME is required to inspect every constraint and verify the displacement. For example, the SME inspects two consecutive observations obtained from the laser range finder and the corresponding pose estimates at time step i and j belonging to the two poses of the constraint, respectively (see Figure 3). During the verification process, the SME can either *accept* or *reject* the constraints previously estimated by the SLAM algorithm. In the case of an *accept*, the SME verifies the final constraint, $\delta^*_{i,j}$, and adjusts the displacement as necessary.

3.2 A Map Assessment Tool

Many of the methodologies being developed for quantifying metric maps employ feature extraction and image registration techniques on an occupancy grid (or image) of the global map. Commonly, these approaches do not take into account the resolution and scale of the grid and do not consider the local consistency within the global map. Therefore, the approach employed at the Interleague Challenge attempt to quantify the errors in a map using the *relative* displacements between poses, i.e. relative motion, as the evaluation criteria. This motivation for using this evaluation criterion is based on two anecdotal factors. First, to consider the pose instead of the resulting occupancy grid based map is advantageous because features in grid-based maps can become obscured beyond recognition, especially after closing loops, although the actual pose error is within centimeters. Second, using the relative displacements of the poses minimizes error propagation during the evaluation, providing a more accurate metric that isolates regional errors when considering the absolute pose error. For example, pose estimates towards the end of the run are not impacted by the pose errors that occurred at the beginning of the run.

Essentially this map assessment tool quantifies errors in a robot-generated map by comparing the constraints from the ground truth generation process and the pose estimates being produced by a given mapping system being considered. An associative relationship is built between the constraints and the pose estimates based on time [1]. It then uses this associative relationship to compute the error in the relative displacements for each of the sets of corresponding data.

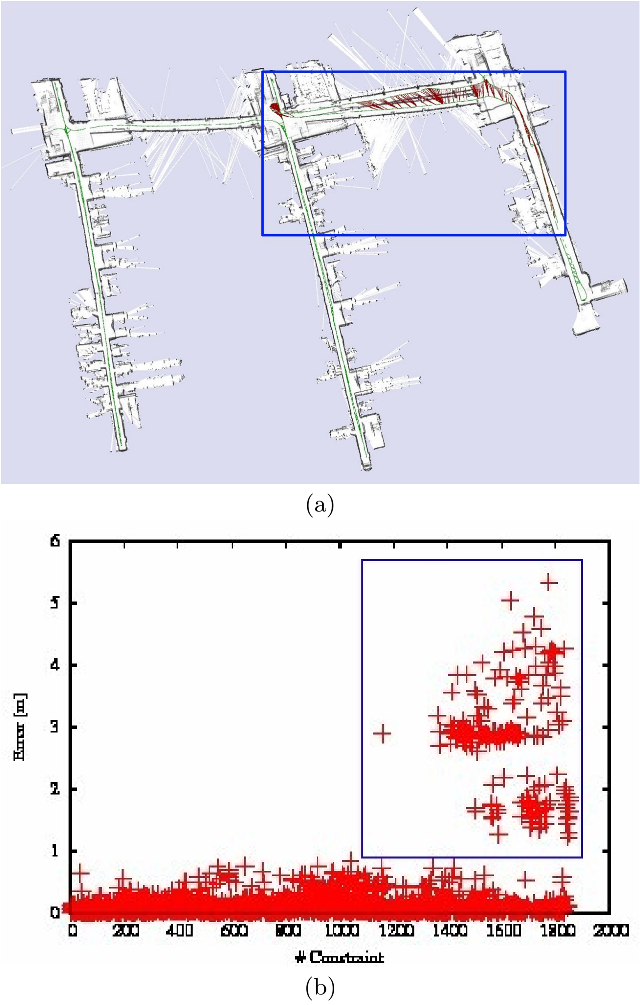


Figure 4: An example illustrating the utilized metric for map assessment. (a) A visualization of the ground truth constraints (red lines) superimposed on the map generated by the SLAM algorithm. (b) The error plot showing the relative displacement error at each constraint plotted over time. As depicted by the blue rectangle the error increases drastically in the last third where the robot returned after driving a long hallway without features.

The map evaluation process can be expressed formally as follows: Given the estimated trajectory of a robot, $x_{1:T}$, consisting of a series of the pose estimates, x_i, produced by the mapping system at timestep i from 1 to T. Let $x^*_{1:T}$ be

[1] While correlating data between two systems based on time is suitable in some situations, it is not a good assumption and it will be addressed in later versions of the tool.

the set of the corresponding reference poses generated during the ground truth process. The *relative* displacement can then be defined as $\delta_{i,j} = x_i \ominus x_j$, where $\oplus$ is the standard motion composition operator and $\ominus$ its inverse. Instead of comparing x to x^* (in the global reference frame), the operation is based on δ and δ^*as

$$\varepsilon(\delta) \quad = \quad \sum_{i,j}(\delta_{i,j} \ominus \delta^*_{i,j})^2. \qquad (1)$$

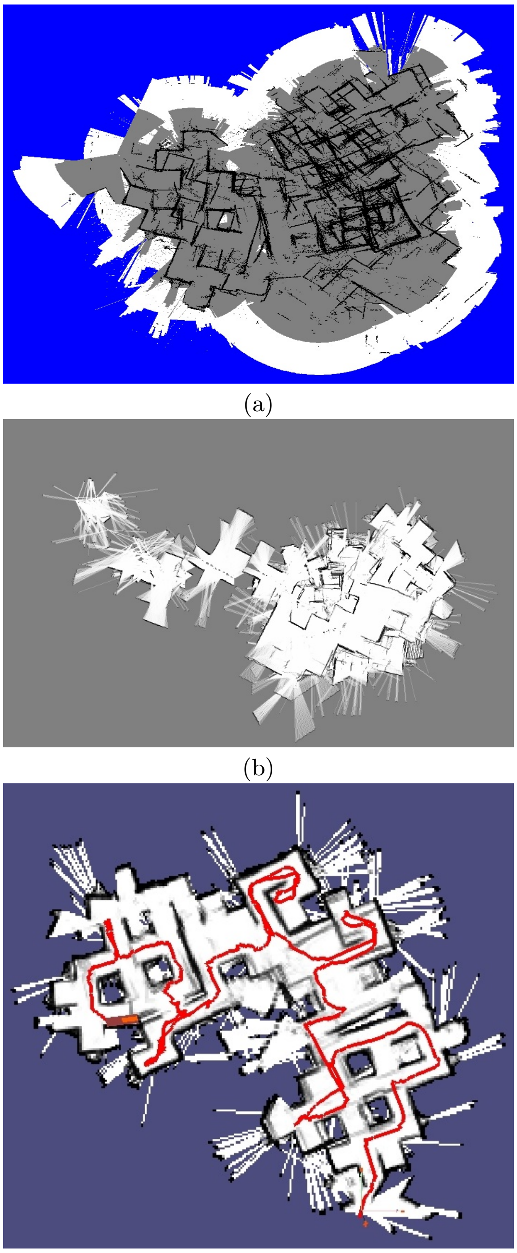
The major advantage of this tool is the ability to adjust the resolution of the evaluation criteria, which is defined by the constraints produced during the ground truth generation process. This suggests that this tool can be dynamically adjusted to evaluate the global or local consistency of a map. It can also be used to identify and test performance singularities in mapping systems by evaluating how different environmental conditions impact a particular mapping system.

A demonstration of the map assessment used to evaluate a SLAM algorithm in a difficult environment with long hallways and glass walls is shown by Figure 4(a). Ground truth constraints are depicted by red lines in the map. Figure 4(b) shows the temporal analysis of errors in the map for each constraint over the course of the entire run. As can be seen, the error increases drastically in the last third (depicted by a rectangle), which corresponds to the real situation where the robot returned after driving a long hallway without features. This cluster clearly identifies a weak point in the estimated algorithm, which did not manage to close the loop in this particular situation, shown by the shearing effect in the corridor. A more detailed description can be found in previous work [Burgard *et al.*, 2009].

4. THE EVALUATION AND RESULTS

At the RoboCup competition in Graz 2009 the Interleague Challenge provided a testing scenario to facilitate the intercomparison of the robotic mapping algorithms developed in simulation and the applicability of these algorithms to real data captured from a real robot while operating complex terrain. This section will provide an overview of the challenge and present the results of the three teams demonstrating the most proficient performance. In the remainder of this section, these teams will be referred to as *Team 1*, *Team 2*, and *Team 5*.

Prior to actual the challenge the common robotic platform (described in Section 2.2) and a variation of the *The Maze* test method (described in Section 2.1) was modeled in USARSim (see Figure 1c) and provided to the teams. These models allowed teams to tune their mapping systems in this particular environment.



(a)

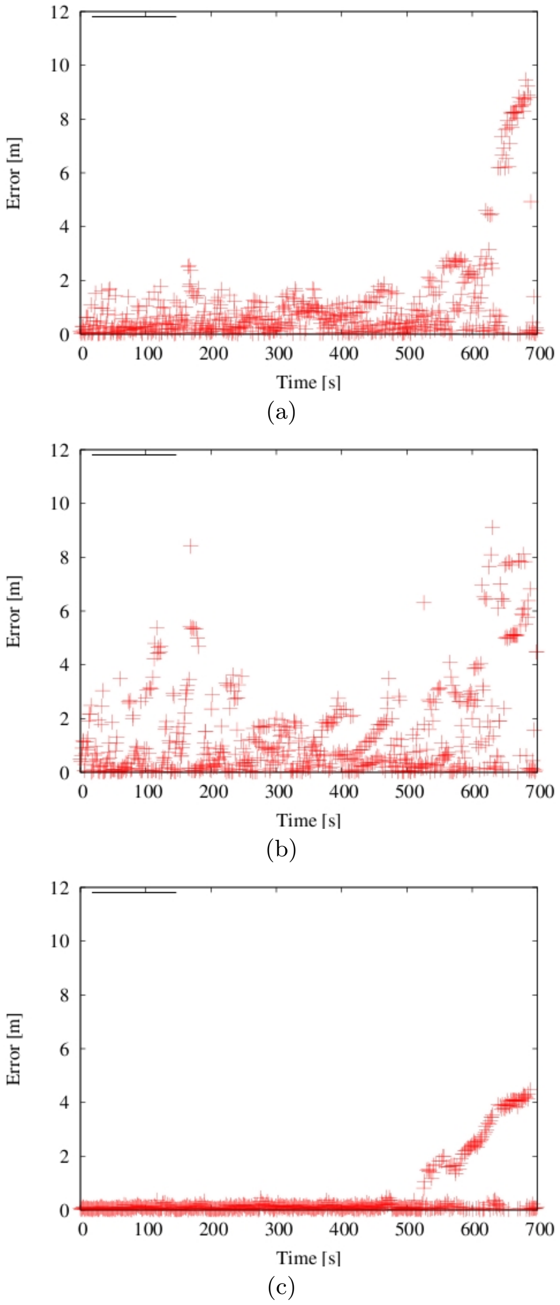
(b)

(c)

Figure 5: (a-c) Visual mapping results from team 1, team 2, and team 5, respectively.

For the actual challenge, the evaluation was based on a sensor data set logged while teleoperating the real common robotic platform through the maze built in support of the RoboCupRescue Robot competition, shown in see Figure 1b. The sensor data set logged the data streams of the two laser range finders, mounted vertically and horizontally, and the IMU sensor using the USARSim message protocol. A simple server application, simulating the USARSim interface to the actual robot, publishes the contents of the sensor data set over a TCP/IP socket at the same data rates found on the actual robot. Simulating a USARSim connection to a real robot not only reduces the integration required for teams to deploy their mapping algorithms but provides a repeatable testing scenario that eliminates the impact of *in situ* decisions on the mapping process.

Teams participating in the mapping challenge ran their respective mapping algorithms against the data from the real robot using the server application mimicking the USARSim



(a)

(b)

(c)

Figure 6: (a-c) Visualization of position errors at each constraint for team 1, team 2, and team 5, respectively. As can be seen, the resulting metric corresponds to the visual impression given by the according maps.

interface to the real robot. At the end of the run, each of the teams were required to submit a log file reporting their results as a list of pose estimates, x_t, y_t, θ_t, computed for each observation at time t. Each of the tuples in the log file, denoted as a $\langle x, y, \theta, t\rangle$, corresponds to observations used to construct ground truth constraints used in the map assessment tool, described in Section 3.2.

Table 1: Quantitative results (avg. position error)

Team	Abs. Err. $[m]$	Sqr. Err. $[m^2]$	Max. Err. $[m]$
1	1.08 ± 1.81	4.44 ± 14.84	9.46
2	1.40 ± 1.77	5.08 ± 11.74	9.11
5	0.48 ± 1.04	1.32 ± 4.00	4.46

The resulting metric maps from each of the teams are shown in Figure 5. Qualitative analysis suggests that divergent behavior has occurred in each of the approaches but does not provide any empirical evidence on which team was able to more accurately map the environment. For instance, the map produced by Team 5 appears to be more consistent but this does not provide a basis for distinguishing differences in the maps produced by Team 1 and Team 2. Additionally, the convolution of features in the maps makes it hard to assess if the divergent behavior seen in the map is due to regional errors that have propagated through the system or a catastrophic error.

The application of the map assessment tool quantifies the errors as they arise during the run and provides the means to temporally assess the errors that occurred in each of the respective maps. The error plots shown in Figure 6 show the relative displacement as compared to the constraints formulated during the ground truth generation process. This temporal analysis shows the stability of the mapping system and helps classify the impact of the errors that have occurred. This shows the map produced by Team 5 appears to be more stable then the competing approaches but is plagued with some regional errors arising during the course of the run, probably due to problems arising from loop closure at the end of the run. It also indicates that the mapping system deployed by Team 1 is more stable then Team 2 and could be suffering from as configuration or data association problem. This information could be used by developers to improve their approach.

The overall performance of the competing mapping systems is summarized in Table 1 by averaging the relative displacements measured by the map assessment tool over the whole run. Again this shows that the approach deployed by Team 5 outperformed the other two approaches.

5. DISCUSSION

The Interleague Challenge made its debut at the 2009 RoboCup Competition in Graz, in the first attempt to literally bridge the gap between simulated and real robot platforms. The challenge, built on a common robotic platform and a common testing apparatus, laid the foundation for not only assessing the applicability of algorithms developed in simulation on real data but also the need to develop quantitative metrics and test methods capable of evaluating the local and global consistency of robot generated maps. This challenge also exemplifies there is still work to be done.

The undulating terrain found in the maze presented a particularly problematic environment for the fixed sensor configuration. As the robot traversed the pitch and roll ramps, the horizontally aligned laser range finder periodically scanned the ceiling and floor of the maze. This produces artifacts in the data that further complicate the data association problem inherent to many techniques used for mapping, such as incremental scan matching. While there are mechanical techniques used to address this issue, i.e. actuating laser scanners to continuously level the orientation of the sensor with respect to world reference frame, the sensor configuration of the common robotic platform was intended to realize a survivable mapping solution that limits the number of moving parts. The surface friction of the OSB exacerbates the non-systematic errors in skid-steered vehicles rendering the encoder-based odometry almost useless.

In order to facilitate the transfer of technologies from simulated world to actual implementations on real robotic platforms, there needs to be a tighter coupling between simulated systems and their real world counterparts. For example, the simulated IMU or INS sensor in USARSim reports location and orientation of the robot assuming a Gaussian noise model, which is not consistent with available inertial measurement systems.

The evaluation of robot-generated maps is often based on qualitative approaches that do not take into account how specific environmental conditions or *in situ* decisions impact the performance of the system. While this type of analysis provides some indication of the overall performance, it does not allow researchers to understand what errors a specific system is prone to, how these errors impact the overall performance of that system, and how performance of that system compares with competing approaches. Developing testing scenarios for evaluating robot-generated maps can greatly benefit from the development repeatable and reproducible testing scenarios that isolate potential failure conditions in a controlled environment.

The development of a common robotic interface that enables developers to seamlessly transition robotic algorithms from simulation to the real world can help foster innovation and expedite the transfer of the technologies from the lab to fielded systems. The continuing development of this interface will lower the entrance barrier to robotics for newcomers and help improve the development cycle.

In the future, we plan to extend the Interleague Challenge to address other performance benchmarks for robotics; for example, an autonomous behavior challenge that will focus on the robot's ability to autonomously navigate a vaguely defined testing apparatus in a known way. In this challenge, teams will have to directly control the real robot via the USARSim interface according to a given task description.

6. ACKNOWLEDGMENTS

We would like to thank the teams during RoboCup 2009 in Graz, Austria for contributing their mapping results.

7. DISCLAIMER

Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by affiliated institutions, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

8. REFERENCES

[ASTM E54.08, 2009] ASTM E54.08. ASTM Committee E54.08.01 on Homeland Security Operations; Operational Equippment; Robots, 2009. http://www.astm.org/COMMIT/SUBCOMMIT/E5408.htm.

[ASTM, 2007] ASTM. In *Form and style for ASTM Standards.* ASTM International, March 2007.

[Burgard *et al.*, 2009] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós. Trajectory-based comparison of slam algorithms. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009. To appear.

[Darpa, 2007] Darpa. Darpa Urban Challenge, 2007. http://www.darpa.mil/grandchallenge/.

[ESA, 2008] ESA. Lunar robotics challenge, 2008. http://www.esa.int/esaCP/SEM4GKRTKMF_index_0.html.

[Scrapper *et al.*, 2008] C. Scrapper, R. Madhavan, and S. Balakirsky. Performance analysis for stable mobile robot navigation solutions. volume 6962, page 696206. SPIE, 2008.

[Wang and Balakirsky, 2009] J. Wang and S. Balakirsky. *USARSim V3.1.3 - A Game-based Simulation of mobile robots.* http://sourceforge.net/projects/usarsim/, 2009.

Mobile Microrobot Characterization through Performance-Based Competitions

Jason J. Gorman[a], Craig D. McGray[b], and Richard A. Allen[b]

[a]Manufacturing Engineering Laboratory
[b]Electronics and Electrical Engineering Laboratory
National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD 20899
jason.gorman@nist.gov, craig.mcgray@nist.gov, richard.allen@nist.gov

ABSTRACT

Recent advances in the design and fabrication of microelectromechanical systems (MEMS) have enabled the development of mobile microrobots that can autonomously navigate and manipulate in controlled environments. It is expected that this technology will be critical in applications as varied as intelligent sensor networks, *in vivo* medical diagnosis and treatment, and adaptive microelectronics. However, many challenges remain, particularly with respect to locomotion, power storage, embedded intelligence, and motion measurement. As a result, the National Institute of Standards and Technology has organized performance-based competitions for mobile microrobots that are designed to: 1) accelerate microrobot development by providing researchers a venue to demonstrate and observe novel technologies, 2) reveal the most pressing technical challenges, and 3) evaluate the most successful methods for locomotion and manipulation at the microscale (e.g., actuation techniques for crawling). This paper will discuss the goals and structure of the competition, results from past competitions, and plans to make performance characterization methods an integral component of future competitions.

Keywords

Microrobotics, microrobots, robot competition, performance characterization

1. INTRODUCTION

Microscale robotics, or microrobotics, has emerged over the last decade as the next wave in intelligent systems. As a result of scaling effects, microrobots have functionalities that will open up application paths that would otherwise have been impossible. Their small size and low unit cost allows them to be embedded into subsystems such as consumer electronics, and their small mass results in extremely high accelerations. Additionally, their size facilitates new modes of operation. As in many systems in nature, microscale robots can form large collaborative networks that can work either together to complete tasks faster or independently to cover more ground. This massive parallelism will result in complex system behaviors that have yet to be explored for macroscale robots. Microrobots are likely to have a major impact on advanced manufacturing, the health care industry, and the continued miniaturization of consumer products over the next two decades. However, this technology faces many new challenges with respect to fabrication, integration, control, power delivery, and embedded intelligence, among many others, which must be addressed for this field to find widespread acceptance.

This paper focuses on a mobile microrobot competition organized by the National Institute of Standards and Technology (NIST) that is designed to accelerate the adoption of this technology by U.S. industries. First, we give an overview of the field with an emphasis on the most successful methods for robot locomotion at the microscale. Next, the goals and structure of the competition are described along with the results from these competitions over the last three years. Difficulties in measuring a number of parameters for microrobots have been identified as a major limiting factor in the development of microrobots. Therefore, a list of measurement needs identified through the framework of the competitions is presented. Finally, plans for future competitions are presented, particularly with respect to the integration of microrobot metrology within the competition so that the performance of different systems can be directly characterized and compared.

2. MOBILE MICROROBOTS

The field of microrobotics is extremely broad and brings together a number of disciplines including microelectromechanical systems (MEMS), precision machine design, biology, materials science, and of course, robotics. Examples of common research thrusts include the manipulation and assembly of MEMS components, insect-inspired flying microrobots, the manipulation of particles and cells in solution, and the mobile microrobots discussed here (see [1] and [2] for an overview.) In all of these cases, either the robot or the manipulated object has microscale dimensions (i.e. between 1 mm and 1 μm). The term *microrobot* has also been used by some to simply mean a small robot (e.g., robots having dimensions on the order of centimeters) but this definition is not utilized here.

The dimensional scaling of robots and manipulated parts down to the microscale presents many challenges in microrobotics, including difficulties in precision fabrication, sensor and actuator integration, power delivery, and the interfacing between the micro- and macroscale domains. Most importantly, though, is the

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
ACM 978-1-60558-747-9/09/09

way in which the role of various forces changes between the macro- and microscales. Due to scaling effects [3], electrostatic, van der Waals, and capillary forces – among others – are significantly larger than inertial forces at the microscale. As a result, adhesion between robots, parts, and the surfaces that they interact with can limit dexterity and mobility. This is particularly true for mobile microrobots, which can easily become stuck while moving on a surface. Therefore, methods of locomotion that can overcome these adhesive forces, or even exploit them, are needed.

One of the first and most common methods of locomotion is based on the electrostatic scratch drive actuator. The untethered scratch drive actuator, developed by Donald et al. [4], consists of a conductive plate that has a bushing positioned on the bottom side of the plate near one side. When placed on a surface the scratch drive actuator sits at a slight angle to the surface due to the placement of this bushing. Motion in the plane is generated by applying an electrostatic force between the plate and surface that is large enough to make the plate snap down to the surface. Consequently, the edge of the plate that contains the bushing moves forward by a small increment (10 nm to 50 nm). When the electrostatic force is removed, the plate straightens but remains in the newly obtained planar position. The repetition of this sequence has been shown to yield repeatable motion with velocities approaching 2 mm/s. The electrostatic force is generated using an engineered surface composed of an interdigitated electrode array and a dielectric coating. By applying a voltage across the electrodes, the electrostatic force can be cycled at high rates (100 kHz) to yield high speed motion. However, this approach only provides unidirectional straight-line motion. Therefore, a turning arm has been added to the scratch drive actuator so that the microrobot can turn, leading to global controllability of the robot in the plane [5].

Electromagnetic forces have also been shown to be effective in actuating microrobots. Floyd, Pawashe, and Sitti [6] have demonstrated microrobots fabricated from a hard magnetic material, which can be as simple as a solid magnet block. Forces are exerted on the microrobot by uniform magnetic fields generated by macroscale multi-axis electromagnetic coils. By adjusting the control currents applied to the coils, the microrobot can be moved on a planar surface. The most repeatable motion has been achieved by applying a pulsed current signal along the desired motion direction, which results in a stick-slip motion caused by balancing the friction forces and electromagnetic forces.

Another electromagnetic actuation approach developed by Vollmers et al. [7] utilizes a resonant drive mechanism. Similar to [6], a set of electromagnetic coils is used to generate a controllable uniform magnetic field in the workspace of the microrobot. However, the microrobot's mechanical design is significantly different. The microrobot consists of two nickel blocks of different size that are connected by a metal spring. The magnetic field is modulated at the first resonant frequency of the mass-spring system to cause the two ferromagnetic blocks to vibrate relative to one another. When the vibration amplitude is large enough to cause the blocks to collide, the resulting impact force moves the microrobot in the plane. In addition to the electromagnetic forces, an electrostatic force is applied normal to the surface by operating the robot on an interdigitated electrode array as described above for scratch drive actuators. The electrostatic force is used to clamp the microrobot to the surface when the two masses are separated. Just before the two masses collide, the clamp is removed and the microrobot moves forward after the collision in a controllable increment.

Although other methods of locomotion have been demonstrated, including thermal impact drives [8] and piezoelectric crawlers [9], electrostatic and electromagnetic locomotion have been the dominant methods for microrobots. Each of the methods discussed above has also been extended to multi-robot control, which is essential in realizing the parallelism that makes microrobotic systems so powerful. Donald, Levey, and Paprotny [10] have shown that multiple electrostatic microrobots can operate on a single electrode array by designing each robot to have independent snap-in voltages for their scratch drive and turning arm. The electromagnetic robot in [6] has been shown to be extendable to parallel operation using a grid of independent electrode arrays for electrostatic clamping [11]. Finally, Kratochvil et al. [12] have demonstrated multi-robot operation using the resonant drive mechanism described in [7] by designing robots to have unique resonant frequencies, which can all be addressed through a single control signal for each degree of freedom of motion. Continued development of multi-microrobot systems is needed to fully utilize the nascent capabilities of microrobots working together collaboratively.

3. PAST COMPETITIONS

The mobile microrobot technologies discussed in the previous section have all been developed over the past decade and with the greatest momentum in the past three years. Although this field is in its infancy, it is a clear extension of the MEMS and robotics technologies that have become integral to many consumer products, manufacturing capabilities, biomedical tools, and military systems. However, mobile microrobotics is also a disruptive technology because microrobot designs are often not compatible with existing MEMS fabrication methods and the complexity of microrobot control presents new challenges in communications and power transmission at the microscale. Therefore, this field will require considerable investment to transition the technology to the marketplace. As a result, NIST has organized competitions over the past three years that are designed to accelerate development in this field while mitigating the risks in adoption of this technology by U.S. industries. The main goals of the competitions are to:

Assess the State of the Art - The competitions bring together a number of experts in the field along with their latest developments in mobile microrobotics. This provides the best vantage point to assess what is currently feasible and where this technology is going.

Accelerate Development - Competing teams must focus their technologies toward specific microrobot tasks in the competition and must meet hard deadlines to participate. This pressure provides considerable motivation to accelerate their research.

Provide Head-to-Head Comparisons - The competitions provide a unique opportunity to compare disparate approaches for realizing controllable microrobots that would not be possible by studying the technical literature alone. This has been particularly useful in evaluating the controllability and repeatability for different methods of locomotion.

Identify Measurement Needs - There are many measurements routinely performed on macroscale robots that cannot be performed on mobile microrobots because of their small size and high speed. NIST gains considerable insight into the shortcomings of existing measurement methods, which motivates the development of new measurement techniques that will aid in the adoption of this technology by U.S. industries.

NIST has organized the annual series of microrobotic performance competitions, beginning in 2007, in association with the RoboCup Federation. The competition events, while presented to fit the soccer theme of the RoboCup organization, are structured to test microrobotic systems in the key performance areas of mobility, maneuverability, and manipulation capability. Robots in these competitions are required to be no bigger than 300 micrometers in their largest dimension, and to have no wires or physical tethers extending outside of a 300 micrometer cube. Participating teams had to complete the following three tasks on a field of play based on a soccer pitch that measures 2 mm long and 2 mm wide and has a goal at each end:

The Two-Millimeter Dash
The microrobot must traverse a straight-line distance of two millimeters in as little time as possible, beginning from, and ending at, a complete stop within a defined area. Most obviously, this event measures the speed of a microrobot, but in practice the responsiveness of the robot to start and stop signals is often the critical capability.

The Slalom Drill
The microrobot must navigate the same two-millimeter course as before, this time avoiding a set of obstacles placed in its path. The number of obstacles is increased as necessary to force more complex paths and to differentiate higher levels of maneuverability.

The Shootout
The microrobot must maneuver through an obstacle course, collecting and delivering microscale silicon discs (soccer balls) to a goal location (see Fig. 1). This performance metric tests the ability of the robot to perform planar pushing manipulation tasks.

Participating teams have the option to attempt these tasks autonomously, using image feedback from a microscope and digital camera, or by teleoperation. However, since teleoperation is easier to implement, a penalty is assessed for this mode of operation. Figure 2 shows several of the microrobots that have been entered in the competition, each of which to date has been based exclusively on the electrostatic and electromagnetic locomotion methods described in the previous section.

So far, only one team has been able to complete The Two-Millimeter Dash and The Slalom Drill autonomously (ETH Zürich), while two teams have completed them by teleoperation. No team has completed The Shootout using the soccer ball shown in Fig. 1, but ETH Zürich has demonstrated goal-scoring with soccer balls developed specifically for their robot. These shortcomings point to the high level of difficulty of the tasks in this competition.

Over the three years that the NIST microrobotics competition program has been operating, a broad variety of microrobotic systems has been evaluated. Tested robots include those operating based on electrostatic attraction, soft magnetic resonant actuators, and hard magnetic actuators. Masses of the competing robots have ranged from 10s of nanograms up to 10 micrograms.Material combinations used for the microrobots have included silicon and chromium; metal thin films and thermoset polymers; nickel and gold; and rare-earth magnetic materials. The microfabrication protocols used to manufacture the evaluated microrobots have included surface micromachining processes, high-aspect-ratio electroplating, and laser micromachining. Despite this tremendous diversity of microrobotic technologies, ***all*** of the evaluated systems have converged on the same class of gaits.

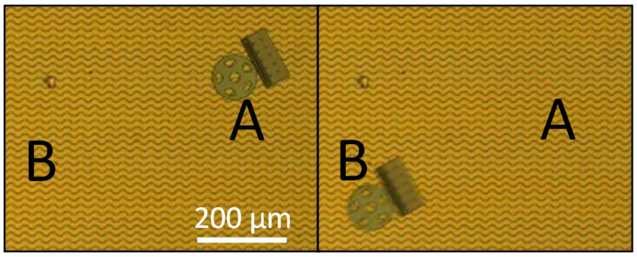


Figure 1 A sequence of two images showing an electrostatic microrobot moving a silicon disc (soccer ball) from point A to point B (elapsed time ≈ 3.5 s)

In contrast to the legged or wheeled modes of locomotion typical of macro-scale robotic systems, which seek to minimize friction, the class of gaits that has become most prevalent in microrobotics consists of a slip-stick motion in which frictional anisotropies are exploited. Typically, the microrobot slides against friction in one part of the motion cycle, then is held fast by friction forces during the recovery portion of the motion cycle.

The success of the slip-stick class of gaits leads to many more questions than answers about the future of microrobotic technologies. Friction at such small size scales typically exhibits non-Amontonian behavior, in which the friction force is not linearly proportional to the normal force. Non-Amontonian friction regimes remain poorly understood and can be difficult to model or predict. In addition, the normal contact forces can change by orders of magnitude in response to variations in the environment or operating surface and in response to wear and electrostatic charging of the contacting surfaces.

Reliability is a significant challenge for microrobotic systems, with performance variations from robot to robot of the same design and for individual robots over time. Operable lifetimes range from minutes to hours, and failure modes are poorly understood.

The operating mechanisms for microrobotic devices are understood primarily in abstract terms, so that optimization of microrobot performance is accomplished mostly on a trial and error basis. For example, robot motions corresponding to new resonant modes were discovered in the midst of the 2009 competition by changing the electrical input parameters.

More detailed models of microrobot operation are required, along with the experimental means to validate them. Validating models of operation is made difficult by the fact that discrete microrobot motions are often much smaller than the microrobots themselves and can be difficult to observe. For example, single steps of the scratch drive actuator are thought to be as small as 10 nm.

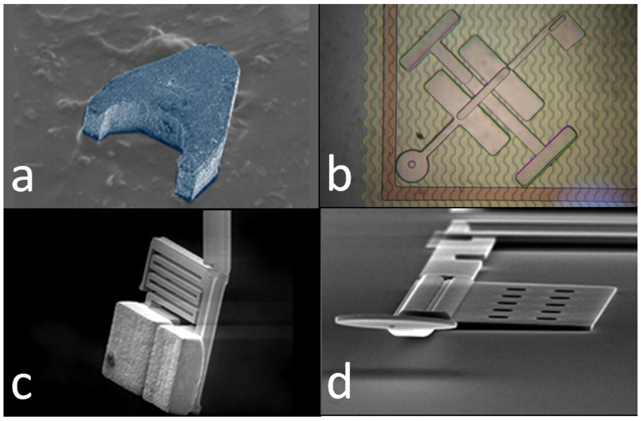


Figure 2 Various microrobots that have been demonstrated in previous competitions: a) hard magnet microrobot (Carnegie Mellon University), b) polymer-based electrostatic microrobot (Simon Fraser University), c) resonant electromagnetic microrobot (ETH Zürich), and d) electrostatic microrobot (U.S. Naval Academy).

4. MEASUREMENT NEEDS

The locomotion mechanics for most microrobots are not understood in detail due to complex force interactions at the micro- and nanoscales. As a result, precision measurement of the physical behavior of microrobots will play a significant role in modeling locomotion mechanics, developing new microrobot designs, and pushing their performance limits. The list below highlights the most pressing measurement needs, which has considerable overlap with those needed for current and prospective commercial MEMS devices. In many cases, suitable instrumentation and methods for these measurements are currently not available.

- *Coarse Motion*. Characterizing the motion of the device on the field of operation. The most successful mobile microrobots utilize a slip-stick gait. Tools are needed to understand the dynamics of this motion as well as the interaction between the microrobot and the surface as the microrobot makes nominally identical steps in a constant direction, as well as how the robot behaves as its direction of motion changes.
- *Fine Motion*. Characterizing the motion of subsystems of the microrobots at the nanoscale; in particular, the motion of the actuators that determine the direction and rate of movement.
- *Actuation Force and Stiffness*. Characterizing the output force of actuators and the stiffness of microrobot components will further understanding of modes of locomotion. Instrumentation that can measure multi-axis forces on the order of micronewtons with measurement bandwidth greater than 100 kHz must be developed.
- *Electromagnetic Properties*. Although the applied magnetic and/or electrostatic fields can be estimated for free space conditions, the presence of the microrobot and the region of operation means that the actual magnetic and/or electrostatic fields applied to the microrobot may differ substantially from the free space estimate. Therefore, tools are needed for local measurement of the magnetic flux density and capacitance.
- *Materials Properties*. Materials at the microscale are dominated by surface, rather than bulk, properties. Novel measurement methods to determine surface properties of microscale elements are needed to characterize the elements that compose a microrobot.
- *Friction and Adhesion*. At the microscale, friction and adhesion forces dwarf inertial forces due to the high surface area-to-mass ratio of microrobots. Methods for measuring non-Amontonian friction, adhesion forces (van der Waals forces, capillary forces, etc.), and quantum mechanical effects are needed.
- *Reliability*. The original promise of MEMS devices in the 1980s was that of small-scale machines incorporating gears and complex motion. This promise has yet to be realized primarily due to the poor reliability of MEMS devices with contact motions. Microrobots provide a platform for evaluating the reliability of a range of microscale contact modes and observing how their performance evolves over time.
- *Environmental Sensitivity*. For microrobots to meet many of the challenges elucidated previously, they must be able to function in a wide variety of environments (temperature, humidity, air, water, etc.). Performance metrics for microrobots to operate under varied environmental conditions will assist in overcoming existing requirements for tightly controlled operational environments.

5. PLANS FOR FUTURE COMPETITIONS

Although significant qualitative data has been captured in previous competitions, quantitative measurements have not been made while the microrobots performed competition tasks. As discussed in the previous section, new measurement methods and extensive data sets are critical for improved understanding of microrobot operation. The competition presents an excellent opportunity to measure the performance of a number of different technologies that would generally not be available in a single research laboratory. Therefore, we intend to incorporate microrobot metrology into the competition, which will be used to evaluate technologies and provide new insights into the mechanics of microrobots.

Unfortunately, many of the measurement technologies required to meet the needs listed in the previous section are complex, expensive, and not portable. However, as a first step in building performance characterization methods into the competition, a high-speed digital video system will be integrated into the competition microscope and an automated image processing application will be developed to provide coarse motion measurements with high motion bandwidth (> 500 Hz). The software will be capable of providing the planar coordinates (x, y, θ) of multiple microrobots and other objects (obstacles, manipulated parts) as a function of time, as well as other variables that can be extrapolated from this data. These include microrobot velocity, acceleration, trajectory tracking precision, turn radius, and motion repeatability, as well as an analysis of their kinematic constraints. Although it is expected that the image processing will be performed off-line for high-speed video, this tool will also be used for visual feedback when operating at slower frame rates (< 60 fps).

6. REFERENCES

[1] Sitti, M. 2007. Microscale and nanoscale robotic systems. *IEEE Robotics and Automation Magazine* 14 (1), 53-60.

[2] Abbott, J. J., Nagy, Z., Beyeler, F., and Nelson, B. J. 2007. Robotics in the small, part I: microrobotics. *IEEE Robotics and Automation Magazine* 14 (2), 92-103.

[3] Fearing, R.S. 1995. Survey of sticking effects for micro parts handling. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, 212-217.

[4] Donald, B. R., Levey, C. G., McGray, C. D., Rus, D., and Sinclair, M. 2003. Power delivery and locomotion of untethered microactuators. *Journal of Microelectromechanical Systems* 12 (2), 947-959.

[5] Donald, B. R., Levey, C. G., McGray, C. D., Paprotny, I., and Rus, D. 2003. An untethered, electrostatic, globally controllable MEMS micro-robot. *Journal of Microelectromechanical Systems* 15 (1), 1-15.

[6] Floyd, S., Pawashe, C., and Sitti, M. 2008. An untethered magnetically actuated micro-robot capable of motion on arbitrary surfaces. *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, 419-424.

[7] Vollmers, K, Frutiger, D. R., Kratochvil, B. E., and Nelson, B. J. 2008. Wireless resonant magnetic microactuator for untethered mobile microrobots. *Applied Physics Letters* 92, 144103.

[8] Sul, O. J., Falvo, M. R., Taylor, R. M. II, Washburn, S., and Superfine, R. 2006. Thermally actuated untethered impact-driven locomotive microdevices. *Applied Physics Letters* 89, 203512.

[9] Oldham, K., Rhee, Choong-Ho, Ryou, Jeong-Hoon, Polcawich, R., and Pulskamp, J. 2009. Lateral thin-film piezoelectric actuators for bio-inspired micro-robotic locomotion. *Proceedings of the ASME IDETC*, San Diego, CA, DETC2009-86427.

[10] Donald, B. R., Levey, C. G., and Paprotny, I. 2008. Planar microassembly by parallel actuation of MEMS microrobots. *Journal of Microelectromechanical Systems* 17 (4), 789-808.

[11] Pawashe, C., Floyd, S., and Sitti, M. 2009. Multiple magnetic microrobot control using electrostatic anchoring. *Applied Physics Letters* 94, 164108.

[12] Kratochvil, B. E., Frutiger, D., Vollmers, K., and Nelson, B. J. 2009. Visual servoing and characterization of resonant magnetic actuators for decoupled locomotion of multiple untethered mobile microrobots. *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, 1010-1015.

Is an Agent Theory of Mind (ToM) Valuable for Adaptive, Intelligent Systems?

Gary Berg-Cross

Knowledge Strategies

Potomac MD 20854

gbergcross@gmail.com

ABSTRACT

This paper serves as a short introduction for the special PerMIS session on Theories of Mind (ToM). The session intends to explore the viability of the ToM concept for R&D and if the ToM hypothesis is mature & relevant to the goal of highly competent systems able to achieve goals in a relatively autonomous way. The question can be considered from philosophical, research and robotic implementations as well as critiques central to the topic.The introduction is organized into 4 parts. Part 1 briefly reviews some of the history of the ToM idea and its recent reformulations. Part 2 discusses the widening use of the concept as an explanatory device within a few areas using developmental studies as a focus. Part 3 introduces the idea that particular types of robotics offer a new kind of tool to investigate cognitive development and the validity of some theories such as a ToM. The paper concludes with an outline of some issues that remain to be explored and advanced to show the value of a ToM theory in general and especially within the domain of intelligent systems.

Categories and Subject Descriptors

Primary Classification: I. Computing Methodologies
I.2 ARTIFICIAL INTELLIGENCE, **I.2.6** Learning

General Terms

Measurement, Experimentation, Design,

Keywords

Theory of Mind, Developmental Robotics, adaptation.

1. INTRODUCTION and History

A Theory of Mind (ToM) refers to reasoning about the mental states of self and others. Empathy, the concept of putting yourself in another person's shoes and relating to his situation, is a good example of theory of mind at work.

This idea fits our everyday understanding of others, or what is called a folk psychology explaining why some things happen in the world. The practice of folk psychology has been a recurrent topic in philosophical and psychological discussions for a long time. In philosophy conceptualizing intentional states such as beliefs and desires has often been seen as dependent upon our linguistic abilities. That is, language ability seems to provide a representational medium for describing our own and others people's actions in an intentional way. Recently, a new perspective on folk psychology has emerged in philosophy of mind and psychology. Such conceptualizations achieved a new purchase when Premack & Woodruff (1978) asked, ‘Does the chimpanzee have a theory of human intentions?’ or as they put it do they have a “theory of mind”. How would we know? The resulting research on this concept included its use as an explanation for autistic children's cognitive deficits building on evidence suggesting that autistic individuals lack an ability to maintain a theory of other minds and to reflect on their own thought processes (Baron-Cohen, 1995). Since this early work the concept of a ToM has come to be broadly used in a developmental perspective of how children come to understand the social and psychological world. Formalized as a ToM theory these propose alternative inherited or acquired paths by which a particular cognitive capacity may arise in a cognitive agent (children) so they understand and predict external behavior of others by attributing unobservable mental states, such as beliefs, desires and intentions**.** Such a theory fits the observation that one human can predict how another human will behave in familiar surroundings because they are maintaining a theory of other people's beliefs. Indeed everyday behavior seems largely based on what an adult person thinks others know, believe or want. But this idea might apply to highly social animals, such as chimpanzees and young children, in general since they need to compete and cooperate effectively with others in their family or group. It would be advantageous to not only to react to what others are doing but also to anticipate what they will do. A practical way of accomplishing this is to act like a junior scientist - observe what others do in particular situations and construct a set of

Conference'04, Month 1–2, 2004, City, State, Country.

‘behavioral rules’ that fit the pattern. The reward is that behavioral prediction in similar situation. But a more flexible way to anticipate others behaviors is posit what the goals of others might be and what state of affairs they are trying to bring about. This goes beyond behavioral prediction in similar situations but may also apply to novel situations.

Scientific evidence for a ToM comes from several sources with a particular favorite being the false belief task described by Wimmer and Perner (1983) which seemed to show that a full-fledged TOM doesn’t develop before the age of 3/4. They set up a series of experimental tests in order to check whether children between 3 and 5 years of age were able to attribute a false belief to someone else. In one experiment a child (Maxi), puts chocolate in a blue cupboard and leaves to play. Following this an adult uses part of the chocolate placing the remaining part into a different color (green) cupboard. Wimmer and Perner reported that when asked where ”Maxi” would look for the chocolate after returning most children under four years of age attributed to Maxi what they understood and had seen themselves, that the chocolate is now in the green cupboard. This suggests that a full-fledged TOM doesn’t develop before the age of 3/4. Younger children believe what it is the case, while older kids show a capacity of distinguishing what another mind might believe that is in reality false.

2. USES OF THE ToM CONCEPT

In the last 20 years ToM has been used to investigate many cognitive issues and is a broad explanatory device. For example studies of autistic children show a significant lower performance on false belief task compared to other cognitive tasks for testing intelligence and language capacities. This has lead some cognitive researchers of a nativist bent to see ToM as a mental mechanism and hypothesis that autism is the consequence of a specific deficit of the Theory of Mind Module (TOMM). A less daring view, called the Theory Theory, is to consider ToM as a naively constructed theory which is one of many conceptual "revolutions” that are achieved by cognitive agents. In this formulation concepts of mental states arise in a way similar to many other heuristics that we generate to deal with the world. Karmiloff-Smith’s (1992), for example proposed a series of such cognitive revolutions based in change of children’s internal representations. These internal changes are called representational redescription (RR). Initially external influences predominate and are represented by semsorimotor procedures. Knowledge as such is implicit and dominated by the concrete appearance of things such as seen in younger children in the false belief test. Over time a need to have this knowledge in a more useful form leads to re-coding it into progressively more explicit forms. Explicitness increases the flexibility and manipulability of the “knowledge” stored in the developing system. As knowledge and a child’s theories become more encapsulated internal modules are formed. This means that input and output processing become more stable and progressively and less accessible to and influenced by other modules and processing. Each higher form/level emerges from a redescription that re-encodes the prior lower level. For example perception of a zebra may be redescribed using the category of animal adding an attribute of stripes. In this redescription some of the perceptual detail and precision has been lost, but some connection to other things has been added and communication with others is enhanced because the child can be conscious of the concepts when procedures are not active. At each successive level, these representations become more explicit and hence more available to linguistic expression. Thus, development proceeds from implicit representations of basic behavioral procedures to successively more abstract, explicit, and flexible structures. This position is essentially consistent with a constructive epistemology, since one need not attribute any special, modular initial ability to children in order to explain how they know what they know. In this view language and a ToM is more like other domains of knowledge and may capitalize on innate perceptual biases and constraints.

Karmiloff-Smith’s model provided a speculative mechanism to explain representational within a constructionist approach emphasizing domain-specific constraints on development that are not realized by pre-fixed modules. Instead children develop their language skills as a repository of language-relevant representations that is tuned by focused/constrained experience within a language community. If this community uses mentalist terms to explain why things happen then this affords development of a ToM as a naïve model which can easily be used to communicate. Evidence for this is that around 24 months (Hirsh-Pasek and Golinkoff, 1996) we see a predominant reliance on a coalition of external cues diminishe as a grammatical system becomes more robust. This is also a time when a child grows more aware of the complex relationships among people and which could develop into a Theory of Mind. This combination is an opportunity to go from early language to later language skills with more abstract meanings communicated about past or future events and feelings. As proposed by Hirsh-Pasek and Golinkoff (1998), paraphrasing Bloom's (1993) Principle of Elaboration a child is driven to formulate (or discover) ways of communicating these ideas in a later phase using language as the preferred mode. A child’s development of a ToM would fit this general path of development, but of course that remains a hotly debated topic in cognitive development.

3. DEVELOPMENTAL ROTOBICS

If it proves viable a ToM could be relevant to the eventual goal of highly competent systems able to achieve goals in a relatively autonomous way. However, theories around a ToM mind need to be fleshed out and critiqued. The empirical base also needs to be expanded since the bulk of early research was a mix of child development studies along with evidence from brain function (e.g. viewing specific brain regions as a component of ToM) or clinical studies or autism, schizophrenia, Asperger and Williams syndrome and some on non-human primates. A broadening of discussion of all these is sponsored in the session and diverse disciplines that study intelligent systems including computer science, AI, cognitive science, psychology, behavioral and social science, and philosophy to provide improvements to our understanding of ToMs. A new way of investigating the theory comes from the field of Developmental/Epigenetic robotics. These combine developmental psychology and robotics in the effort to mimic cognitive development and show emergent behavior in these intelligent systems. Given a focus on emergent of behavior ToM represents an important topic in the field since it takes an accumulation of cognitive skills and knowledge as discussed by Brian Scassellati in his "Foundations for a Theory of Mind for a Humanoid Robot "(2001). Such work uses a robot with high-level cognitive skills coupled to the low-level perceptual abilities of a humanoid robot draws from a ToM and while interested in advancing and testing the theory is also interested in the practical question of how to build machines that interact naturally with people. This means building intelligent systems that can both interpret the behavior of others using social rules as well as display the social cues that will allow people to naturally interpret a system's behavior.

4. ISSUES REMAINING TO BE EXPLORED

Robotic tests of a ToM ask several fundamental questions which have only partly been addressed:

- o Can robots employ and exhibit a ToM like humans including beliefs?
- o How might a ToM `be represented in a robot implementation?
- o How can we use developmental robots to test how a ToM might be learned through their interactions with a surrounding environment?
- o What is the role of social involvement, language and understanding?
  - ▪ including: Human-robot and robot-robot interaction,
- o How important is embodiment to developing a ToM?

Only a small part of these have been explored and much needs to be considered. In additional other important topic areas can be directed at the ToM topic. These include:

- Critical philosophical analysis of the hypothesis that beliefs and desires that are the central mental states required to make sense of behavior and resulting questions about what heuristics/lower cognitive abilities are needed for a ToM to develop and to make a ToM computationally practical.
- Developmental issues to better understand the consistent path of ToM
  - o For example, the interactions and mix of higher-order, executive functions (e.g. self-regulatory cognitive processes, working memory and control of attention along with resistance to interference)
  - o Observational methods (e.g. false belief task, imitation) to test a ToM including work with children and non-human primates
- The relation of self reflection, joint attention, communication, imitation, or episodic memory to a ToM,
- Modular vs. explanatory theory formulations of a ToM.
- How important is ToM to levels of autonomy?
- What cognitive architectures can support a ToM?

Such understanding may support computational theory of mind as an emerging phenomena with the practical possibility that such intelligent machines could "help produce wealth i.e., goods and services that people want and need.".

5. REFERENCES

[1] Baron-Cohen, S. (1995). *Mindblindness: An essay on autism and theory of mind*. Cambridge, MA: The MIT Press.

[2]

[3] Hirsh-Pasek, K. & Golinkoff, R. M. (1996). The Origins of Grammar: Evidence from Early Language Comprehension. MIT Press; Cambridge, MA.

[4]

[5] Karmiloff-Smith, A. (1992). Beyond Modularity. Cambridge, MA: MIT Press.

[6]

[7] Premack, D. & Woodruff, G. (1978) Does the chimpanzee have a theory of mind?

[8] *Behavioral and Brain Sciences* 4:515–26.

[9]

[10] Scassellati, B. (2002) Foundations for a Theory of Mind for a Humanoid Robot, *Autonomous Robots*, *12*, 13-24.

[11]

[12] Wimmer, H. & Perner, J. (1983) Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition* 13:103–28.

Towards a simple Robotic Theory of Mind

Kyung-Joong Kim
Mechanical & Aerospace Engineering
Cornell University, Ithaca, NY 14853, USA
Department of Computer Engineering
Sejong University, Seoul 143-747, Korea
kimkj@sejong.ac.kr

Hod Lipson
Mechanical & Aerospace Engineering
Computing & Information Science
Cornell University
Ithaca, NY 14853, USA
hod.lipson@cornell.edu

ABSTRACT

Theory of mind (ToM) is a cognitive function in which an agent can infer another agent's internal state and intention based on their behaviors. Can robots realize ToM like humans? There are many issues to be tackled to address this challenging problem, such as the representation, discovery and exploitation of an actor's self models. In this paper we study how robots can represent other's self with artificial neural networks and an evolutionary learning mechanism. This framework was tested with simulated and physical robots and a novel prey-predator scenario was introduced to measure the performance of ToM learning. Experimental results showed that the proposed ToM approach can recover other's self models successfully.

Categories and Subject Descriptors

I.2.0 [Artificial Intelligence – General]: Cognitive Simulation

General Terms

Algorithms, Performance, Design, Reliability, Experimentation, and Verification

Keywords

Robotics, Evolutionary Computation, Estimation-Exploration Algorithm, Theory of Mind, Neural Network, Robot Test

1. INTRODUCTION

Theory of Mind (ToM) is a cognitive capability that allows us to understand another's internal states (intention, goal, and belief) and predict future behaviors of others [1]. From the observation of other's behavior, facial expression, and speech, we can infer the person's internal state (emotions, thought, decision making, and plans). It was known that this function is supported by widely distributed areas of human brain [2][3]. For Chimpanzees, they have ToM but it is a bit different with human's one [4].

ToM has gained great interest from an engineering society. Scassellati built "finding faces and eyes and distinguishing animate from inanimate stimuli" functions for humanoid robots [5]. Buchsbaum *et al*. developed an anthropomorphic animated mouse character that uses his own behavior repositories to interpret other's behavior [6]. Hegel *et al*. studied human's theory of minds for different shapes of robots [7]. Ono *et al*. used theory of mind mechanism to improve human's understanding on robot's intention [8].

Implementing ToM has great difficulty because it is a kind of reverse inference based on observation. Other's self model is hidden and it exists inside of objects. It is not possible to see the internal model directly and it is only indirectly observable. The only thing that we can observe is that the reaction of the object to the inputs from environments. The model with continuous input-output signals is more difficult to be discovered than discrete one.

In this paper, each robot has its own self and the problem of ToM is to discover other robot's self as close as possible. In case of human, the self is located inside of human brain and represented with biological neural networks. The problem of ToM for human is to build models inside of my brain that approximate the behaviors originated from other's internal self. Like other's original self, the inferred other's self is also represented as biological neural networks. The problem can be reformulated as finding another biological neural network that shows close behavior with the original one. Robot uses the similar mechanism to do the ToM.

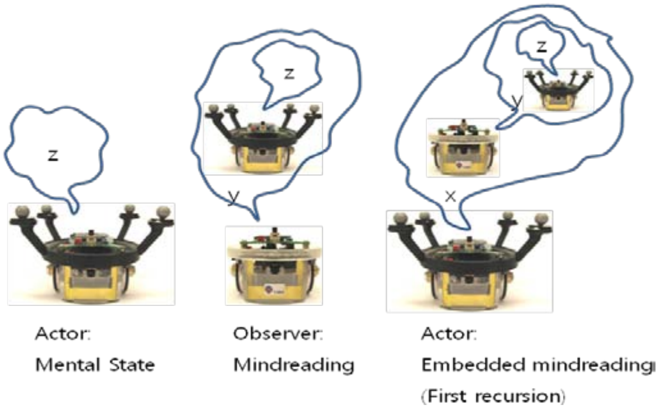
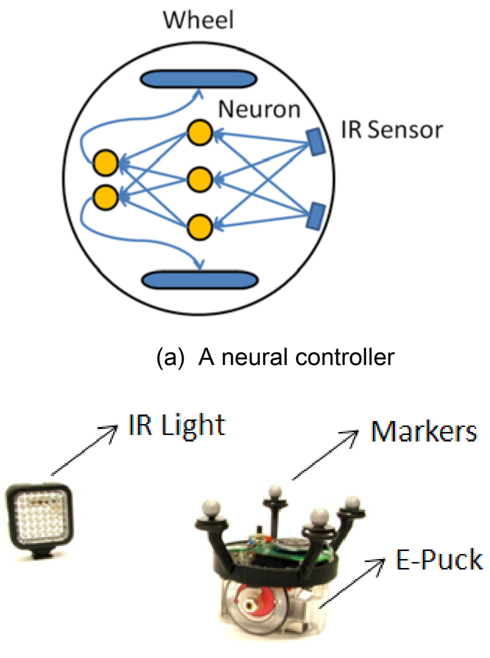


Figure 1. Theory of Mind in robots

In this paper, robots do theory of minds by inferring the neural networks inside other robots based on their movement (Figure 1). An artificial neural network controls the movement of robot's wheels based on sensory inputs. The inference is based on the exploration-estimation algorithm (EEA) used in reverse engineering of nonlinear-dynamical systems [9] and robot's self modeling [10]. After building other's self models, the robot exploits them to predict other robot's behavior. Figure 2 shows a neural network and virtual/physical robots used.



(a) A neural controller

(b) A real robot

Figure 2. An artificial neural network and robots

2. RELATED WORK

Premack and Woordruff asked “does the chimpanzee have a theory of mind?” in 1978 [1]. Heyes surveyed experimental evidences of non-human theory of mind in 1998 [11]. After 30 years research from the initial question, it was revealed that chimpanzee do have a theory of mind but do not understand others like humans do [12]. Series of experiments were conducted with chimpanzees to know what they know about others [13][14][15]. Two chimpanzees compete for food when only one of them has complete information about the location of food. They concluded that chimpanzees know what other can see and exploit it for food competition.

Childhood autism is related to the lack of theory of mind [16][17]. Baron-Cohen *et al*. compared normal, autistic, and Down’s syndrome subjects using a belief question to test theory of mind. The results for Down’s syndrome and normal subjects were similar (85% and 86% success ratio). On the other hand, 80% of autistic children failed the belief question. Ozonoff *et al*. tested the relationships between autism and first-order, second-order theory of mind [18].

Based on [19], theories for “theory of mind” are classified into four categories: Modular theory, simulation theory, theory-theory, and executive function theory. In modular approach, the theory of mind is functionally dissociable from other cognitive functions [17][20]. They assume that there are one or more neural structures specifically dedicated to theory of mind. In simulation theory [21], there is no general theory guiding the theory of mind. Instead, human’s brain mentally simulates other person’ s situation by placing himself to the other person’s place. This perspective-taking view of theory of mind does not support specialized, distinct neural structure for this cognitive skill. In theory theory view, child has a theory about how other minds operate and it evolves over time [22]. Some theorists argue that a distinct theory of mind does not exist and executive functions are sufficient for the cognitive skills [18].

Recently, there are new finding about theory of mind of humans. Herrmann *et al*. compared theory of mind ability among human, chimpanzee, and orangutan with gaze following and intention understanding tasks [23]. They concluded that human outperforms other species in theory of mind. Falck-Ytter *et al*. investigated proactive goal-directed eye movements in 12-month old and 6-month old infants using a specialized system for action perception [24]. They concluded that 12-month old infants do the proactive goal-directed eye movements and this is evidence on the action understanding of infants. False-belief test is a representative method to know whether infants have theory of mind. Onishi *et al*. proposed a novel nonverbal task to examine 15-month old infant’s ToM ability [25]. Rosenbaum *et al*. conducted theory of minds tests for someone with severe impairment of episodic memory and autonoetic consciousness [26]. They reported that there is no difference of the ToM ability between normal and impaired persons. Bloom’s research suggested that theory of mind is important to learn meanings of words [27].

There are works on verifying theories of “theory of mind” with neuroscience knowledge. Gallese *et al*. [28] related to the theory of “theory of mind” and the discovery of mirror neurons in human and monkey’s brain. They argued that the finding supports “simulation theory” but not “theory-theory.” Blakemore *et al*. supports the simulation theory based on the psychophysical and neurophysiological studies [29]. Ramnani *et al*. tested “simulation theory” by comparing human brain’s activation for preparing one’s own actions with one for predicting the future actions of others [30]. The conclusion was that both of them use action control system of the human brain but activate different action sub-systems. This result suggests that a simple form of simulation cannot be the only mechanism involved in ToM [31]. Siegal *et al*. reviewed recent findings on the relationships between brain regions and theory of mind [32]. Some functional components found were not solely dedicated to the theory of mind. However, domain-specific component (centered on the amygdale circuitry) was included in the region. This result supports modularity view. Saxe *et al*. related developmental psychology and functional neuroimaging research and supported the modular approach by arguing the existence of a specialized neural system for ToM [33].

Brain-imaging technology has been widely used to pinpoint region of brain for theory of mind [2]. Frith *et al*. used “story comprehension task” to invoke theory of mind and revealed several active regions (medial prefrontal cortex and posterior superior temporal sulcus) of human brain by ToM [34]. McCabe *et al*. reported that prefrontal cortex is highly activated to the cooperator in “trust and reciprocity” games for cash rewards against human [35]. Gallagher *et al*. reviewed several functional imaging works for theory of mind [36]. Krach *et al*. tested human’s ToM with human-robot game and the activation of brain regions related to ToM is related to the human-likeness (computer<functional robot<anthropomorphic robot<human) [37]. Hampton *et al*. investigated the activation of human brain using fMRI when they play simple two-player strategy game [38]. In their game, players use three different strategies (reinforcement

learning, fictitious play based on history of other players, and sophisticated ToM). They investigated brain activation regarding to the choice of the strategy.

The works that implemented ToM are categorized into two groups based on the level of implementations. Some of them focused on the demonstration only with simulation. A few demonstrated their works in real physical robots. The complexity increases when the work is realized in physical robots.

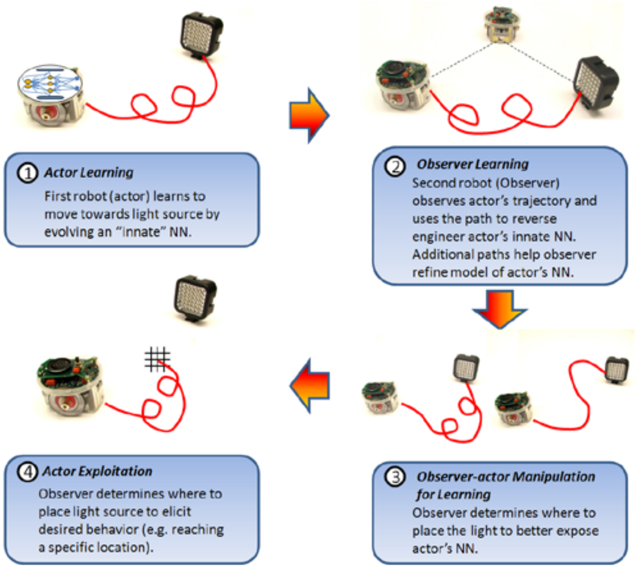
Christopher developed synthetic vision, memory, and theory of mind module for embodied conversational agents [39]. In his work, agent has three theories to do ToM: “Have they seen me”, “Have they seen me looking”, and “interest level.” Robinson *et al*. invented a mind-reading machine recognizes human’s mental states (discrete six states) from video input of human’s facial expression [40]. Breazeal *et al*. developed synthetic mouse characters that recognize other mouse’s behavior based on their own repositories [6]. Treur *et al*. proposed a two-level BDI (Belief, Desire and Intention) model for ToM [41]. The first level was used to model self’s BDI and the other was for reasoning about other agent. Marsella *et al*. developed a social simulation tool, PsychSim whose agents have beliefs about other agents [42]. Arita *et al*. [43] and Zanlungo [44] applied ToM to complex agent-based simulations and discussed about the effect of the level of ToM. Kondo *et al*. used the ToM in “carrying a stick task” for the cooperation of two computer programs [45]. Bringsjord *et al*. created a virtual character with a reasoning engine and they demonstrated that the character can pass the false-belief task by inserting “If someone sees something, they know it and if they don’t see it, they don’t” statement [46].

Kelley *et al*. developed a physical robot that uses own learned experience to detect the intentions of the humans [47]. The experience of robot was encoded into Hidden Markov Models. Breazeal *et al*. created animated robot LEONARDO that infers other person’s goals based on the simulation theory [48]. It passes a basic false-belief task. Scassellati developed ToM for a humanoid robot COG based on two representative ToM theories [5]. Yokoya *et al*. used a recurrent neural network to model the relationships between robot’s movement and actual object’s reaction [49]. After building its own model, it observes human’s behavior of rotating objects (blocks) and expanded the original self model to model human’s one. Demiris *et al*. followed “simulation theory of mind” and used robot’s own motor system to understand other robot’s behavior [50]. Takanashi *et al*. inferred other robot’s behavior based on its own behavior repositories in the game of robot soccer [51].

There are several works targeted to theory of mind. Kuniyoshi *et al*. developed several skills of simulated and embodied robots for theory of minds: “learning by watching,” and “imitation” [52]. Kozima *et al*. proposed a framework to implement and exploit theory of mind from indirect experience of infant humanoid robot [53]. Ono *et al*. assumed that human’s theory of mind model is organized as Baron-Cohen’s modular view and implemented an interface system to help humans understand robot’s intention [54]. Agents migrate from physical robots to user’s computer for shared attention. Ito *et al*. also focused on factors related to human’s ToM in the interaction of artifacts [55][56]. Scassellati *et al*. built a self model from the relationships between visual input and actual motor movement of robot and used it to discriminate others from self [57]. This is an important skill to do theory of mind. Kramer provided with an overview of the theory of mind in communication with virtual humans [58]. McCabe *et al*. introduced the concept of theory of mind to interpret the results of theoretical games played by humans [59]. They mentioned that the form of games is related to the human’s theory of mind execution and produce different outcomes. Boella *et al*. stressed the importance of theory of mind in the construction of social reality with multi-agent systems [60]. Akiwa *et al*. recognized that just imitating human’s behavior is not interesting to human demonstrator and proposed a system to predict subject’s next action based on past experience [61]. The prediction was done based on the difference between current behavior and past one. Flax modeled Leslie’s modular view on the theory of mind using first-order modal logic with an example of a scenario [62]. Hall *et al*. used theory of mind assessment of children to evaluate a virtual character system [63].

3. METHODS

In [64], authors tested ToM learning in simulated robots. In this real robot testing, simulation and a real robot was used together to do the ToM. In actor learning, simulation is used. In observer learning, the trajectories were collected from real robots and simulation was used in EEA. In actor exploitation, the position of new light source to seduce actor’s robot was determined with simulation and tested in real robots.



① Actor Learning
First robot (actor) learns to move towards light source by evolving an “innate” NN.
② Observer Learning
Second robot (Observer) observes actor’s trajectory and uses the path to reverse engineer actor’s innate NN. Additional paths help observer refine model of actor’s NN.
③ Observer-actor Manipulation for Learning
Observer determines where to place the light to better expose actor’s NN.
④ Actor Exploitation
Observer determines where to place light source to elicit desired behavior (e.g. reaching a specific location).

Figure 3. Overview of ToM learning

3.1 Actor Learning

In the first stage, the neural network controller is evolved for the actor robot. The architecture of neural network is fixed and only the weights are evolved. The sensory inputs (light level) are inputted to the neural network and the output is the movement of wheels. Figure 4 explains the details of the evolutionary algorithm used. Each controller is represented with a vector of weights and each entry has an associated self-adaptive parameter. The mutation operator updates the weights based on the self-adaptive parameter’s value. A task is to follow light source and a fitness function is defined based on the distance to the light source.

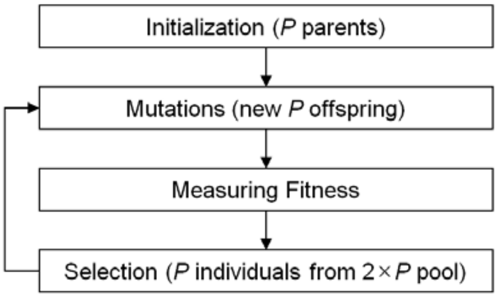

Initialization (*P* parents)
Mutations (new *P* offspring)
Measuring Fitness
Selection (*P* individuals from 2×*P* pool)

Figure 4. The evolutionary procedure to evolve actor robot's controller

3.2 Observer Learning

The goal of this stage is to discover actor robot's self (the neural network evolved) based on their real trajectories. It uses EEA (Estimation-Exploration Algorithm) to learn other's self models [9]. Initially, one trajectory is observed from the actor robot. In Estimation step, it runs learning other's self models multiple times with different random seed and produces multiple candidates (neural networks). In Exploration step, using the candidates, a number of starting points are tested and the EEA chooses the one with the maximum disagreement of the candidates as a next observing point. The next trajectory is observed from the new starting point chosen and the two trajectories are used for the next estimation step. A new population of the estimation step is initialized with the best candidates of the previous estimation step. Evolutionary algorithm is used to learn the other's self model in the estimation step. It is a kind of active incremental learning algorithm.

Figure 5 explains the fitness function in the evolutionary algorithm. The trajectory of the robot is a time-series sequence of the X-Y coordinates. At time t, the robot is placed in (X(t), Y(t)) in the environment and the next position is estimated by a candidate neural network. The fitness was calculated based on the difference between the original position and the estimated one.

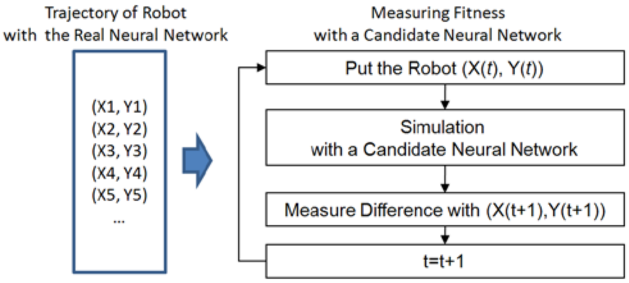

Trajectory of Robot
with the Real Neural Network
(X1, Y1)
(X2, Y2)
(X3, Y3)
(X4, Y4)
(X5, Y5)
...
Measuring Fitness
with a Candidate Neural Network
Put the Robot (X(*t*), Y(*t*))
Simulation
with a Candidate Neural Network
Measure Difference with (X(t+1),Y(t+1))
t=t+1

Figure 5. Fitness measuring of a candidate neural network

3.3 Actor Exploitation

Once actor's self models are discovered, they can be used to predict the robot's trajectory and observer robot can catch it with a trap based on the estimation. A trap is placed in the middle of the light source and a starting position of the other robot. With the actor's self models discovered, a new light position can be estimated to seduce the other robot to the trap. This is called "ToM estimation." The easiest way to predict the other robot's movement is "straight line estimation" assuming that the robot will go straightly to the light source. However, the movement of robot evolved is not straight line and shows several interesting patterns. The two approaches are compared to measure the goodness of our method.

4. EXPERIMENTAL RESULTS

The proposed method was tested in various settings from simulations to real physical robots. In a simulation side, PhysX (A simulator with physics engine) and EnKi (for E-Puck robot) are used. In a physical side, E-Puck robots are used to get results. The robot has two light sensors (left and right) and controls the robot by adjusting the wheels. In PhysX simulation, the neural network outputs are "the rotational angle" and "speed" of wheel. For E-Puck robot, the speed of left and right wheels is outputs of the network. In case of visible trap, the robot can detect the trap located, and left and right sensors digitize the strength of signals from the trap. Each neuron in a neural network has a bias parameter and the arc tangent function is used as a transfer function.

Based on the success of the virtual experiments [64], our experiments were expanded to the real physical robots. In our experiment, E-Puck mobile robots were used. It has two wheels and eight infra-red sensors. Like the virtual cases, only two sensors were used. As a light source, infra-red LED light was used. The trajectory of robot was recorded using Vicon motion capture system. Reflexive balls were attached to robot's custom-built mounting base and the Vicon system recognized the position and angles of the robot based on the balls detected. Our simulator was implemented based on EnKi simulator. In our simulator, a sensor model was built based on sampling data (129 positions ×15 different angles × 8 sensors). Additionally, wheel speed level was readjusted based on real sampled data.

The actor's neural network was evolved at each setting. Figure 6 shows trajectories of the evolved controllers at various starting positions. Their trajectories are not straight line and have a lot of curves. Also, they are very complex and have a lot of rotations to reach the goal position (light source). Although the controllers are evaluated at one starting position in the evolution, they can generalize well for different starting positions.

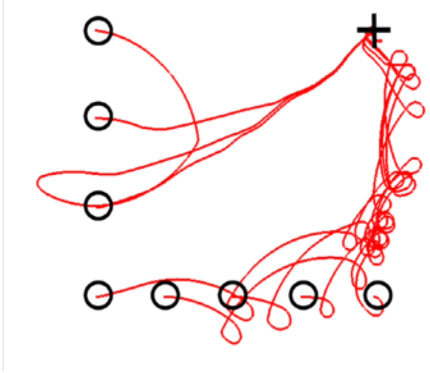


Figure 6. Trajectories of evolved neural controller (Black circle = Initial position, Black cross = Light)

Figure 7, Figure 8 and Figure 10 shows the progress of EEA learning. Figure 9 shows successful exploitation results for real physical robots. In EEA learning, real trajectories were collected from actor's robot. In the exploitation scenario, the new light position was estimated with simulation and tested with real robots. It shows that the reconstructed controllers can be used successfully to seduce the actor robot to the trap.

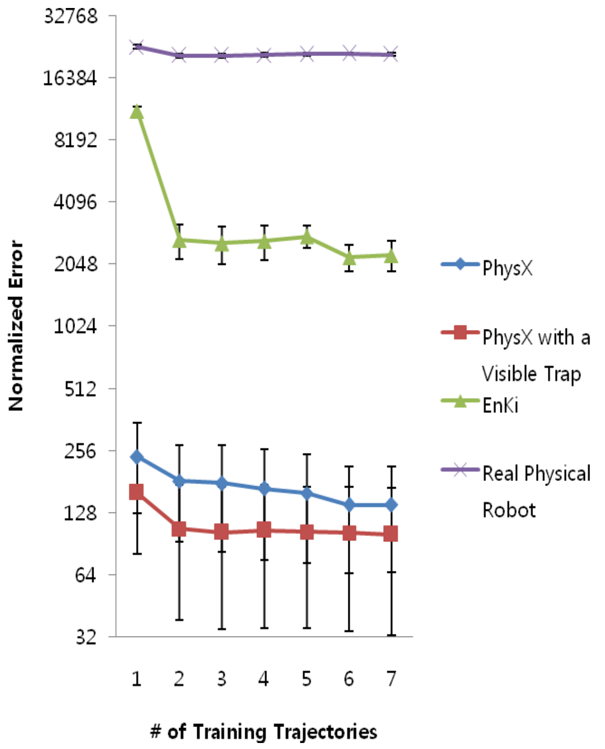


Figure 7. The progress of the observer learning in various environments

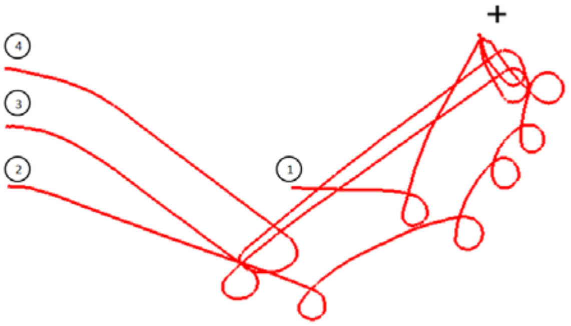


Figure 8. The trajectories actively chosen by the observer learning

Table 1 summarizes errors of all experimental environments. The ToM was compared with straight line estimation (assume that the robot will go straightly to the light source). For all cases, the ToM method can beat the straight line estimation method. In PhysX case, the ensemble of five candidate neural networks was successful and outperforms the single best neural network candidate and the straight line estimation. However, it is not true for the EnKi case and the ensemble method was not used for real robots.

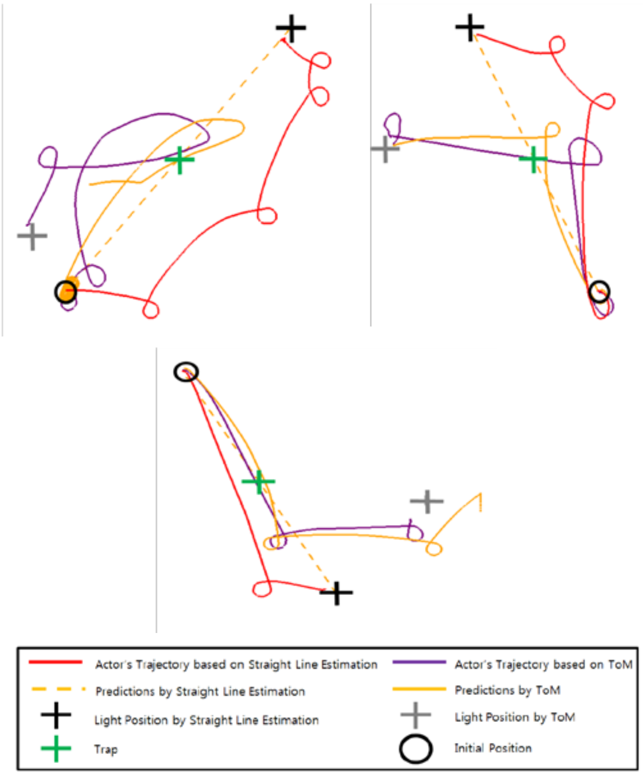

Actor's Trajectory based on Straight Line Estimation
Actor's Trajectory based on ToM
Predictions by Straight Line Estimation
Predictions by ToM
Light Position by Straight Line Estimation
Light Position by ToM
Trap
Initial Position

Figure 9. An example of exploitation for real robots.

Table 1. Statistical summary

	Straight Line Estimation	ToM (Single neural network)	ToM (Ensemble of 5 neural networks)
PhysX[1]	5.93 ± 0.54	3.87 ± 0.69	1.10 ± 0.28
PhysX with a Visible Trap[1]	18.21 ± 2.60	18.29 ± 2.72	11.97 ± 2.24
EnKi[1]	10.75 ± 1.26	0.89 ± 0.30	29.08 ± 5.75
Real Robots[1] **(Simulation)**	54.28 ± 2.84	35.37 ± 3.35	-
Real Robots[2]	34.80 ± 7.66	26.59 ± 9.33	-

1: Average of 100 points
2: Average of 10 points

5. CONCLUSIONS

In this paper, a variety of experiments were conducted to show the possibility of theory of mind implementation for robots. Each robot can model other robot's internal self model (neural network) based on their observation using EEA learning algorithm. Once the model was built, they can be used to predict other robot's future behavior. In these experiments, several virtual experiments and real physical robot testing successfully show the benefit of the other's self modeling.

In this paper, it is assumed that the neural structure of an actor robot is the same with the one of observer and there is no process to identify the fundamental structure. The number of the input-output neurons has to be analyzed to determine the structure of neural networks. After then, there are also many structural considerations: The number of layers, the number of hidden nodes for each layer, and the existence of recurrent links. The solution might be evolving topology and weights of neural networks simultaneously.

6. ACKNOWLEDGMENTS

This work has been supported in part by the U.S. National Science Foundation (NSF) Creative-IT program, grant #0757478. K.-J.K. was supported by the Korea Research Foundation Grant (KRF-2007-357-D00220) funded by Korean Government (MOEHRD) and Korea Health 21 R&D Project, Ministry for Health, Welfare and Family Affairs (A040163)

7. REFERENCES

[1] D. G. Premack, and G. Woodruff, “Does the chimpanzee have a theory of mind?,” Behavioral and Brain Sciences, vol. 1, pp. 515-526, 1978.

[2] C. Zimmer, “How the mind reads other minds,” Science, vol. 300, pp. 1079-1080, 16 May 2003.

[3] M. Siegal, and R. Varley, “Neural systems involved in ‘theory of mind’,” Nature Reviews-Neuroscience, vol. 3, pp. 463-471, June 2002.

[4] J. Call, and M. Tomasello, “Does the chimpanzee have a theory of mind? 30 years later,” Trends in Cognitive Sciences, vol. 12, no. 5, pp. 187-192, 2008.

[5] B. Scassellati, Foundations for a Theory of Mind for a Humanoid Robot, Ph.D. Thesis, Massachusetts Institute of Technology, 2001.

[6] D. Buchsbaum, B. Blumberg, C. Breazeal and A. N. Meltzoff, “A simulation-theory inspired social learning system for interactive characters,” IEEE International Workshop on Robots and Human Interactive Communication, pp. 85-90, 2005.

[7] F. Hegel, S. Krach, T. Kircher, B. Wrede, and G. Sagerer, “Theory of mind (ToM) on robots: A functional neuroimaging study,” Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, pp. 335-342, 2008.

[8] T. Ono, and M. Imai, “Reading a robot’s mind: A model of utterance understanding based on the theory of mind mechanism,” Proceedings of the 17th National Conference on Artificial Intelligence, pp. 142-148, 2000.

[9] J. Bongard, and H. Lipson, “Automated reverse engineering of nonlinear dynamical systems,” Proceedings of the National Academy of Science, vol. 104, no. 24, pp. 9943-9948, 2007.

[10] J. Bongard, V. Zykov, and H. Lipson, “Resilient machines through continuous self-modeling,” Science, vol. 314, no. 5802, pp. 1118-1121, 2006.

[11] C. M. Heyes, “Theory of mind in nonhuman primates,” Behavioral and Brain Sciences, vol. 21, pp. 101-148, 1998.

[12] J. Call and M. Tomasello, "Does the chimpanzee have a theory of mind? 30 years later," Trends in Cognitive Sciences, vol. 12, no. 5, pp. 187-192, 2008.

[13] B. Hare, J. Call, and M. Tomasello, “Do chimpanzees know what conspecifics know?,” Animal Behaviour, vol. 61, pp. 139-151, 2001.

[14] B. Hare, J. Call, B. Agnetta, and M. Tomasello, “Chimpanzees know what conspecifics do and do not see,” Animal Behaviour, vol. 59, pp. 771-785, 2000.

[15] M. Tomasello, J. Call and B. Hare, “Chimpanzees understand psychological states – The question is which ones and to what extent,” Trends in Cognitive Sciences, vol. 7, no. 4, pp. 153-156, 2003.

[16] S. Baron-Cohen, A. M. Leslie, and U. Frith, “Does the autistic child have a “theory of mind”?,” Cognition, vol. 21, pp. 37-46, 1985.

[17] S. Baron-Cohen, Mindblindness, MIT Press, 1997.

[18] S. Ozonoff, B. F. Pennington, and S. J. Rogers, “Executive function deficits in high-functioning autistic individuals: Relationship to theory of mind,” Journal of Child Psychology and Psychiatry, vol. 32, no. 7, pp. 1081-1105, 1991.

[19] G. L. Youmans, Theory of Mind in Individuals with Alzheimer-Type Dementia Profiles, Ph.D. Thesis of College of Communication at the Florida State University, 2004.

[20] A. M. Leslie, O. Friedman, and T. P. German, “Core mechanisms in ‘theory of mind’,” Trends in Cognitive Sciences, vol. 8, no. 12, pp. 528-533, 2004.

[21] R. Langdon, and M. Coltheart, “Visual perspective taking and schizotypy: Evidence for a simulation-based account of mentalizing in normal adults,” Cognition, vol. 82, 1-26, 2001.

[22] A. Gopnik and H. Wellman, “Why the child’s theory of mind really is a theory,” Mind and Language, vol. 7, pp. 145-171, 1995.

[23] E. Herrmann, J. Call, M. V. Hernandez-Lloreda, B Hare, and M. Tomasello, “Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis,” Science, vol. 317, pp. 1360-1366, 2007.

[24] T. Falck-Ytter, G. Gredeback and C. von Hofsten, “Infants predict other people’s action goals,” Nature Neuroscience, vol. 9, no. 7, pp. 878-879, 2006.

[25] K. H. Onishi, and R. Baillargeon, “Do 15-month-old infants understand false beliefs?,” Science, vol. 308, pp. 255-258, 2005.

[26] R. S. Rosenbaum, D. T. Stuss, B. Levine and E. Tulving, “Theory of mind is independent of episodic memory,” Science, vol. 318, p. 1257, 2007.

[27] P. Bloom, “Precis of how children learn the meanings of words,” Behavioral and Brain Sciences, vol. 24, no. 6, pp. 1095-1103, 2001.

[28] V. Gallese and A. Goldman, “Mirror neurons and the simulation theory of mind-reading,” Trends in Cognitive Sciences, vol. 2, no. 12, pp. 493-501, 1998.

[29] S.-J. Blakemore, and J. Decety, “From the perception of action to the understanding of intention,” Nature Reviews – Neuroscience, vol. 2, pp. 561-567, 2001.

[30] N. Ramnani, and R. C. Miall, “A system in the human brain for predicting the actions of others,” Nature Neuroscience, vol. 7, no. 1, pp. 85-90, 2004.

[31] N. Sebanz and C. Frith, “Beyond simulation? Neural mechanisms for predicting the actions of others,” Nature Neuroscience, vol. 7, no. 1, pp. 5-6, 2004.

[32] M. Siegal, and R. Varley, “Neural systems involved in ‘theory of mind’,” Nature Reviews-Neuroscience, vol. 3, pp. 463-471, June 2002.

[33] R. Saxe, S. Carey, and N. Kanwisher, “Understanding other minds: Linking developmental psychology and functional neuroimaging,” Annual Review of Psychology, vol. 55, pp. 87-124, 2004.

[34] C. D. Frith, and U. Frith, “Interacting minds-A biological basis,” Science, vol. 286, pp. 1692-1695, 1999.

[35] K. McCabe, D. Houser, L. Ryan, V. Smith, and T. Trouard, “A functional imaging study of cooperation in two-person reciprocal exchange,” Proceedings of the National Academy of Sciences, vol. 98, no. 20, pp. 11832-11835, 2001.

[36] H. L. Gallagher, and C. D. Frith, “Functional imaging of ‘theory of mind’,” Trends in Cognitive Sciences, vol. 7, no. 2, pp. 77-83, 2003.

[37] S. Krach, F. Hegel, B. Wrede, G. Sagerer, F. Binkofski, and T. Kircher, “Can machines think? Interaction and perspective taking with robots investigated via fMRI,” PLOS One, vol. 3, no. 7, e2597, 2008.

[38] A. N. Hampton, P. Bossaerts, and J. P. O’Doherty, “Neural correlates of mentalizing-related computations during strategic interactions in humans,” Proceedings of the National Academy of Sciences, vol. 105, no. 18, pp. 6741-6746, 2008.

[39] C. Peters, “A perceptually-based theory of mind for agent interaction initiation,” International Journal of Humanoid Robotics, vol. 3, no. 3, pp. 321-339, 2006.

[40] R. E. Kaliouby, and P. Robinson, “Mind reading machines: Automated inference of cognitive mental states from video,” IEEE International Conference on Systems, Man and Cybernetics, pp. 682-688, 2004.

[41] T. Bosse, Z. A. Memon, and J. Treur, “A two-level BDI-agent model for theory of mind and its use in social manipulation,” Proceedings of the Artificial and Ambient Intelligence Conference, pp. 335-342, 2007.

[42] D. V. Pynadath, and S. C. Marsella, “PsychSim: Theory of mind with decision-theoretic agents,” Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1181-1186, 2005.

[43] M. Takano, and T. Arita, “Asymmetry between even and odd levels of recursion in a theory of mind,” Proceedings of ALIFE X, pp. 405-411, 2006.

[44] F. Zanlungo, “A collision-avoiding mechanism based on a theory of mind,” Advances in Complex Systems, vol. 10, no. 2, pp. 363-371, 2007.

[45] K. Kondo, and I. Nishikawa, “The role that the internal model of the others plays in cooperative behavior,” Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, pp. 265-270, 2003.

[46] S. Bringsjord, A. Shilliday, D. Werner, M. Clark, E. Charpentier, and A. Bringsjord, “Toward logic-based cognitively robust synthetic characters in digital environments,” Proceedings of the First Artificial General Intelligence, pp. 87-98, 2008.

[47] R. Kelley, C. King, A. Tavakkoli, M. Nicolescu, M. Nicolescu, and G. Bebis, “An architecture for understanding intent using a novel hidden markov formulation,” International Journal of Humanoid Robotics, vol. 5, no. 2, pp. 1-22, 2008.

[48] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg, “Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots,” Artificial Life, vol. 11, pp. 31-62, 2005.

[49] R. Yokoya, T. Ogata, J. Tani, K. Komatani, and H. G. Okuno, “Discovery of other individuals by projecting a self-model through imitation,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1009-1014, 2007.

[50] Y. Demiris, and M. Johnson, “Distributed, predictive perception of actions: A biologically inspired robotics architecture for imitation and learning,” Connection Science, vol. 15, no. 4, pp. 231-243, 2003.

[51] T. Takanashi, T. Kawamata, M. Asada, and M. Negrello, “Emulation and behavior understanding through shared values,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3950-3955, 2007.

[52] Y. Kuniyoshi, Y. Yorozu, Y. Ohmura, K. Terada, T. Otani, A. Nagakubo, and T. Yamamoto, “From humanoid embodiment to theory of mind,” Lecture Notes in Artificial Intelligence, vol. 3139, pp. 202-218, 2004.

[53] H. Kozima, and J. Zlatev, “An epigenetic approach to human-robot communication,” Proceedings of the 2000 IEEE International Workshop on Robot and Human Interactive Communication, pp. 346-351, 2000.

[54] T. Ono, and M. Imai, “Reading a robot’s mind: A model of utterance understanding based on the theory of mind mechanism,” Proceedings of the 17th National Conference on Artificial Intelligence, pp. 142-148, 2000.

[55] K. Terada, T. Shamoto, H. Mei, and A. Ito, “Reactive movements of non-humanoid robots cause intention attribution in humans,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3715-3720, 2007.

[56] K. Terada, T. Shamoto, and A. Ito, “Utilizing theory of mind on human agent interaction,” IEEE International Symposium on Robot and Human Interactive Communication, pp. 757-762, 2006.

[57] P. Michel, K. Gold, and B. Scassellati, “Motion-based robotic self-recognition,” Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2763-2768, 2003.

[58] N. C. Kramer, “Theory of mind as a theoretical prerequisite to model communication with virtual humans,” Lecture Notes in Artificial Intelligence, vol. 4930, pp. 222-240, 2008.

[59] K. A. McCabe, V. L. Smith, and M. Lepore, “Intentionality detection and “mindreading”: Why does game form matter?,” Proceedings of the National Academy of Sciences, vol. 97, no. 8, pp. 4404-4409, 2000.

[60] G. Boella, L. van der Torre, “From the theory of mind to the construction of social reality,” Proceedings of CogSci, pp. 298-303, 2005.

[61] Y. Akiwa, Y. Suga, T. Ogata, and S. Sugano, “Imitation based human-robot interaction-roles of joint attention and motion prediction,” Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication, pp. 283-288, 2004.

[62] L. Flax, “Logical modeling of Leslie’s theory of mind,” Proceedings of 5th IEEE International Conference on Cognitive Informatics, pp. 25-30, 2006.

[63] L. Hall, S. Woods, R. Aylett, and A. Paiva, “Using theory of mind methods to investigate empathic engagement with synthetic characters,” International Journal of Humanoid Robotics, vol. 3, no. 3, pp. 351-370, 2006.

[64] K.-J. Kim and H. Lipson, “Towards a “theory of mind” in simulated robots,” *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation Conference*, pp. 2071-2076, 2009.

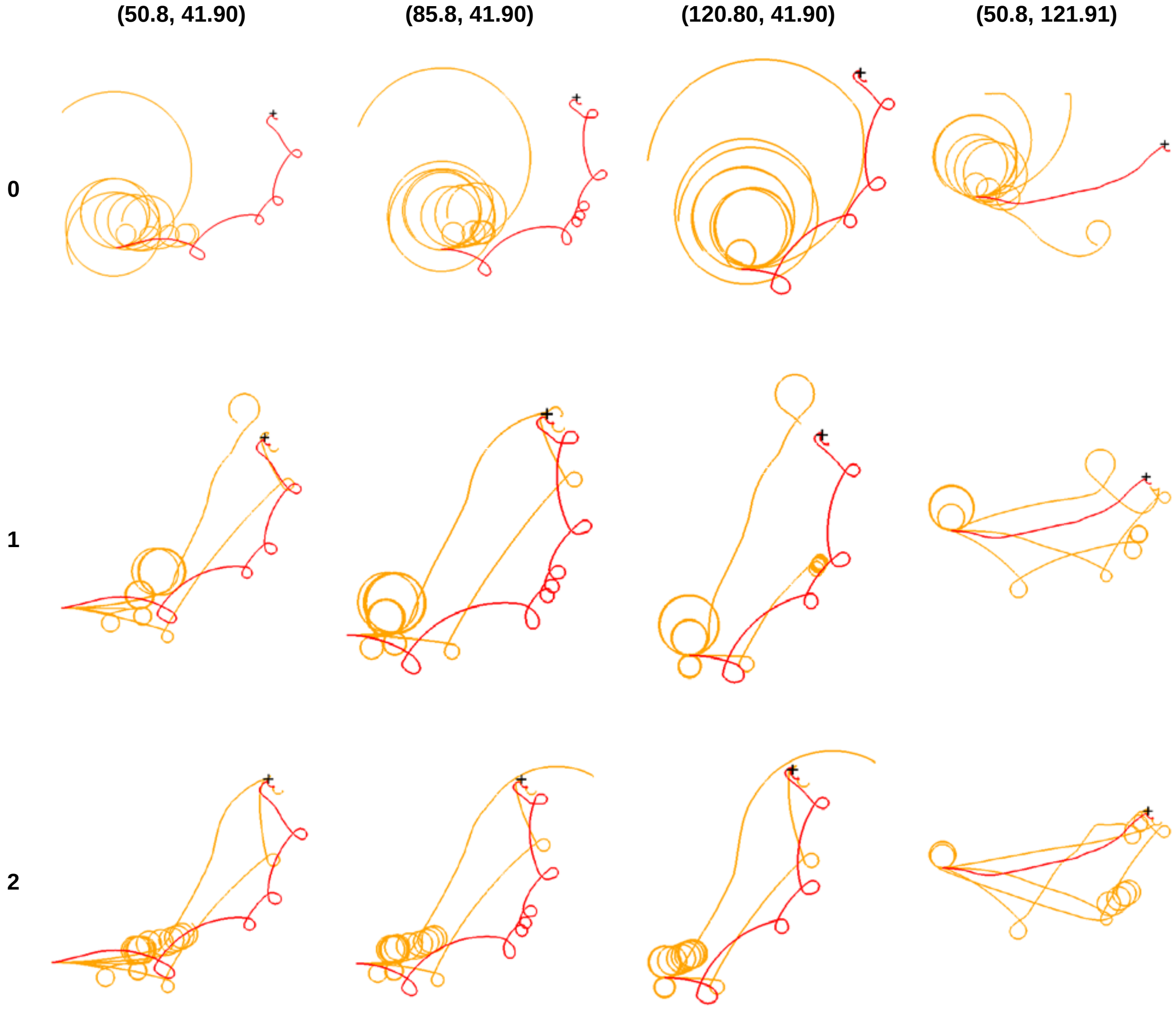


Figure 10. Progress of observer learning for real robots (Red line = Real Trajectory, Yellow line = Predicted trajectory)

Resilient Behavior through Controller Self-Diagnosis, Adaptation and Recovery

Juan Cristobal Zagal
Computational Synthesis Laboratory
Mechanical & Aerospace Engineering
Cornell University
Ithaca, NY 14853, USA
jcz35@cornell.edu

Hod Lipson
Computational Synthesis Laboratory
Mechanical & Aerospace Engineering
Cornell University
Ithaca, NY 14853, USA
hod.lipson@cornell.edu

ABSTRACT

We explore robot behavior recovery through a process akin to self-reflection. A robot contains two controllers: A primary "innate" reactive controller, and a secondary "reflective" controller that can observe, model and control the primary controller. The reflective controller adapts the innate controller without access to the innate controller's internal state or architecture. Instead, the reflective controller models the innate controller and then synthesizes input/output filters that adapt the innate controller's existing capabilities to new situations. The innate controller is subjected to a variety of sensory, motor, and internal control damage scenarios. The reflective controller diagnoses the level of failure using a self-model and the observed sensorimotor time-series data and is able to recover performance.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.9 **[Artificial Intelligence]:** Robotics---*Autonomous vehicles*

General Terms

Algorithms

Keywords

Self-reflection, damage recovery, machine learning, evolutionary robotics, self-modeling.

1. INTRODUCTION

Living systems are able to maintain their performance while compensating for changes in their environment and in their own structure [16]. Here we explore processes that allow machines to model their own behavior and use those models to detect controller failure and identify paths to recovery. Such ability is known to exist in the fish and reptiles [6][21] and it has been recently discovered in some structures of the adult mammalian brain [1][9][10][7] at the scale of changing and reorganizing neurons and their function by experience.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.

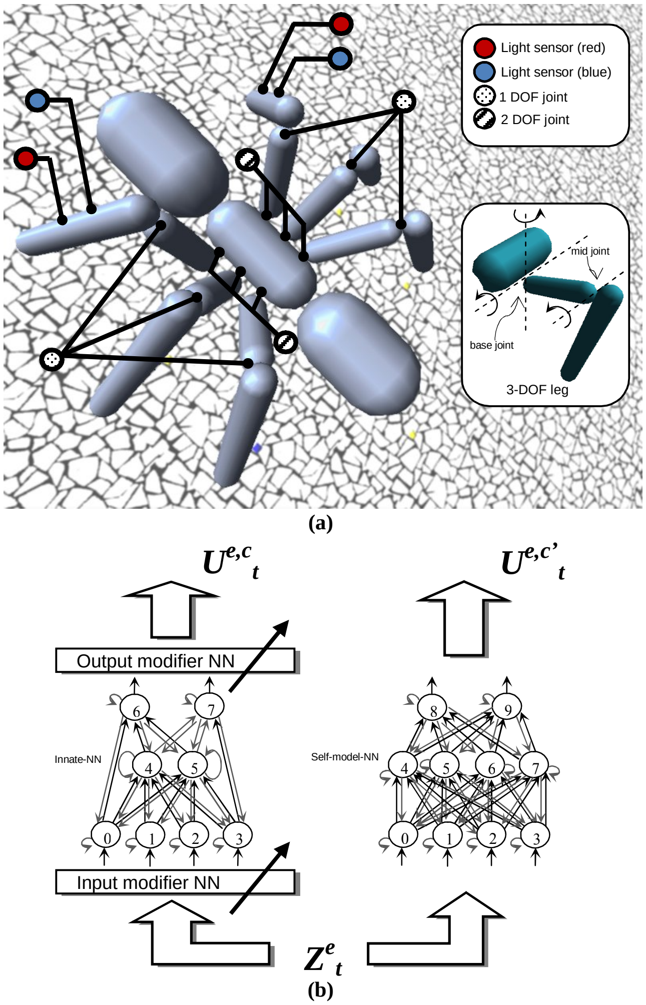
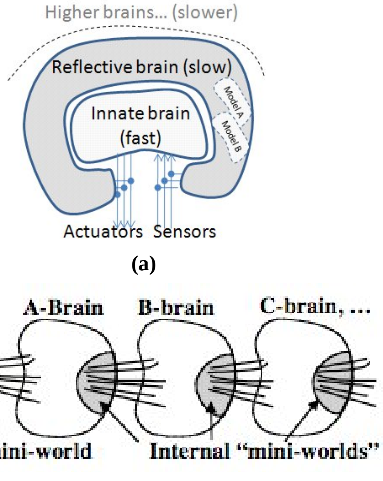


Figure 1. (a) Six-legged robot used for the experiments. The robot perceives light intensity by means of color specific sensors (red and blue) located at either side of its front legs. Each leg is independently controlled by three motors, two for the 2-DOF base joint and one for the 1-DOF mid joint. The robot is driven by 18 motors. A total of 15 rigid bodies complete the robot architecture. (b) The function recovery is achieved by the synthesis of modifier networks. A self-model (right) of the innate controller (left) is used to synthesize these modifiers.

In recent studies, we showed that a robot's resiliency increases when self-modeling its own morphology [3][4][18]. We explored how self-models of a robot body can be used for damage recovery [2]. Here we wish to take this concept a step further, by having a robot model its own *controller* as well. Just as a robot benefits from modeling its own morphology and then using that model to determine how to best compensate for a new situation, can a robot benefit from modeling its own *controller*, then use it to compensate for a new situations?

The first question to answer is why a robot would need to model its own controller at all, instead of directly accessing and manipulating it. The reasons for this are many fold: First, there are many aspects of a control system that cannot be explicitly described, even if its architecture is perfectly known: Sensor and actuation lag time, noise, and computational errors and delays, for example. Second, the controller may change in unanticipated ways due to failure or change in the environment. Manual modeling of an existing controller also takes time and effort, and direct manipulation of a controller could require an unwarranted increase in software and hardware complexity. Finally, in some cases the controller of a robot is simply inaccessible – either locked by design, or obfuscated by legacy code. The ability to modify performance of an existing controller without directly accessing it also serves as a safe adaption strategy, since the original controller is never modified and therefore its behavior can be restored at any time. This process may also shed light on the evolution of more opaque controllers such as biological nervous systems.



(a)

(b)

Figure 2. (a) A proposal of nested brains architecture. (b) Minsky's brain chain from [12].

The approach we use here is based on the assumption that there are two co-resident controllers; one reflecting on the other (Figure 2a). This architecture is a form of *metacognition*: the ability to reflect upon one's own mental processes and to self-regulate them. Such metacognitive processes are recognized to be present in humans, non-human primates, and a few other mammals [8][15]. It has been recently demonstrated to exist in the rat as well [5], suggesting that metacognition concepts might be applicable to simpler systems such as robots.

Minsky [11] points out that a brain can be better understood as a "society of minds" interacting with each other. He proposed a thought experiment consisting on dividing an artificial brain in two parts. While the input-outputs of the first part (A-brain) are connected to the external world, the second part (B-brain) is only connected to the A-brain; thus A is the only world seen by B (Figure 2b). As proposed by Minsky, the B-brain might help to the A-brain even without having access to the real world, and by just looking at the activity of the A-brain. Simple questions such as Are you repeating? Are you feeling better? And How do you think? might help to produce a better brain state in the world.

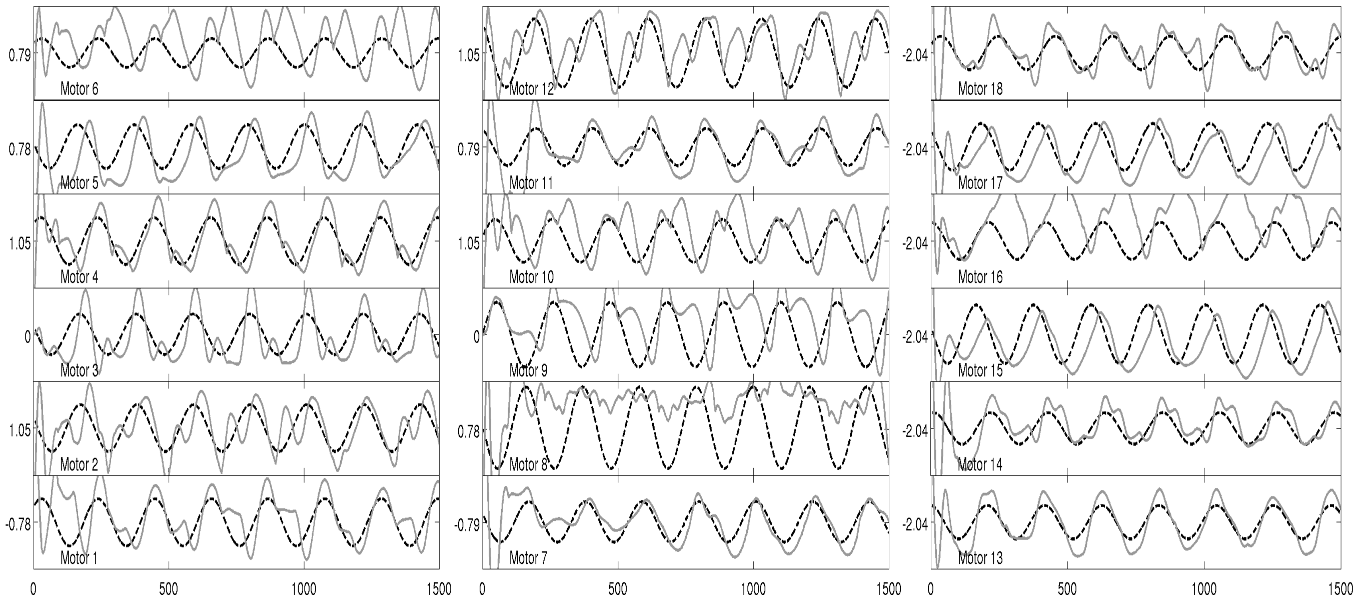
As pointed out by Minsky in recent work [13] there must be some brains that critique the performance of other brains or sets of brains. They might also be able to identify certain ways of thinking and to reconfigure thinking states by activating or reconnecting certain brain areas. If we are about to explore how to implement such a system using current robots the first question that arises is *how a critic system might identify and manipulate a certain way of thinking?* We hypothesize that minds should be able to perform some sort of self-modeling of other minds as a way to compare and reason about patterns of activation.

The remainder of this paper is organized as follows (Figure 8 presents a description of the entire process): In section 2 we describe the simulated robot and experimental environment used for our study. We also describe the generation of an innate robot behavior. In section 3, we present experiments showing the first stage of self-reflection by reverse engineering of the robot innate controller. In section 4, we describe different types of controller damage used for the experiments. In section 5 we describe how the self-model is used for damage diagnosis. In section 6 we describe the process of damage recovery through the synthesis of input/output modifiers. Finally in section 7 we present the conclusions of this work.

2. ROBOT AND ENVIRONMENT

We used a simulated six-legged robot for our experiments. The robot is free to walk on its environment, comprising a plane surface covered with moving emitters of red and blue light (see Figure 4); these light sources are initially randomly distributed along the surface and follow different patterns of motion. The robot architecture is illustrated in Figure 1a. The central portion of the robot is composed by three solid cylinders. The central cylinder is connected to six legs by means of 2-DOF motors. The forward walking behavior is the result of the rhythmic oscillation of the limbs. Each motor $i=\{1,\ldots,18\}$ follows a reference signal r^i = $\theta_i + a_i sin(\omega t + \phi_i)$,where θ_i is a pre-defined central angle of oscillation for motor ***i***. PID dynamic compensators are in charge of ensuring successful reference following for each motor. These pattern generators were obtained by evolving the amplitude a_i and relative phase ϕ_i of each oscillator such that the robot maximizes frontal displacement. A similar strategy was reported in [17]. Figure 3 shows comparisons of the reference signals given by the oscillators versus the actual angle which is achieved by each motor during a normal walking of the robot along an observation period of duration $T=1500$ time steps.

Two pairs of color-specific light sensors are located at the front legs of the robot, generating the measurement signals z^0 (blue) and z^1 (red) from sensors located at the left leg, and signals z^2 (red) and z^3 (blue) from sensors located at the right leg. At each simulation step, the read-out of a light sensor z^k is computed as the instantaneous light intensity at the sensor due to contributions of all the environment light sources l_i of corresponding color using and inverse-square law.



(a) **(b)** **(c)**

Figure 3. Angular reference signals for each motor r^i : $i = \{1,\dots,18\}$ (dash line) versus the instantaneous angle achieved while the robot walks (gray solid line). The central angle of oscillation is presented in radians at the left of each plot. In (a,b) the reference following for motors attached to the robot torso are illustrated. In (c) the reference following for the femur-tibia joints are shown.

Once the robot is provided with the forward-moving behavior, it learns to follows the moving sources of blue light while avoiding the sources of red light. This behavior was obtained by evolving the weights of an "innate" recurrent neural network (RNN) controller (Innate-NN) shown on Figure 1b (left). Four input neurons {0, 1, 2, 3} are fed by the four light-measurement signals. The network contains two hidden nodes {4, 5} and two output nodes {6, 7} that generate the left u_0 and right u_1 motor modulation signals. The signal u_0 modulates the amplitude of the left legs oscillators {1,2,3,7,8,9,13,14,15} and u_1 the amplitude the remaining right leg oscillators {4,5,6,10,11,12,16,17,18}. The output y_k of neuron k is computed as

$$y_k = \phi\left(\sum_j w_{kj} x_j - \theta_k\right) \qquad (1)$$

where $\phi(\cdot)$ is the sigmoid activation function, x_j are the input signals, w_{kj} are the connection weights and θ_k is the threshold of neuron k. The controller is represented by a genome c of $N_c = 34$ scalar parameters (in the range [−1, 1]): 26 connection weights, and 8 activation thresholds. The reward perceived by a robot is defined in equation (2) by assigning a positive (negative) reward to the amount of blue (red) light intensity that is collected during the evaluation of controller c under environment e after a period of T time steps.

$$F_t^{e,c} = \int_0^t \left(z_t^{0,e,c} - z_t^{1,e,c} + z_t^{2,e,c} - z_t^{3,e,c} \right) dt \qquad (2)$$

To avoid exploiting the peculiarities of a unique environment, we used a set of $N_e = 3$ randomly generated environments and we defined the fitness of a candidate controller c as follows:

$$F^c = \prod_{e=1}^{N_E} f_T^{e,c} \qquad (3)$$

Due to perceptual aliasing (where different locations trigger the same sensor state, albeit requiring different control actions) an optimal motor action $U_t = \{u_t^0, u_t^1\}$ cannot be purely determined by the sensor state $Z_t = \{z_t^0,\dots, z_t^3\}$ at a single time t; however, we hypothesize that a causal controller (such as a RNN) might have good overall performance over the evaluation period T.

The algorithm runs with a probability of mutation $p_m = 1/N_c$ using Cauchy mutation and a probability of crossover $p_c = 0.9$. The population size was set to 30 individuals per generation. Figure 4 shows the innate behavior that result after about 1000 evaluations.

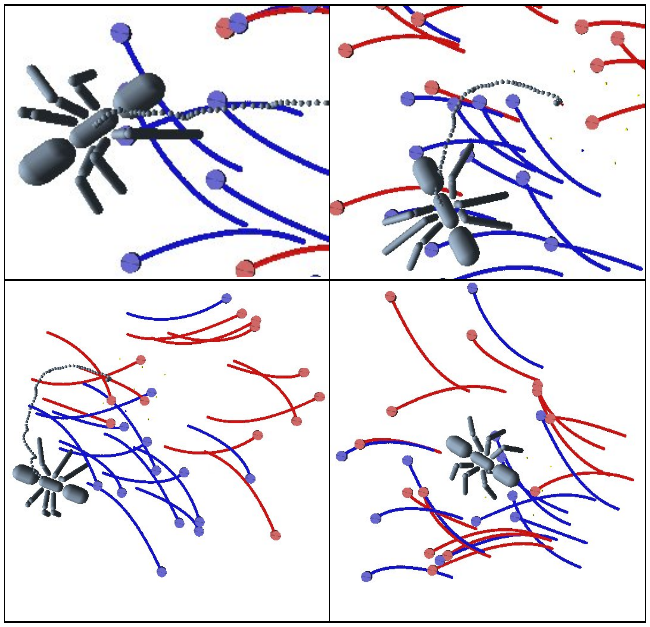


Figure 4. Illustration of the innate behavior. The robot moves toward areas of higher intensity of blue light while avoiding red lights (the discs are emitters of colored light). The figure shows four different scenarios. Motion traces are represented with the same color of the source.

3. SELF-MODELING CONTROLLER

In this section we describe the process by which the reflective controller acquires a self-model of the innate controller. Going back to Minsky's formulation, this is equivalent to the B-brain asking the A-brain about its way of thinking. The self-model is represented as a recurrent neural network controller (Self-model-NN) that has the same number of input and output nodes as the Innate-NN, though not necessarily the same number of hidden units. Figure 1 (right) presents the architecture of the self-model that we use for our experiments (contains 4 hidden nodes). It is represented with a genome *c'* of $N_{c'} = 62$ scalar parameters (in the range [−1, 1]): 52 connection weights and 10 activation thresholds.

We also considered the special case of a self-model (Self-model-NN$_T$) that has the same architecture of the Innate-NN. This is for the sake of testing some hypotheses described in the next section. We note corresponding test genome as c_T'.

We allowed the robot to operate freely under an environment *e* while executing its Innate-NN controller and we recorded vector time series of sensor data $Z^e = \{Z_t^e : t \in T\}$ and motor data $U^e = \{U_t^e : t \in T\}$. We fed the inputs of each candidate Self-model-NN controller *c'* with the recorded time series of sensor data Z^e, resulting in predicted motor actuation data $U^{c',e} = \{U_t^{c',e} : t \in T\}$. We then measured the quality of each candidate self-model controller *c'* by its ability to reproduce the same input-output patterns as those observed during the operation of the Innate-NN controller in different environments. To find the best self-model, we minimized the signal distance *D(c)* described by equation (4).

$$D(c) = \sum_e \int_0^T \left\| U_t^{c,e} - U_t^e \right\| dt \qquad (4)$$

Figure 5 shows results from optimizing the self-model, using a genetic algorithm, with the settings described in previous section. The search is steered toward minimizing the distance *D(c)* over the space of candidate self-model controllers. The figure corresponds to an average of eight runs of the minimization procedure.

The standard error is depicted with vertical error bars. In order to avoid overfitting, a 30% of the data was used for validation. The minimization stops when the error in the validation data starts to increase (early stopping). The convergence of candidate self-models is illustrated on Figure 7. The figure shows how the similarity of 3D trajectories increases when minimizing the distance *D(c)* over the training environment (a) and over different test environments (b,c). As it can be seen from the figure, resulting self-models can predict the robot performance under unseen environments to a large extend.

We note as c'^* the optimal self-model solution obtained during this stage.

4. CONTROLLER DAMAGE

In this section we describe the different scenarios of damage introduced to the innate controller for testing the here explored recovery method. Figure 6 shows the Innate-NN under six different types of controller damage.

The scenarios are summarized on Table 1. They consists on disconnecting different inputs to the network, disconnecting synaptic links inside the network, swapping synaptic links and also introducing constant perturbations on specific neurons.

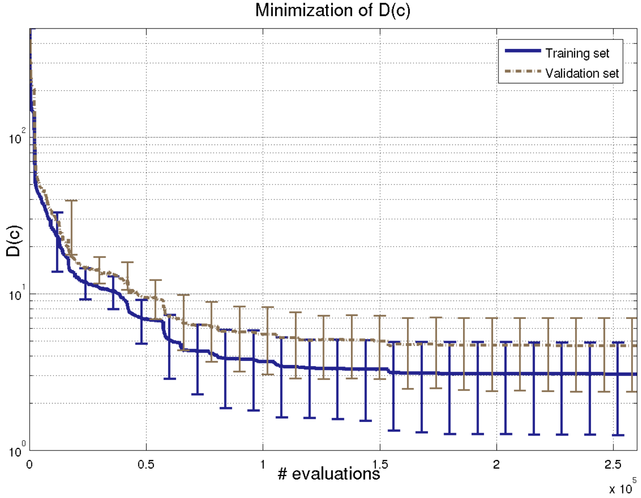


Figure 5. Results of minimizing the distance *D(c)* using genetic search over the space of self-models of the robot controller. Blue continuous line corresponds to the minimum distance achieved at corresponding evaluation. Burlywood line shows results obtained over validation data set. Standard error is depicted with error bars.

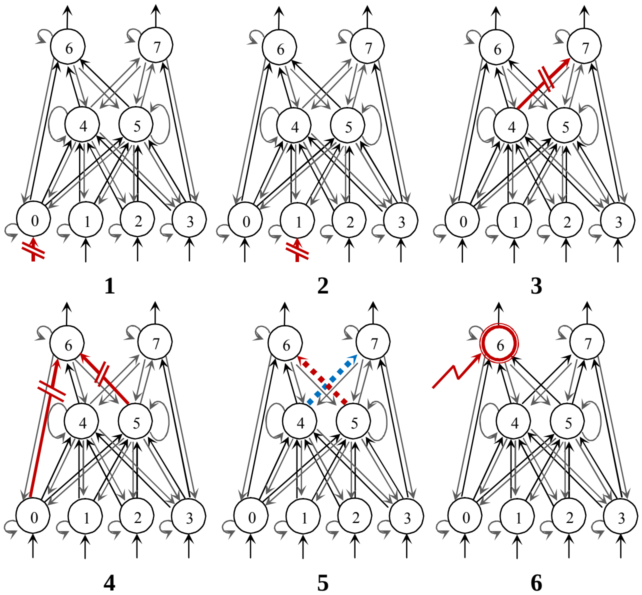


Figure 6. Different damage scenarios introduced to the Innate-NN controller for testing. Red or blue colors are used to indicate the location of the change introduced to the network. Link disconnections are presented in {1,2,3,4}, an exchange of neuron connections is introduced in {5} and a constant perturbation is introduced in {6}.

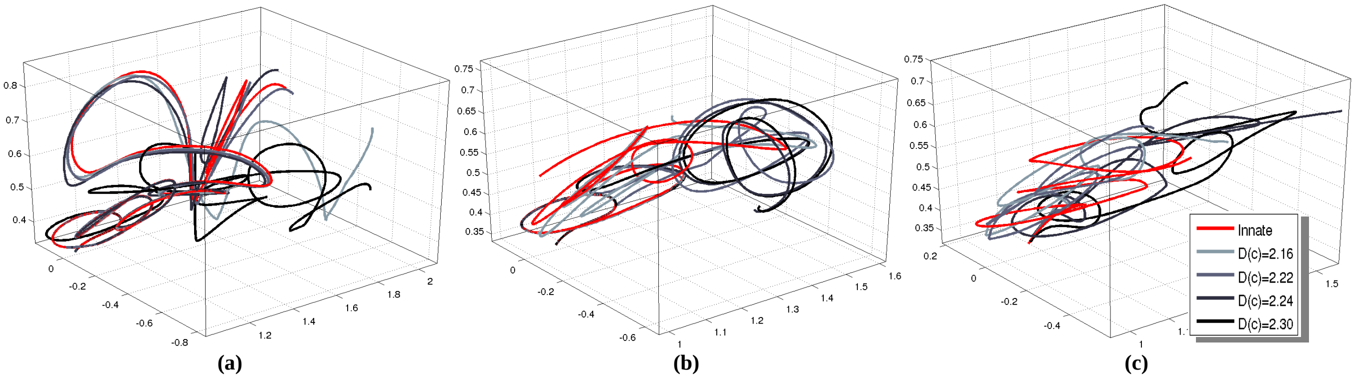


Figure 7. Convergence of robot trajectories induced by different candidate self-models (varying shades of blue). As it can be seen, a minimization of *D(c)* increases the similarity of resulting robot trajectories versus the innate target (red). Trajectory curves were obtained from environments of varying initial configuration of lights. Curves shown in (a) correspond to the environment used as training, (b) and (c) correspond to environments of increasing variation from the innate environment. As figures (b) and (c) show, resulting self-models are able to predict the robot behavior under unseen situations to a great extend. The axis scaling was adjusted for the data to fit in a cube.

Table 1. Damage Scenarios Tested

Scenario	Description
1	The left blue sensor input is damaged (input to node 0 is disconnected).
2	The left red sensor input is damaged (input to node 1 is disconnected).
3	The link from neuron 4 to neuron 7 is disconnected (weight changed to take the value of 0).
4	Two links entering node 6 are disconnected (weights changed to take the value of 0).
5	The weights connecting neuron 5 with neuron 6 and neuron 4 with neuron 7 are swapped.
6	A constant perturbation is introduced to left motor output (neuron 6).
7	No change

5. FAILURE DIAGNOSYS

In this section we describe how the self-model constructed by the reflective controller can be used to diagnose the damage introduced to the innate controller. As previously described, the self-model was derived from data collected during normal operation of the robot.

First we introduce one of the failures described on Table 1 to the innate controller and we then let the robot to operate on its environment while executing its now damaged innate controller. During this period we collected time series of sensor $Z^e = \{Z_t^e : t \in T\}$ and motor $U^e = \{U_t^e : t \in T\}$ data.

It is natural to expect at this point a difference between the sensorimotor relationships observed under failure versus those already explained by the self-model. We also expect a reduction in reward received from the environment.

Since the self-model was proven to provide good behavioral explanations of the innate sensorimotor relationships it is expected that small deviations from that solution might have some explanatory power of the failure.

Moreover we hypothesize that the topological distribution of these variations might have some degree of correlation with the actual source of the perturbation itself; we will further analyze these hypotheses in the remainder of this section.

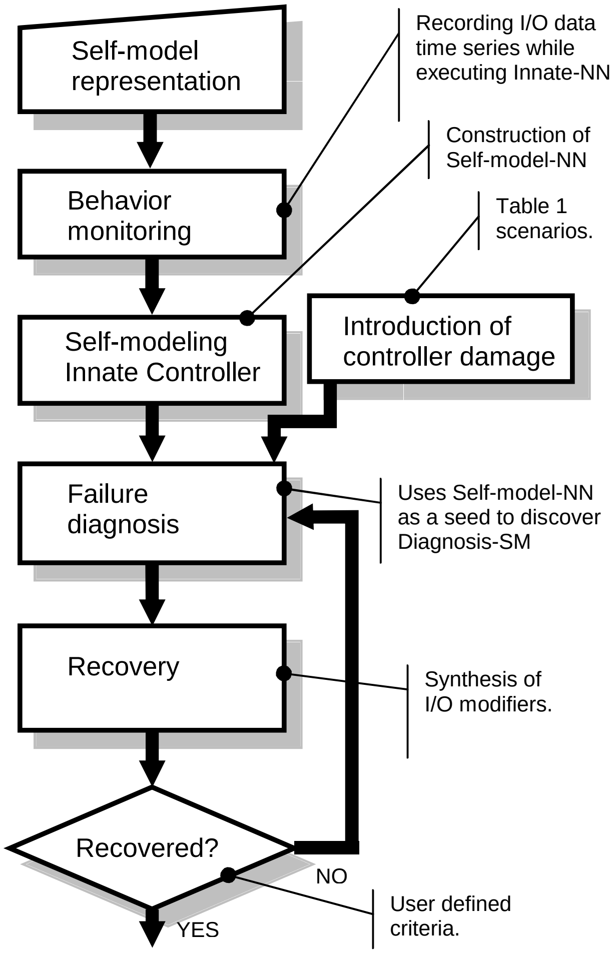

Self-model representation
Recording I/O data time series while executing Innate-NN
Behavior monitoring
Construction of Self-model-NN
Table 1 scenarios.
Self-modeling Innate Controller
Introduction of controller damage
Failure diagnosis
Uses Self-model-NN as a seed to discover Diagnosis-SM
Recovery
Synthesis of I/O modifiers.
Recovered?
NO
YES
User defined criteria.

Figure 8. Flow diagram of the method to obtain a robot resilient behavior through controller self-diagnosis adaptation and recovery.

We used the following strategy for producing new diagnosis self-models (Diagnosis-SM): Given the sensorimotor signals obtained during the perturbed functioning of the robot, we started a new genetic search over the same self-model search space that was described in section 3. A first population of diagnosis self-models was strongly seeded with the optimal self-model (c'^{*}). In this case the search was steered toward minimizing the difference between the Diagnosis-SM outputs versus the outputs observed during the operation of the damaged innate controller, this was achieved by feeding each candidate Diagnosis-SM with the observed sensor time series Z^{e}, and by using the distance described in equation (4). If there were no damage at all, the solution would be the seed itself (validated by our experiments). In presence of damage, however, genetic search will steer the Diagnosis-SM network to produce similar motor patterns as those provided by the damaged Innate-NN. We trained the Diagnosis-SM using the same stopping criterion as described in section 3. We note the resulting optimal Diagnosis-SM as c''^{*}.

Before analyzing our diagnosis results we should note some hypothesis:

H1: *The solutions for Diagnosis-SM, c''^{*} and for Self-Model, c'^{*} have a similar explanatory power over the innate system c.*

Thus, any explanatory property assigned by an observer to the Self-model must also hold for the Diagnosis-SM.

H2: *The self-model synaptic difference vector Δc = $|c''^{*}$- $c'^{*}|$ represent topological changes of the self-model network, but does not necessarily have any counterpart in the real system.*

This is since the topology of the self-model is different from the topology of the innate controller (with exemption of the examples presented here to test H3).

H3: *The topological changes represented by Δc might better approximate changes of a target system whose architecture is similar to the topology of the self-models.*

A first estimation of the level of failure is given by quantitative measurements of parameter variation under each failure scenario. Figure 9 shows the standard deviation of the synaptic difference vector Δc that results when comparing the Self-Model with the Diagnosis-SM on each test scenario. The figure shows two cases. The first is when Δc is defined in a space of self-models of generic architecture (Δc = $|c''^{*}$- $c'^{*}|$). The second is when Δc is defined in a space of self-models whose architecture intentionally match the innate architecture (Δc = $|c_T''^{*}$- $c_T'^{*}|$).

It is interesting to observe (see Figure 6 as reference) that when disconnecting a blue sensor at the left side (failure 1) the level of compensation is greater than disconnecting a red sensor (failure 2). It appears that swapping the neuron connections also induces a small level of compensation (failure 5). The remaining failures {3, 4, 6} appear to induce a similar level of compensation as those required by failure 1.

Another type of diagnosis deals with the topological localization of the synaptic compensations that are hypothesized from the resulting difference vector Δc. We remark however, that this entails the careful consideration of the abovementioned hypotheses.

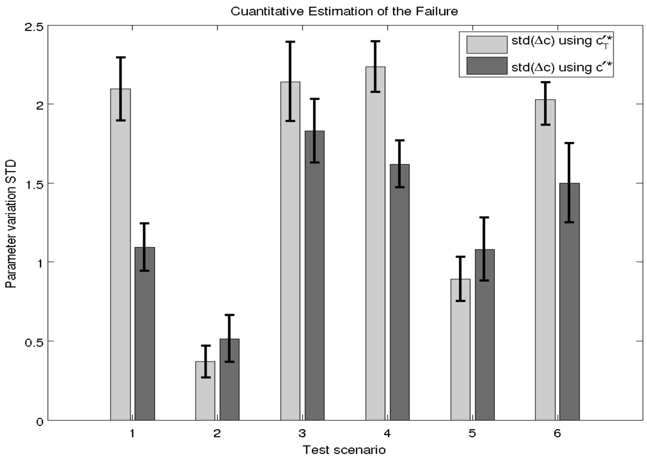


Figure 9. Quantitative estimation of the level of damage introduced under each test scenario to the innate neural network. The estimation is a result of comparing the original synaptic weights of the Self-model with the new synaptic weights resulting from evolving a Diagnosis-SM. Each bar represents the standard deviation of the weight difference vector *Δc*. Results are presented for the generic self-model architecture as well as for a test architecture.

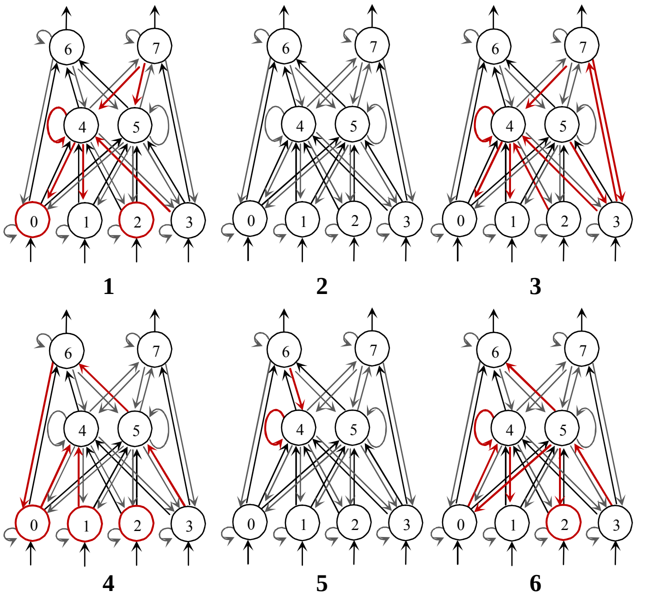


Figure 10. Most relevant compensatory modifications resulting from comparing Diagnosis-SM with the Self-Model for each one of the damage test scenarios. Arrows or nodes in red represent parameters on the Diagnosis-SM deviating more than the typical standard deviation (2.0) from the Self-model. In this case the Self-Model-NN$_T$ architecture was used.

Figure 10 shows the localization of the synaptic weights that are experiencing a larger degree of change on each scenario. In the case of failure 1, it is interesting to observe that disconnecting a blue sensor induces a change on the bias of the same sensor as well as on the bias of the counter sensor of the opposite side. There is also a change on the synaptic weight of the counter motor modulator.

Disconnecting a forward connection between neuron 4 and neuron 7 produces a larger degree of compensation over most part of the network (failure 3). Disconnecting two forward connections produces a strong compensation on the network as well, see failure 4.

Figure 11 shows bar plots comparing the original Self-model synaptic weights (grey) versus the weights of resulting Diagnosis-SM (black) on each test scenario.

6. ADAPTATION AND RECOVERY

In this section we study how to recover function by filtering the inputs and/or outputs of the innate controller which is being subject of different types of damage. The filters are implemented using RNN's as described in [20]. Figure 1b shows a diagram of this procedure when applying the filters to the Innate-NN during robot functioning. The idea is to recover the innate function by synthesizing input/output modifier networks using the Diagnosis-SM as a model of the damaged innate controller to be recovered, and the Self-Model-NN that was generated in 3, as a model of the target system to be recovered.

These self-models are exploited by re-injecting sensor signals that were recorded during the robot operation on a real environment. The procedure allows evaluating the quality of modifier filters without further accessing the damaged innate controller. More details of this process are presented in [19][20].

Figure 12 shows reward levels resulting from the operation of the robot on two different environments. The different damage scenarios are presented. In each case there is a bar representing the reward level obtained before (black) damage, after damage (grey) and after recovery (white). As it can be seen, the procedure allows recovering performance to a great extend in most of the failure scenarios under analysis.

7. CONCLUSIONS

We have illustrated how the resiliency of a simulated robot increases using a self-reflection process that involves controller self-diagnosis, adaptation and recovery.

The proposed approach might also be useful for reusing existing hardware for new tasks, by applying the presented monitoring and controlling stages. The algorithm could be implemented, for example, inside a pre-existing robot, and modify its behavior by modulating its original controller's input and output signals.

In a broader sense, we have presented a case where system identification techniques can be used to infer the parameters of a controller, instead of the parameters of the dynamical system under control. While adaptive control aims to dynamically compensate for a plant whose parameters are uncertain, we address here the problem (and envision possible advantages) of adding uncertainty to the controller itself.

The proposed technique for reverse engineering a controller (section 3) can be of interest beyond robotics. For example, a common problem faced by growing production plants is related to the task of inferring the actual inner workings of their legacy control components.

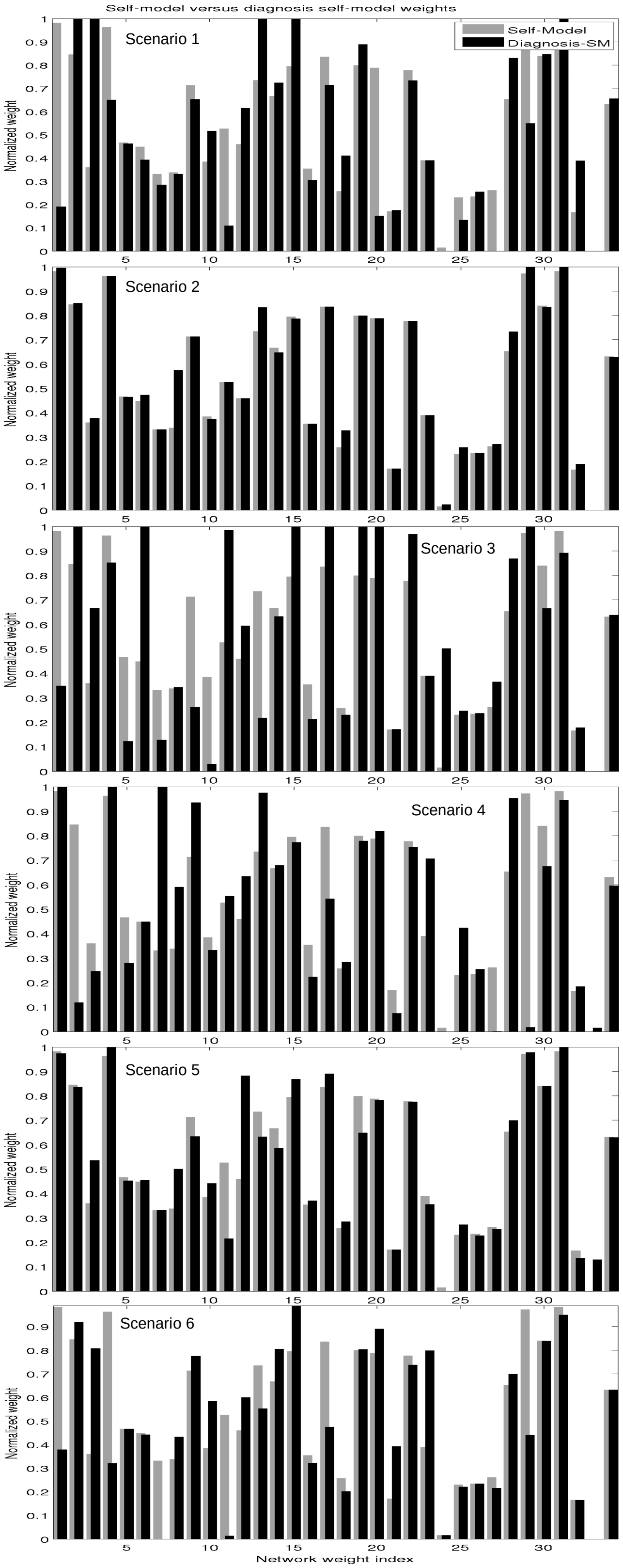
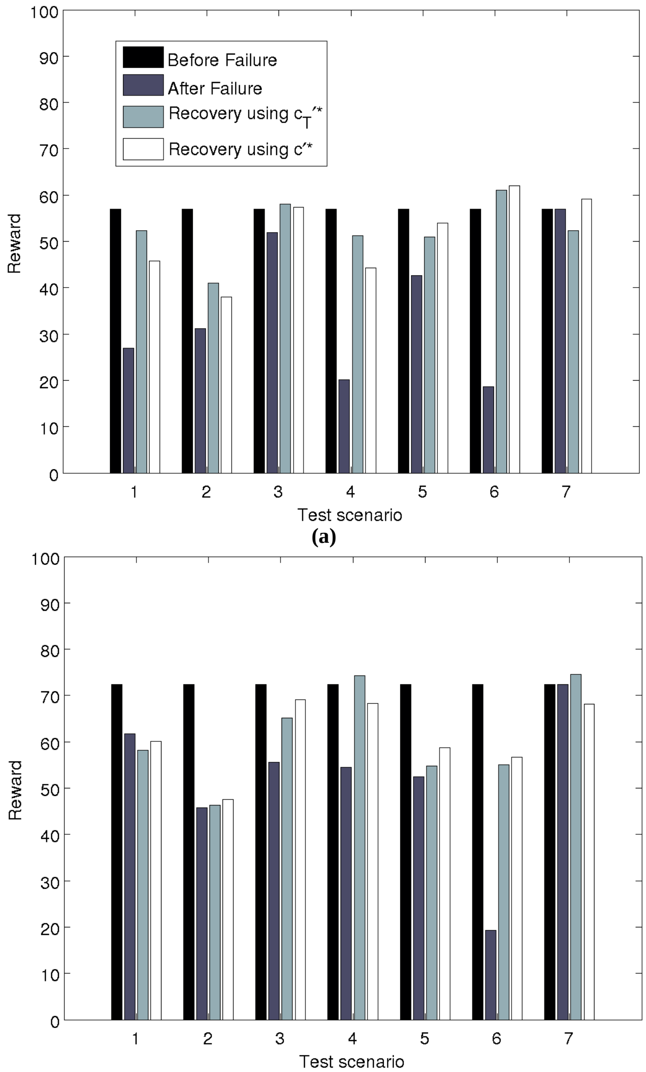


Figure 11. Comparing synaptic weights of Self-model (grey) versus resulting Diagnosis-SM (black) on each test scenario.



(a)

(b)

Figure 12. Reward levels obtained by the robot before failure, after failure and after recovery. Recovery results were obtained for the case of using self-models of ideal test architecture (solution $c_T'^*$) and the generic architecture (solution c'^*). Resulting reward levels are shown for two different environments (a,b) and considering the different failure scenarios. The simulation was set to be deterministic, having the same initial conditions under each scenario.

Although such knowledge is usually available for third party contractors it may be lost or too expensive. In theory, a meta cognitive system can be superimposed on an existing system, adding the possibility to monitor, control and further expand the capabilities of an existing system.

8. AKNOWLEDGEMENTS

This work has been supported in part by the U.S. National Science Foundation (NSF) Creative-IT program, grant #0757478 and Chilean research FONDECYT project number #3080048.

9. REFERENCES

[1] Bjorklund, A., Lindvall, O. 2000. Self-repair in the brain. Nature, 405, 892.

[2] Bongard, J., Lipson, H. 2004. Automated damage diagnosis and recovery for remote robotics. In IEEE International Conference on Robotics and Automation. Proceedings. ICRA '04. 3545-3550.

[3] Bongard, J., Lipson, H. 2004. Once more unto the breach: Co-evolving a robot and its simulator. In Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, 57–62.

[4] Bongard, J., Zykov, V., Lipson, H. 2006. Resilient Machines Through Continuous Self-Modeling. Science, 314(5802):1118–1121.

[5] Foote, A., Crystal, J. 2007. Metacognition in the Rat. Current Biology, 17(6):551–555.

[6] Font, E., Desfilis, E., Pérez-Cañellas, M. M., García-Verdugo, J. M. 2001. Neurogenesis and neuronal regeneration in the adult reptilian brain. Brain Behav Evol, 58, 276-295.

[7] Draganski, B., Gaser, C., Busch, V., Schuierer, G., Bogdahn, U., May, A., et al. 2004. Neuroplasticity: changes in grey matter induced by training. Nature, 427(6972), 311-2.

[8] Hampton, R. 2001. Rhesus monkeys know when they remember. Proceedings of the National Academy of Sciences, 98(9):5359.

[9] Horner, P. J., Gage, F. H. 2000. Regenerating the damaged central nervous system. Nature, 407(6807), 963-70.

[10] Kempermann, G., van Praag, H., Gage, F. H. 2000. Activity-dependent regulation of neuronal plasticity and self repair. Progress in brain research, 127, 35-48.

[11] Minsky, M. 1986. The society of mind. Simon & Schuster, Inc. New York, NY, USA.

[12] Minsky, M. 2005. Interior grounding, reflection, and self-consciousness. In: Proceedings of International Conference on Brain, Mind and Society. Tohoku University, Japan.

[13] Minsky, M. 2007. The Emotion Machine, Simon & Schuster Inc. New York, NY.

[14] Nelson, T., Narens, L. 1990. Metamemory: A theoretical framework and new findings. The psychology of learning and motivation, 26:125–141.

[15] Smith, J., Shields, W., Washburn, D. 2004. The comparative psychology of uncertainty monitoring and metacognition. Behavioral and Brain Sciences, 26(03):317–339.

[16] Varela, F. G., Maturana, H. R., Uribe, R. 1974. Autopoiesis: the organization of living systems, its characterization and a model. Currents in Modern Biology, 5, 187.

[17] Zagal, J.C., Ruiz-del-Solar, J., Palacios, A.G. 2008. Fitness based identification of a robot structure. In Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems. 733-740.

[18] Zagal, J.C., Delpiano, J., Ruiz-del-Solar, J. 2009. Self-Modeling in humanoid soccer robots. Robotics and Autonomous Systems, 57(8), 819-827.

[19] Zagal, J.C., Lipson, H. 2009. Self-Reflection in Evolutionary Robotics: Resilient Adaptation with a Minimum of Physical Exploration. In Proceedings of the 11th Genetic and Evolutionary Computation Conference, GECCO'09, Montreal, Canada. 2179-2188.

[20] Zagal, J.C., Lipson, H. 2009. Towards Self-Reflecting Machines: Two-Minds in One Robot. In Proceedings of the 10th European Conference on Artificial Life. ECAL'09, Budapest.

[21] Zupanc, G. K. 2006. Neurogenesis and neuronal regeneration in the adult fish brain. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology, 192, 649-670. Spring

Neurodynamics of Cognition and Consciousness

Robert Kozma
Department of Mathematical
Sciences University of Memphis
Memphis, TN 38152, USA
1-901-678-2497
Robert.kozma.ctr@hanscom.af.mil

Walter J. Freeman
Division of Neuroscience
University of California at Berkeley
Berkeley CA 94720, USA
1-510-624-4220
dfreeman@berkeley.edu

ABSTRACT

Human cognition performs a granulation of the seemingly homogeneous temporal sequences of perceptual experiences into meaningful and comprehendible chunks of fuzzy concepts and complex behavioral schemas, which are accessed during future action selection and decisions. In this work a dynamical Theory-of-Mind (ToM) is presented to interpret experimental findings. In our approach meaningful knowledge is continuously created, processed, and dissipated in the form of sequences of oscillatory patterns of neural activity described through spatio-temporal phase transitions. The proposed approach has been implemented in computational and robotic environments.

General Terms

Algorithms, Experimentation, Theory.

Keywords

Neurodynamics, EEG, Phase Transition, Autonomous Agent, Intentionality.

1. INTRODUCTION

During the past years, strong evidence has emerged in the literature about the existence of sudden jumps in measured cortical activities. Lehman [1] identifies "micro-sates" in brain activity and jumps between them. Rapid switches in EEG activity have been described [2-4]. Synchronization of neural electrical activity while completing cognitive tasks is studied in various animals, e.g., in cats, rabbits, gerbils, and macaque monkeys [5-9]. Behavioral correlates of transitions between metastable cortical states have been identified [10-13]. A comprehensive overview of stability, metastability, and transitions in brain activity is given in [14-16]. One of the most influential theories of consciousness is the global workspace theory [17]. There is striking similarity between the cognitive content of phase transitions and the act of conscious broadcast in global workspace theory. It can be hypothesized that cortical phase transitions are in fact manifestations of such conscious broadcast events.

Freeman interpreted these findings using dynamic systems theory [8, 18]. Accordingly, the brain's basal state is a high-dimensional/chaotic attractor. Under the influence of external stimuli, the dynamics is constrained to a lower-dimensional attractor wing. The system stays in this wing intermittently and produces an amplitude modulation (AM) activity pattern. Ultimately, the system jumps to another wing as it explores the complex attractor landscape. Evidence from AM pattern analysis during tasks requiring sensory discrimination demonstrates the potential existence of multiple modes in neocortex that are mutually exclusive and cannot interact when accessed one at a time. The perceptual content is found in the phase plateaus of human scalp EEG. The EEG shows that neocortex processes information in frames like a cinema. The phase jumps show the shutter. The coordinated activity reveals self-similarity of the global dynamics that may form Gestalts (multi-sensory percepts) [19]. In the present work we use the theory of Freeman's K sets to model and implement biologically-motivated approach to intelligence.

2. NEURODYNAMIC MODEL USING K SETS

A hierarchical approach to spatio-temporal neurodynamics, based on K sets, was proposed by Freeman in the 70's [20]. K sets consist of a hierarchy of components with increasing complexity, including K0, KI, KII, KIII, KIV, and KV systems. They model the hierarchy of the brain starting from the mm scale to the complete hemisphere. Today, K sets are used in a wide range of applications, including classification [21-22], image recognition [23], and robot navigation [24-25]. Recent developments include KIV sets [26-28] for sensory fusion and modeling intentional behavior. They are applicable to autonomous control.

Intentionality means in the context of the present approach the cyclic operation of prediction, testing by action, sensing, perceiving, and assimilation. The significance of the dynamical approach to intelligence is emphasized by our hypothesis that nonlinear dynamics is a necessary condition of intentional behavior and intelligence in biological systems [25]. Therefore, understanding dynamics of cognition and its relevance to intentionality is a crucial step toward building more intelligent machines [29]. Specifically, nonconvergent dynamics continually creates new information as a source of novel solutions to complex

Performance Metrics for Intelligent Systems Workshop (PERMIS), September 21-23, 2009, 2004, National Institute of Standards and Technology. NIST, Gaithersburg, Maryland, USA.

problems. The proposed dynamical hypothesis on intentionality and intelligence goes beyond the basic notion of goal-oriented behavior, or sophisticated manipulations with symbolic representations to achieve given goals. Intentionality is endogenously rooted in the agent and it cannot be implanted into it from outside by any external agency.

The KIV model of the brain consists of three major components: cortex, hippocampal formation, and midline forebrain. Further, the amygdala striatum and brain stem provide link to the external motor part of the limbic system [30]. In the model, three types of sensory signals are distinguished. Each of these sensory signals provides stimulus to a given part of the brain, namely the sensory cortices, midline forebrain, and the hippocampal formation, respectively. The corresponding types of sensory signals are (i) exteroceptors; (ii) interoceptors (including proprioception); (iii) orientation signals. The convergence location and output are provided by the amygdala.

Experiments have been designed and implemented in computer simulation to demonstrate the potential of KIV operating on intentional dynamic principles. In the experiments we used an autonomous agent moving in a 2-dimensional environment. During its movement, the agent continuously receives two types of sensory data: (1) distance to obstacles; (2) and orientation toward some preset goal location. KIV makes decisions about its actions toward the goal. The spatio-temporal dynamics of this system shows sudden changes in the simulated cortical activity, which is in agreement with properties of metastable AM patterns observed in EEG data. These results indicate that the KIV model is indeed a suitable level of abstraction to grasp essential properties of cortical phase transitions as evidenced in intracranial and scalp EEG and MEG data.

3. CONCLUSIONS

Large-scale synchronization in the cortex, interrupted intermittently by short periods of desynchronization through phase transitions, is an emergent property of the cortex as a unified organ and it is an attribute to higher cognitive functions of the mind. Intensive studies are conducted towards the interpretation of the content of the metastable AM patterns separated by brief transitory periods of phase transitions. The intermittent synchronization-desynchronization cycle is a neuro-phyisiological correlate of intentionality and consciousness. The KIV model is capable of demonstrating this intentional dynamics and it is a candidate of implementing intentionality is artificial systems. Modeling is done in the context of the given level of knowledge available on the behavior of the autonomous system and its interaction with the environment.

4. REFERENCES

[1] Lehman D, Strik WK, Henggeler B, et al *Int. J Psychophl.*, 29:1–11, 1998.

[2] Stam CJ, Breakspear M, Cappellen,et al *Hum Brain Mapp* 19:63-78, 2003.

[3] Fingelkurts A .A., Fingelkurts A. A. *Mind and Brain*, 2:262–296, 2001.

[4] Fingelkurts A .A., Fingelkurts A. A. *Int J.Neurosci,* 114:843–862, 2004.

[5] Barrie JM. Freeman WJ, Lenhart M. (1996) *J. Neurophysiol*. 76: 520-539.

[6] Ohl, FW, Scheich, H and Freeman, WJ *Nature* 412: 733-736, 2001.

[7] Ohl FW, Deliano M, Scheich H et al *Biol. Cybernetics*, 88, 374-379, 2003.

[8] Freeman WJ, Burke B, Holmes M. *Hum. Brain Mapp.* 19: 248-272, 2003.

[9] Bressler SL. *Neuropsychopharmacology*, 28:S35-S39, 2003.

[10] Kelso, J.A.S.*"Dynamic patterns: The self-organization of brain and behavior."* MIT Press, 1995.

[11] Bressler SL, Kelso JAS. *Trends in Cognitive Sciences*, 5:26-36, 2001.

[12] Bressler SL. *Current Directions in Psychol. Sci.*, 2002, 11:58-61, 2002.

[13] Kelso, J.A.S.,Engstrom,D.A.*The complementary nature* MIT Press, 2006.

[14] Le Van Quyen M, Foucher J, et al *J. Neurosci. Meth*. 111: 83-98, 2001.

[15] Werner G. *BioSystems*, 87: 82-95, 2007.

[16] Tsuda I. *Beh. and Brain Sci*., 24(5):793–810, 2001.

[17] Baars, B. J. *A cognitive theory of consciousness.* MIT Press, MA, 1988.

[18] Freeman, W.J. *Clin. Neurophysiology*, 116, 1118-1129, 2005.

[19] Freeman, W.J. Proposed cortical 'shutter' mechanism in cinematographic perception, in:*"Neurodynamics of Cognition and Consciousness*," Perlovsky, L., Kozma, R. (eds), Springer, 2007.

[20] Freeman, W.J. *Mass Action in the Nervous System.* Acad. Press, N.Y, 1975

[21] Chang, H.J., Freeman, W.J., Burke, B.C. *Neur. Netw.*, 11, 449-466, 1998.

[22] Freeman, W.J., Kozma, R., Werbos, P.J. *BioSystems*, 59(2), 109-123, 2001

[23] Li, G., Z. Lou, L. Wang, X. Li, W.J. Freeman *Springer LNCS*, Vol. 3610, pp. 378-381, 2006.

[24] Harter, D., Kozma, R. *IEEE Trans. Neur. Netw*., 16(4), 565-579, 2005.

[25] Harter, D., Kozma, R. *Int. J. of Intelligent Systems*, 21, 955-971, 2006

[26] Kozma, R., and Freeman, W.J. *J. Integrative Neurosci*., 2, 125-140, 2003.

[27] Kozma, R., Freeman, W.J., Erdi, P. *Neurocomp.*, 52-54, 819-825, 2003.

[28] Huntsberger, T., Tunstel, E., Kozma, R. "Onboard learning strategies for planetary surface rovers," in: *Intelligence for Space Robotics*, A. Howard, E. Tunstel (eds). pp. 403-422, TCI Press, S.A., TX, 2006.

[29] Kozma, R., Fukuda, T. *Int. J. Intell. Syst*., 21(9), 875-879, 2006.

[30] Kozma, R., W.J. Freeman, *Neural Networks*, 22(3): 277-285, 2009.

Theory of Mind, Computational Tractability, and Mind Shaping

2009 Performance Metrics for Intelligent Systems Workshop
Tad Zawidzki
Philosophy, Mind-Brain-Evolution Cluster, Mind-Brain Institute
George Washington University

1. Introduction

Philosophers and psychologists have traditionally understood theory of mind as a human capacity for understanding and predicting human behavior based on the attribution of unobservable mental states, like beliefs and desires. The classical model views the human "mind reader" as a kind of scientist, formulating hypotheses about the unobservable causes of the behavior of her fellows, and then testing them through observation (Gopnik & Wellman 1995). Many argue that the attribution of unobservable, theoretical mental states increases the power of human social cognition over mere sensitivity to patterns of observable behavior, of the kind that characterizes the social cognition of most, if not all non-human animals (Tomasello & Call 1997). In this paper, I review some familiar problems with this view and suggest a novel strategy for dealing with them. In section 2, I explain why the timely and accurate attribution of mental states appears to be a computationally intractable task. In sections 3 and 4, I consider two standard models of human cognitive architecture aimed at mitigating problems of computational tractability: modularity and fast and frugal heuristics, respectively. I argue that these are unlikely to help in the case of theory of mind. In the final section, I show how "mind shaping" (Mameli 2001; Zawidzki 2008) – roughly, the practice of socializing individuals in ways that make human populations more homogeneous – can mitigate some of the problems raised in the earlier sections.

2. The Apparent Computational Intractability of "Mind Reading"

In the philosophical literature on theory of mind, and in much of the psychological literature, beliefs and desires are taken to be the central mental states required to make sense of behavior. The central "law" of so-called "folk psychology" is, roughly, the following: if an agent desires that P, and believes that not P unless she does Q, then the agent will desire to do Q. However, this and related laws must inevitably be qualified by potentially indefinite numbers of exceptions. In principle, any behavior is compatible with any finite set of mental states, given enough adjustments in other mental states. For example, just because someone *says* she supports Barack Obama, does not mean that she *believes* he is the best candidate, or that she will act to get him elected, etc. She might believe that John McCain is the best candidate, yet, at the same time, *desire* to conceal this fact from her interlocutors.

Philosophers call this the problem of "holism" (Morton 1996, 2003). Behavior is not correlated with finite sets of mental states. Rather, behavior is correlated with *whole systems* of indefinitely many mental states (thus the term "holism"). The holism problem jeopardizes the *timely* accuracy of any theory of mind based on the attribution of mental states like beliefs and desires. Human social cognition is extraordinarily powerful, yet, at the same time, extraordinarily efficient. We can often accomplish dramatic feats of interpersonal coordination in constantly shifting, dynamic social circumstances, where there does not appear to be enough time to consider and rule out all possible hypotheses about our interactants' mental states. It seems unlikely that some kind of brute search through all possible sets of mental states compatible with behavioral cues emitted by our interactants can support such fluid socio-cognitive competence.

3. Why a Theory of Mind *Module* Cannot Help

"Modularity" is the classic response to issues of computational tractability. If the human mind-brain deploys *encapsulated,* computational modules, with *pre-specified*, *domain-specific* databases, tractably searchable by *dedicated* processes, then, it is often claimed, problems of computational tractability can be avoided (Carruthers 2006). The idea is that, when confronting some domain-specific problem, e.g., predicting the behavior of another person, the human mind-brain need not search all information to which it has access. Such problems trigger activity in dedicated, domain-specific modules – in the case of the social domain, the theory of mind module – which are informationally isolated from other parts of the mind-brain, making the search for solutions exponentially more tractable. However, it is not clear that modularity can help in the case of social cognition. The reason is that, as Currie and Sterelny (2000) point out, *any* information might be relevant to the tasks of interpreting and predicting human behavior. They give the example of detective work: figuring out who committed a crime and for what reasons, and predicting the criminal's next move, is precisely the kind of problem that a dedicated, informationally encapsulated module cannot solve. The reason we like detective fiction is that there is no way of knowing, in advance, what sorts of information might be relevant to cracking a case.

Fodor (1983) introduced the notion of computational/cognitive modules, contrasting them with "central systems", which, he argued, are responsible for most belief fixation in human beings. Unlike modules, central systems are, according to Fodor, "isotropic", i.e., any information is potentially relevant to belief fixation. He focuses on examples from science, e.g., information about fluid behavior turned out to be relevant to fixing beliefs about the behavior of light in Nineteenth Century physics. Arguably, much everyday reasoning is similarly isotropic. There does not seem to be a way of pre-specifying what kinds of information might be relevant to selecting among products in a supermarket, or deciding whom to date, etc., in the way there would need to be were such problems tractable by encapsulated modules.

If Fodor is right that most human belief fixation is isotropic, and therefore a product of non-modular central systems, then a general case can be made against the modularity of theory of mind. For, the goal of theory of mind is to determine what beliefs are likely operative in another agent. But, if Fodor is right, the processes by which an agent fixes her beliefs are isotropic and therefore non-modular. So any interpreter of that agent cannot, herself, rely on some modular, informationally encapsulated theory of mind to determine which beliefs the agent will acquire and act on. If the interpretive target's decisions are determined by non-modular processes then so must be the interpreter's hypotheses about those decisions, if they have any chance of succeeding. Furthermore, besides figuring out how her target solves the belief fixation problems she faces, the interpreter must also determine to what information the target likely has access and what problems the target is most motivated by – problems the target need not solve. This compounds the problem facing the interpreter – not only must she, in effect, solve the same isotropic belief fixation problems as her target, she must also determine the parameters governing her target's solutions. Thus, the problems of computational tractability that arise for theory of mind do not appear to admit of a classical modularist solution.

4. Why Fast and Frugal Theory of Mind Heuristics Cannot Help Either

Fodorian modularity is an extreme solution to the problem of computational tractability, which seems unhelpful in many domains, particularly social cognition. However, there may be other kinds of modularity that evade some of the problems that have been raised for Fodorian modularity. Recently, Carruthers (2006) has defended a kind of modularity that is *not* based on some pre-specification of the *kind* of information that might be relevant to solving tasks in some domain. Instead, Carruthers argues that the problem of computational tractability can be solved using content-neutral, "fast and frugal heuristics" (Gigerenzer et al. 1999). Because such heuristics are content-neutral, there are no limits on the kinds of information they can consult. This is promising in the case of mind reading since, as we have seen, almost any kind of information can be relevant to this task. However, computational tractability is maintained by fast and frugal heuristics because there are strict limits on the *quantity* of information they can consult.

For example, "Take the Best" is a well-known fast and frugal heuristic. It requires that one recall criteria previously used to distinguish between alternatives in some domain, determine which criterion distinguished best, and use that criterion on one's current decision. For example, when asked which of two German cities is larger, one might recall that, previously, having a professional soccer team distinguished best between larger and smaller cities, and so one asks which, if either, has a professional soccer team. If neither or both do, one then proceeds to the next best criterion. In order to avoid intractable search, "Take the Best" has a "stopping rule" that suspends search if it cannot arrive at an answer after some small, finite number of iterations (Gigerenzer et al. 1999; Carruthers 2006).

Fast and frugal heuristics combine computational tractability with openness to a wide variety of potentially relevant information. For example, although "Take the Best" is restricted to considering only information that a particular agent has recently consulted when reasoning about a certain domain, this restriction is content neutral. It includes different information for agents with different histories of reasoning about a domain. There is no reason why, for a different agent reasoning about relative population sizes of German cities, [1] "Take the Best" could not consult relative crime rates instead of presence of professional soccer teams (Carruthers 2006). Such heuristics are computationally tractable because they restrict search based on an agent's current epistemic context, including relevant recent searches, not because they restrict search based on the content of the relevant domain, e.g., social, or physical, etc. So there is no need to pre-specify the kinds of information likely to be relevant to each domain. This mitigates the problem that Currie and Sterelny (2000) raise for modular theory of mind. Any information, e.g., the number of asparagus spears left on a plate, may be relevant to a theory of mind task. But only information that has recently been useful on similar tasks is consulted on any particular occasion.

[1] Or for the same agent at a different time.

Unfortunately, this proposal seems unlikely to work for theory of mind tasks. The problem is that fast and frugal heuristics work only in domains characterized by extreme homogeneity. "Take the Best" solves new problems based on strategies that a particular agent has successfully applied to similar problems in the recent past. Unlike more sophisticated, statistical learning algorithms, such heuristics make no attempt to insure that the sample from which they generalize is unbiased. Their quickness and frugality consists precisely in the fact that they avoid such formal niceties. Strategies that *happen* to have worked for a particular agent in the recent past are taken to be appropriate for current and future problems. Such contingent regularities can safely be assumed in certain extremely homogeneous, well-behaved domains. However, there is every reason to deny that the social domain is like this.

Ironically, if human beings rely on fast and frugal heuristics to fix their beliefs, then the dependence of such heuristics on the idiosyncratic background knowledge of particular individuals is likely to make discovering their beliefs computationally intractable for fast and frugal *theory of mind* heuristics. This is because such social cognition would require quickly and frugally uncovering the idiosyncratic background knowledge on which one's interpretive targets rely. But fast and frugal theory of mind heuristics can depend only on what has worked for the *interpreter* in the recent past, and there is no reason to think that this is any guide to the idiosyncratic background knowledge of a new interpretive target. Furthermore, as Sterelny (2003) emphasizes, human beings appear to have strong, biological incentives to behave in ways that are unpredictable relative to heuristics that their potential competitors have previously used to predict them. Lastly, there are good reasons to think that individual variation among human beings is extreme, compared with other species: we have unmatched capacities for creative cognition and conation which involve random processes heavily dependent on idiosyncratic learning history (Carruthers 2006), and extreme phenotypic plasticity is likely a human adaptation to extremely variable physical and social environments (Sterelny 2003). So, we have every reason to suppose that fast and frugal heuristics, the reliability of which relies on extreme homogeneity in the domains to which they apply, cannot support reliable social cognition.

Perhaps there are more specific mind reading heuristics that can evade some of these problems. The most influential models of fast and frugal social cognition appeal to some kind of simulation (Goldman 2006). Interpreters save on the computational costs of interpretation by simply projecting, in some sense, their own decision procedures onto others. But the accuracy of such simulation heuristics obviously depends on extreme homogeneity in human populations: interpreters and their targets must prioritize problems in similar ways and make decisions based on similar information and heuristics. And, as we have seen, there are good reasons to doubt that such homogeneity exists in human populations.

5. Mind Shaping as Human Homogenizer

Let me end by proposing a sketch of how I think humans solve the problems reviewed above. I propose that certain low-level, automatic mind-shaping mechanisms, prevalent in human populations, work to homogenize them, thereby making fast and frugal theory of mind heuristics more effective.

Suppose human beings have an automatic, default disposition to compare their own behavior to that of others, monitoring for any discrepancies. If the other is higher status, discrepancies tend automatically to issue in attempts at self-modification: one tries to change one's dispositions such that one's behavior better matches that of the high status model. If the other is lower status, discrepancies tend automatically to issue in attempts to modify the other: one tries to change the other's dispositions, as in teaching offspring or punishing norm transgressors, such that the other's behavior better matches one's own. Assuming that judgments of status are largely homogeneous in a population, such mind-shaping dispositions would tend to further homogenize populations, counteracting the "centrifugal" forces causing individual variation. And such homogeneity would render fast and frugal theory of mind heuristics more reliable. For example, the "Take the Best" heuristic would be more likely to work. In a population of similarly socialized individuals, decision strategies that happen to have worked well in recent interpretive contexts are more likely to work in new circumstances. Similarly, variations of the simulation heuristic would also be more reliable: procedures that interpreters use in their own decision-making would be more likely to be used by their interpretive targets as well.

Solutions to problems of computational tractability that arise for theory of mind, I want to urge, lie not *within* human mind readers, but, rather, *outside* of them. Rather than deploy intractably sophisticated theories of each other's minds, we make use of a variety of low-level mind-shaping dispositions to insure that our fellows are sufficiently familiar so that fast and frugal heuristics can help us accomplish our socio-cognitive goals. We teach our children to behave in ways that make them easier to interpret (Bruner 1983, Mameli 2001, McGeer 2001). We sanction those who behave in ways that are harder to interpret (think of the damage to status which often results from weakness of the will or absentmindedness) (Zawidzki 2008). We display unconscious, automatic, and irresistible tendencies to conformity, such as the "chameleon effect" (Chartrand & Bargh 1999). Such mechanisms shape our socio-cultural environment in ways that make coordination exponentially more tractable than it would be were we to exhaustively search the mental state hypothesis spaces compatible with the finite sets of potential-interactant behaviors with which we are familiar.

References

Bruner, J. 1983. *Child's talk*. New York: Norton.

Carruthers 2006. *The Architecture of the Mind*. New York: Oxford University Press.

Chartrand, T.L., and J.A. Bargh. 1999. The chameleon effect: The perception-behavior link and social interaction. *Journal of Personality and Social Psychology* 76: 893 – 910.

Currie, G., and K. Sterelny 2000. How to think about the modularity of mind reading. *Philosophical Quarterly* 50: 145-60.

Fodor, J. 1983. *The Modularity of Mind*. Cambridge, MA: MIT Press.

Gigerenzer, G., P. Todd, and the ABC Research Group 1999. *Simple heuristics that make us smart*. New York: Oxford University Press.

Goldman, A. 2006. *Simulating minds*. Oxford: Oxford University Press.

Gopnik, A., and H. Wellman. 1995. Why the child's theory of mind really is a theory. In *Folk psychology*, ed. M. Davies and T. Stone, 232 – 58. Oxford: Blackwell.

Mameli, M. 2001. Mindreading, mindshaping, and evolution. *Biology and Philosophy* 16: 597 – 628.

McGeer, V. 2001. Psycho-practice, psycho-theory and the contrastive case of autism. *Journal of Consciousness Studies* 8, no. 5 – 7: 109 – 32.

Morton, A. 1996. Folk psychology is not a predictive device. *Mind* 105: 119 – 37.

Morton, A. 2003. *The importance of being understood*. London: Routledge.

Sterelny, K. 2003. *Thought in a Hostile World*. Oxford: Blackwell.

Tomasello, M., and J. Call. 1997. *Primate cognition*. New York: Oxford University Press.

Zawidzki, T. 2008. The function of folk psychology: mind reading or mind shaping? *Philosophical Explorations* 11(3): 193 – 210

Data Collection Test-Bed for the Evaluation of Range Imaging Sensors for ANSI/ITSDF B56.5 Safety Standard for Guided Industrial Vehicles

Will Shackleford
National Institute of Standards and Technology(NIST)
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
(301) 975-4286
shackle@nist.gov

Roger Bostelman
National Institute of Standards and Technology(NIST)
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
(301) 975-3426
roger.bostelman@nist.gov

ABSTRACT

In this paper, we describe the process by which we collected sensor data for the evaluation of 3D LIDAR (Light Detection and Ranging). Data were also collected simultaneously from SONAR (Sound Navigation and Ranging) sensors, navigation systems, 2D Laser Measurement Sensor (LMS) and a color camera. We describe software developed to perform data collection and allow for evaluation of the data both offline and in real-time during the data collection and briefly cover the experiments themselves where various obstacles were placed in front of a moving vehicle and results were recorded as to whether the obstacle was detected or not.

Categories and Subject Descriptors

I2.10 [**Vision and Scene Understanding**]: 3D/stereo scene analysis

General Terms

Performance, Design, Experimentation, Standardization,

Keywords

LIDAR, LADAR, Data-Collection, Autonomous Guided Vehicles(AGV), B56.5, Sonar.

1. INTRODUCTION

The Industrial Truck Standards Development Foundation (ITSDF) manages the “ANSI/ITSDF B56.5 Safety Standard for Guided Industrial Vehicles and Automated Functions Of Manned Industrial Vehicles” as approved by the American National Standards Institute (ANSI)[2]. The National Institute of Standards and Technology (NIST) has been performing measurements to be used as background information towards changes in the standard. Automated Guided Vehicles (AGVs) are typically programmed to follow prescribed paths but still need sensors to detect obstacles such as closed doors, equipment, personnel or material left temporarily in the vehicle's path. Currently they rely heavily on 2D line scanners with a physical bumper as the final backup to stop the vehicle in these cases. The 2D line scanners work well against most vertical obstacles but it takes many of them to completely protect against overhanging obstacles and even then they do not scan the full volume of space the vehicle will travel through. Flash LIDAR is a relatively new class of range imaging sensors with the potential to scan 3D volumes faster than the line scanning systems. To evaluate them, a consortium of AGV vendors was formed that took preliminary data with several Flash range imaging systems and selected one for further development and investigation. This is the sensor used for this work. In 2008 work was done with the stationary vehicle and with a manually moved cart.[1] In 2009, the data collection system was integrated with the Mobility Open Architecture Simulation and Tools (MOAST) framework.[3] This allowed the system to collect data while being driven autonomously.

2. Test-Bed Hardware

All of the sensors are mounted on a robot. The sensors include:*

- Spinning Laser Positioning System (SLPS) - Provides absolute position using a spinning laser that detects special reflective targets mounted on walls and fixed structures.
- Safety Laser Measurement Sensor(LMS) – 2D line scanner which detects obstacles but only at a single height.
- FLASH – 3D Flash LIDAR Camera provides range/intensity for every pixel in the image.
- Color Camera – Provides better documentation of each test.
- Positioning Camera (CamPos). -- A camera system pointed at the ceiling to provide absolute position using special targets mounted on the ceiling.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
ACM 978-1-60558-747-9/09/09

* Certain commercial equipment, instruments, or materials are identified in this paper to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

- SONAR – Sound navigation and ranging sensors mounted to the vehicle to detect obstacles in the vehicle's path.

The positions of the sensors are shown in Figure 1.

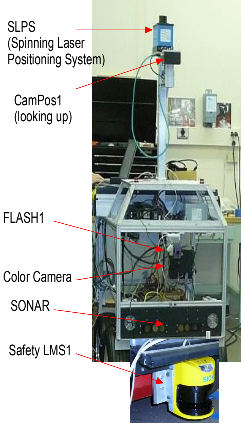


Figure 1: Sensor Positions

3. Software Architecture

3.1 Main Software Architecture

Figure 3.1 provides the main software architecture diagram for the system. The moastLogRecordSuper supervises and coordinates MOAST controller moves with the starting and stopping of the data collection. The MOAST framework aids in the development of autonomous robots. It includes an architecture, control modules, interface specs, and data sets and is fully integrated with the USARSim (Unified System for Automation and Robot Simulation) system.[3] The MOAST controller is used to generate trajectories for Player and send a stream of desired translation and rotational velocities to Player. Player is a cross-platform robot device interface and server that supports a number of robot platforms and sensors including the commercial platform used as our base and LMS1.[5]

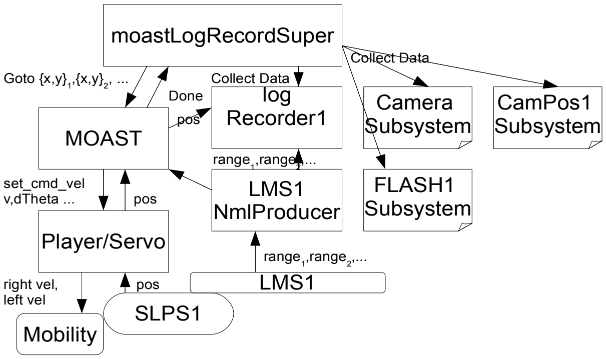


Figure 2: Main Software Architecture Diagram

3.2 Neutral Message Language(NML)

The Neutral Message Language (NML) [4] provides both the communication system and the facilities for reading and writing the data files in a portable, transport, platform and programming language independent manner. NML is part of the Real-time Control System (RCS) Library[11]. It provides a common API (Application Programming Interface) for both potentially remote TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) communications as well as faster shared memory based communications. It is used for all inter-process communications within the system including internal MOAST communications with the exception of communication between MOAST and Player [5] which uses a Player-defined socket interface.

3.3 Sensor Subsystem Software Architecture

Each sensor is handled by a similar subsystem as shown in Figure 3.2 Each sensor comes on a data bus such as Firewire, USB, RS232 Serial, or Ethernet with a format and protocol usually unique to that model sensor. A separate process is used for each sensor that essentially acts as a device-driver and converts data received from the bus to an NML message and writes it to both a queued and non-queued NML buffer. The queued buffer is read by the LogRecorder which then writes the data to disk. A non-queued buffer is used to provide a real-time display of the data. The real-time display is needed during data collection to ensure that the sensor is working, and that objects of interest are within the field of view and to adjust sensor configuration parameters. Using separate processes for the NmlProducer and LogRecorder means that intermittent delays in writing to the disk will not cause the system to miss frames as the NmlProducer could continue filling the QueuedBuffer while the LogRecorder is delayed. It also isolates the code most likely to need debugging and not to be portable, which is the sensor specific driver code. Each viewer is built so that it can display either live data from the NML channel or logged data from files.

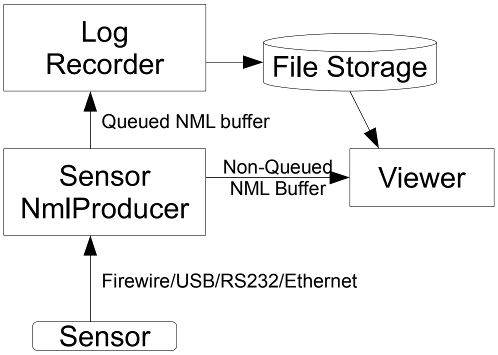
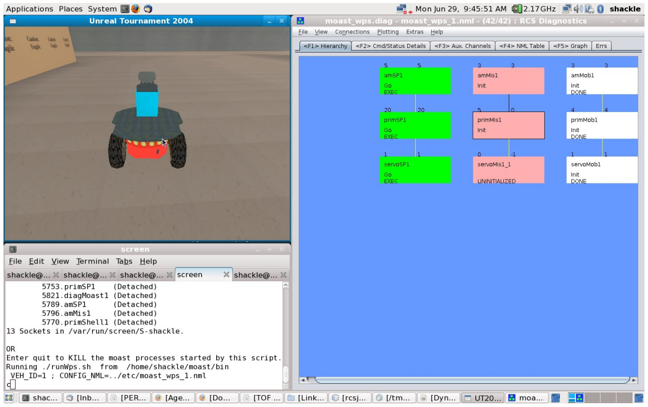


Figure 3: Sensor Subsystem Architecture

The LogRecorder and Sensor NmlProducer are written in C++ for performance and low-level access to hardware or operating system resources. Each viewer is written in Java so that the logged data may be evaluated on any platform.

3.4 MOAST

MOAST is a control system framework that works both in simulation and on real hardware. The simulation uses a commercial gaming engine and is developed under the Performance Simulation Project at NIST.[3][10] Figure 4 shows the USARSim module/plugin showing a small simulated robot in a warehouse setting and the RCS Diagnostics tool connected to the bottom 3 levels of the MOAST (AM,PRIM,SERVO). The Autonomous Mobility (AM) level combines data from the various sensors into a map and uses the map to compute a path from the vehicle's current position to the final goal position that avoids all known obstacles. The PRIM level takes a list of intermediate positions or arc segments produced by the AM level and considers the dynamics and kinematics of the vehicle to produce a series of SERVO commands sent one at a time as the vehicle moves along the path. The SERVO level takes commanded right and left wheel velocities and interfaces with the hardware to achieve those velocities. The moastLogRecordSuper can connect either to the PRIM or AM levels. When commands are sent directly to PRIM the robot will simply follow a set of way-points and ignore the sensors. When commands are sent to the AM level the robot will use the sensors to build a map and plan around obstacles. All of the tests done so far have sent commands directly to the PRIM level which gives us greater control over exactly where the robot travels but the option of using the AM level for future tests remains available.



```
        5753.primSP1    (Detached)
        5821.diagMoast1 (Detached)
        5789.amSP1      (Detached)
        5796.amMis1     (Detached)
        5770.primShell1 (Detached)
13 Sockets in /var/run/screen/S-shackle.

OR
Enter quit to KILL the moast processes started by this script.
Running ./runWps.sh  from  /home/shackle/moast/bin
 VEH_ID=1 ; CONFIG_NML=../etc/moast_wps_1.nml
q
```

Figure 4: Left: Unreal Tournament with USARSim module simulation/animation, RIGHT: RCS Diagnostics tools showing AM,PRIM,SERVO levels of MOAST.

3.5 NML Packed Message Files

All of the data after collection is stored in NML packed message files. The format will hopefully provide the openness and flexibility of text, CSV (comma-separated values)[8] or XML (Extensible Markup Language)[9] files and the efficiency in both disk space and processing time of binary formats. This file format allows for easy reading and writing of even complex data structures. Generic tools can be used to display the contents of the files or users can write their own programs to read and write the files with a simple API. Also a memory map file listing the offset to every variable allows the files to be accessed outside the API. Tools were written to convert these files to plots, movies, and separate still image files as appropriate.[6] A website is under development that should allow users to access the collected data using any of the tools discussed below, download the message files, export appropriate subsections to a spreadsheet , and etc.

The file sizes for one LIDAR data frame for the FLASH1, which has a resolution of 144 pixels x176 pixels (25344 pixels), are compared in Table 1. The packed format includes configuration information and XYZ coordinates from every pixel as well as a range and intensity value. Obviously if a display is all that is needed, saving JPEGs requires by far the least data but it does not include configuration data or the XYZ point cloud and even the exact range and intensity values cannot be recovered from the JPEG. The CSV and spreadsheet files with the same data are larger and do not contain the configuration information or XYZ coordinates although those could be added. There are many ways the information could be stored in XML, but one of the most straight-forward methods produced a file six times larger than the original packed data.

File Format	XYZ ?	Config.?	Size(Bytes)
XML	Yes	Yes	3047494
Spreadsheet	Yes	No	2829824
CSV	Yes	No	1374147
NML Packed	Yes	Yes	507007
JPEG	No	No	3382

Table 1: LIDAR Data Frame File Sizes

3.6 Flash LIDAR Display

Flash LIDAR cameras generally provide both a range image (Figure 5) and and intensity image (Figure 6). Although all tests done this year were with the FLASH1, the data structure used to store the data was originally developed with two other Flash Lidar Cameras and therefore all tools should work with all three cameras. It is often easier for people to see objects in the intensity image. However, it is the range image that is of most use for the AGV's obstacle detection and avoidance algorithms. One goal of the experiments was to determine whether the obstacles were detected by the sensor at various ranges. Unfortunately this could be somewhat subjective. Just because a person (especially one already familiar with the scene) could make out something is no guarantee that it is possible to use the data to build a reliable automatic obstacle detection algorithm. For this reason the viewer includes its own obstacle image classification window. The results of the obstacle detection are shown in Figure 7.

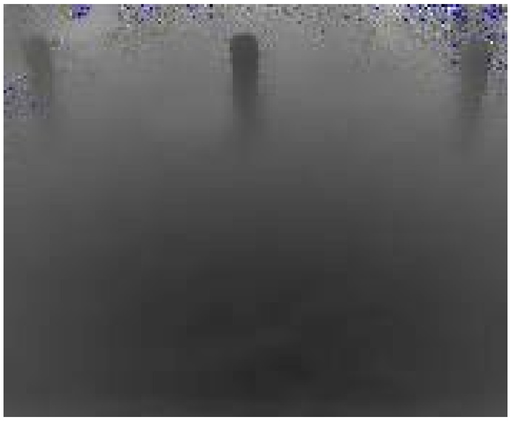


Figure 5: Flash LIDAR Range Image of 3 Cylinders in front of AGV (Black=near, White=far)

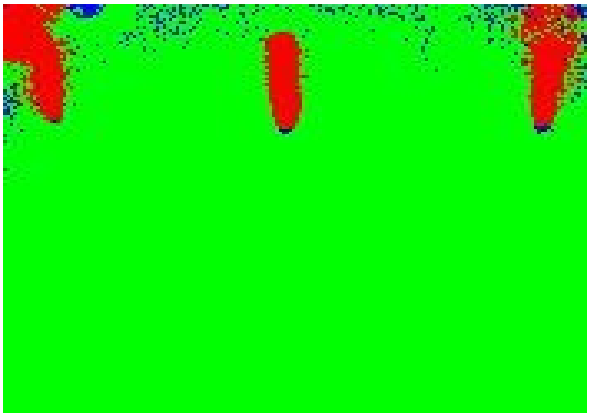


Figure 7: Obstacle Detection Based On Flash LIDAR Range Image (Green=ground,Red=Obstacle, Blue=Unknown)

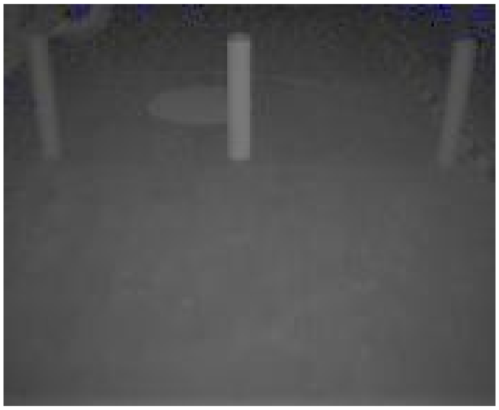


Figure 6: Flash LIDAR Intensity Image of 3 Cylinders in front of the AGV

The obstacle detection includes filters to eliminate points with too high/low intensity, too high/low range values, isolated points or points not near neighboring points. It then simply rotates the point cloud to adjust for the mounting of the sensor and applies a height threshold. Points below the threshold are ground and points above the threshold are obstacles. All the parameters are adjustable both in real-time and when displaying logged data from a Graphical User Interface to allow for a very conservative to very lenient obstacle detector. A text display allows min/max and average intensity or range values to be obtained for any selected rectangle in the image.

The sensor has two problems. First, it cannot distinguish obstacles at multiples of its modulation wavelength (about 6 m). i.e., if the modulation wavelength was 6 m an object 7 m away returns the same value as one 1 m away. Second, when near highly reflective surfaces such as the ones commonly used by the AGV's navigation system the entire scene is strongly distorted. For this reason all of our tests have the sensor pointing down towards the floor which eliminates any possibility of an object being farther away than the modulation wavelength and also keeps the sensor away from the eye-level navigation reflectors.

3.7 Camera Positioning Navigation System/Spinning Laser Positioning System

Spinning Laser Positioning System (SLPS) is typically used for industrial AGV's. CamPos is a more recent alternative to the spinning laser based navigation systems that uses a camera and 2D bar code targets mounted on the ceiling.[7] One set of tests that we completed was to record both the CamPos and SLPS positions simultaneously while driving the ATRV manually. We purposely mounted the 2D bar code targets in a fairly regular ceiling pattern of 1.2 m spacing and disregarded partially occluded ceiling obstructions. Where obstructions mostly or completely covered targets, we moved those targets to a less obstructed ceiling location. Although the manufacturer suggests non-obstructed targets, we are looking for ways to measure performance of these systems when they are in the ideal and non-ideal configurations. Figure 8 (left) shows clear view and partially occluded views of ceiling-mounted 2D bar code targets. A similar situation can occur with the SLPS positioning of wall-mounted reflectors as shown in Figure 8 (right). Here, reflectors are shown in clear view and partially occluded views that could also result in less than robust vehicle positioning. In previous tests, we also found issues with this system when highly reflective surfaces appear to the sensor as system reflectors.

For this recent experiment, we tested only the CamPos system targets being partially obstructed. The experiment showed that while the CamPos tracked the SLPS position well over much of the approximately 36 m long x 10 m wide course, there were measurement issues in places where the CamPos could not simultaneously see more than one target (see Figure 9 left and right). These were caused by an overhead crane system (as shown in Figure 8 (left)) and ceiling supports that obscured a few bar code targets.

The user of both the CamPos and SLPS systems can ideally mount sensor targets appropriately to get maximum accuracy as specified by the manufacturer. As targets are relatively inexpensive for these sensor systems, adding more and calibrating the targets mounted in non-occluded areas easily solves these issues.

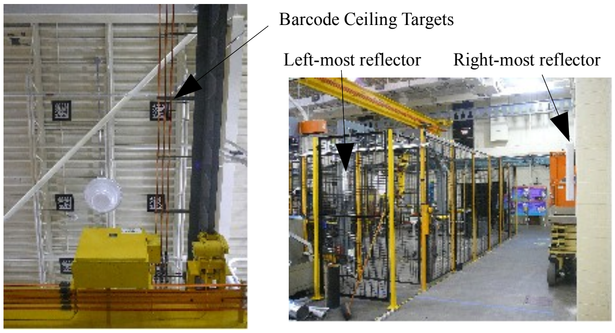


Figure 8: (left) Crane electrical bars partially occluding the ceiling-mounted 2D bar code targets of the CamPos system; (right): clear view of the right-most reflector of the SLPS system and partially occluded view of the left-most reflector by a robot cage

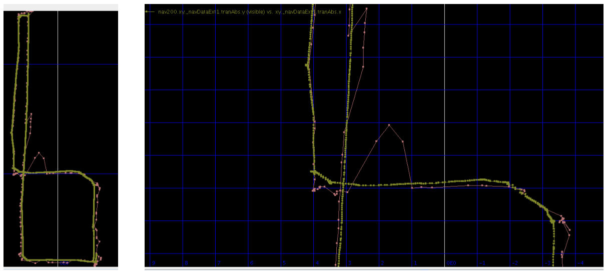


Figure 9: (left): CamPos (pink) versus SLPS position (green/yellow) plots; (right): Zoom of CamPos versus SLPS position where ceiling barcodes are partially occluded.

Both the CamPos and SLPS data are recorded with independent running implementations of the sensor subsystem as described in section 3.3. The data was plotted from the recorded data offline using the plotter included with the RCS(Real-Time Control System) Diagnostics Tool.

3.8 Safety Laser Measurement Sensor (LMS)

The LMS is a laser scanning system that currently detects obstacles at longer ranges and higher reliability than the Flash LIDAR but only in a single plane. In Figure 10 the LMS was scanning through a fence that will be placed around a robot work station. In the raw sensor data the LMS sees both the fence and the object on the other side of the fence.

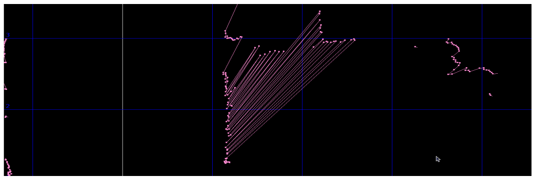


Figure 10: LMS1 data scanning both a fence and obstacles on the other side of the fence.

3.9 Color Camera

The main use of the color camera currently is to overlay images from the 3D Flash LIDAR to better identify the source of artifacts as shown in Figure 11.

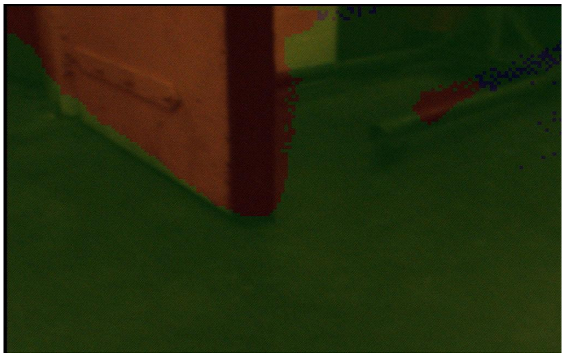


Figure 11: Color camera image overlaid with obstacle detection data from the flash LIDAR sensor

4. Static Test Results

A series of static tests were performed using the sensors listed in section 2 Test-bed Hardware. The tests included covering three different test pieces with a variety of surfaces and testing with all sensors recording at a variety of positions, orientations and ranges. The test pieces included the two pieces already part of the B56.5 standard, a 200 mm diameter x 600 mm long horizontal cylinder and a 70 mm diameter x 400 mm high vertical cylinder. The third test piece was the new 500 mm x 500 mm flat surface target. The coverings were selected to change the reflectivity and specularity of the test pieces for the optical sensors as well as the sound absorption properties for the SONAR.

For both optical and SONAR sensors, changing the angle of reflection with the flat target had a significant effect on the measured intensities and ranges and in some cases whether the sensor received a return or not. The reflectance and specularity of the coverings also had an effect on the optical sensors however none of the sound absorbing materials tested had a significant effect on the SONAR.

5. Changes to the ANSI/ITSDF B56.5 Safety Standard for Guided Industrial Vehicles

As a result of tasks conducted using this test bed, changes were recommended to the ANSI/ITSDF B56.5 committee to add an additional flat target, to test at a variety of reflectance and specularity values and at different reflection angles and ranges. There was also a recommendation to perform dynamic tests at various vehicle speeds. We may provide further recommendations after completing the dynamic tests.

5.1 Conclusions

NIST has created a unique test bed and data-collection platform. Although its initial use was to provide input to the ITSDF standard development process, it should be possible to provide industry and/or the research community with independent evaluations of sensor technologies or provide data for obstacle detection algorithm development or verification. The test bed will continue to be updated with additional sensors.

6. REFERENCES

[1] Roger Bostelman, MS; William Shackleford, "Time of Flight Sensors Experiments Towards Vehicle Safety Standard Advancements", submitted to Computer Vision and Image Understanding Journal

[2] Industrial Truck Standards Development Foundation, (2005). ITSDF B56.5 Safety Standard for Guided Industrial Vehicles and Automated Functions of Manned Industrial Vehicles, http://www.itsdf.org.

[3] Mobility Open Architecture Simulation and Tools (MOAST) framework, https://sourceforge.net/projects/moast

[4] Neutral Message Language (NML) , http://www.isd.mel.nist.gov/projects/rcslib/NMLcpp.html

[5] Player Project, http://playerstage.sourceforge.net/index.php?src=player

[6] NML Message Files, http://www.isd.mel.nist.gov/projects/rcslib/message_files.html

[7] SkyTrax System, http://www.sky-trax.com/products/STS.php

[8] Wikipedia: Comma-separated values, http://en.wikipedia.org/wiki/Comma-separated_values

[9] W3C Extensible Markup Language(XML), http://www.w3.org/XML/

[10] Performance Simulation Project, http://www.nist.gov/mel/isd/ks/persim.cfm

[11] Real-Time Control Systems Library, http://www.isd.mel.nist.gov/projects/rcslib/

Ground Truth Data Using 3D Imaging for Urban Search and Rescue Robots

Nicholas A. Scott, Alan M. Lytle
National Institute of Standards and Technology
100 Bureau Drive Stop 8611
Gaithersburg, MD, 20899-8611, USA
{nicholas.scott, alan.lytle}@nist.gov

ABSTRACT

The National Institute of Standards and Technology (NIST) is leading an effort to develop performance standards for urban search and rescue robots (US&R). An important component of developing performance standards for these robots is capturing ground truth data that represents the geometry of the robot operating environment. This paper describes two ground truth data collection efforts conducted in 2006 and 2008 at the Texas Engineering Extension Service Disaster City training facility in College Station, Texas. Several indoor and outdoor training scenarios were captured with 3D imaging systems and the data is now publicly available through NIST to support research and development of robotic technologies for the US&R domain.

Keywords

robotics, ground truth, 3D imaging, urban search and rescue

1. INTRODUCTION

The National Institute of Standards and Technology (NIST) is leading an effort to develop performance standards for urban search and rescue (US&R) robots [5]. As part of this effort, NIST organizes events that allow emergency responders, robot manufacturers and robotics researchers to work shoulder-to-shoulder within world-class responder training facilities [4]. Commercial-off-the-shelf products and laboratory prototypes are operated by responders in realistic operating scenarios while being observed and supported by the technical experts. These events allow responders to better understand state-of-the-art robot capabilities and limitations, and provide manufacturers and researchers unfiltered access to subject matter experts. An additional activity at these responder events is the exercise of various test methods and performance metrics under development to support the overall goal of creating a suite of performance standards for US&R robots. An important component of developing performance standards for these robots is capturing ground truth data of the training scenarios and the test methods.

This paper describes two ground truth data collection efforts conducted in 2006 and 2008 at the Texas Engineering Extension Service (TEEX) Disaster City training facility in College Station, Texas. Several indoor and outdoor training scenarios and two test methods were captured with 3D imaging systems[1] and the data is now publicly available through NIST to support research and development of robotic technologies for the US&R domain.

This paper begins by describing the motivation behind the data collection efforts. Section 3 provides information regarding the captured scenarios and Section 4 presents metrics for the data collected. Finally, Section 5 discusses future NIST efforts using this data.

2. MOTIVATION

When a disaster occurs, previously benign terrain may become difficult or impossible to traverse. Buildings collapse, roads and bridges are destroyed, and previously smooth, obstacle free terrain may contain large obstacles and discontinuities. In order to perform search and rescue operations, responders must assess the terrain in order to employ assets that possess the correct mobility to get to desired locations. For responders to effectively use robotic technologies on US&R missions, they must understand how different robotic platforms perform in diverse terrain. Developing tests and performance metrics to enable this understanding not only supports the responder's use of robots in the field, but also provides important information to support further research and development.

An essential element in defining these performance metrics is the independent capture of an accurate ground truth representation of the robot's operating environment. This ground truth data can support a wide range of research including mobility performance metrics, terrain characterization, mapping algorithm evaluation, and virtual environment construction.

For US&R robotics, both qualitative and quantitative measures of the environments in which platforms are tested and deployed to support mobility performance metrics and terrain characterization are of great interest. For examples of qualitative measures of an environment, consider trail rating systems for ski slopes or the Beaufort Wind Force Scale for estimating wind speed from sea state. Quantitative US&R terrain characterization metrics would enable predictable and consistent ways of representing difficult ter-

PerMIS'09 September 21-23, 2009, Gaithersburg, MD, USA
ACM 978-1-60558-747-9/09/09.

[1] A 3D imaging system is a non-contact measurement instrument used to produce a 3D representation (for example, a point cloud) of an object or a site [1].

rain (e.g., rubble) and provide fair comparison of platforms. An example of quantitative metrics in the US&R context could be a specific measure of the traversibility of the terrain surface derived using techniques such as height, slope, and roughness estimation from plane fitting, fractal dimensional analysis or wavelet energy statistics. Traversibility is a well-studied discipline, particularly in the context of unmanned ground vehicle path planning. The challenge is to standardize a universally accepted measure for US&R robot evaluation.

In addition to understanding the terrain, ground truth data supports mapping algorithm evaluation and virtual environment construction. More and more developers are including map creation in their operator toolkits, and map generation is a highly desired capability amongst responders. The ground truth data collected can serve as a baseline for evaluating the performance of 2D and 3D mapping systems deployed on mobile sensors. Ground truth data used to evaluate mapping can also be used in simulation environments. NIST, along with partner organizations, is investigating how to represent the point clouds and/or derivative terrain models within simulation environments such as NIST's USARSim [2]. Importing point, polygonal, or surface models of realistic training scenarios into simulation systems can make the training scenarios themselves accessible to a wider set of developers. Responders, researchers, developers, and other interested personnel will be able to navigate the scenarios, to some degree of fidelity, without having to physically travel to the location. Intelligent behaviors for semi-autonomous robots can also be virtually tested within the models.

3. DATA COLLECTION

To support the need for ground truth data, NIST researchers gathered high-resolution 3D image data for five training scenarios and two test methods at Disaster City, the TEEX National Emergency Response and Rescue Training Center in College Station, Texas. Disaster City is a 52-acre site that provides full-scale collapsible structures, rubble piles, and wrecked transportation structures for search and rescue training and is considered by many to be the most comprehensive emergency response training facility presently available. The ground truth data was collected during two separate NIST organized events at Disaster City.

3.1 Collection One

The first data collection effort was held on April 4-6, 2006 and focused on outdoor US&R environments. Data for three different scenarios was collected:

- Concrete Rubble Pile
- Wood Rubble Pile
- Passenger Trains

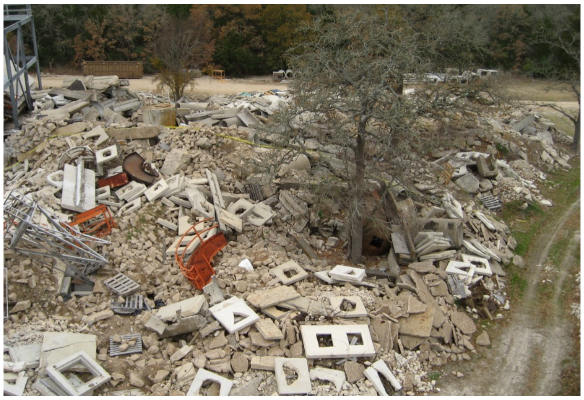


Figure 1: Concrete rubble pile training scenario which simulates a fully collapsed concrete structure.

The concrete rubble pile scenario is depicted in Fig. 1. This scenario simulates a fully collapsed concrete structure with interior voids. The rubble pile primarily consists of concrete and reinforcing bars (rebar). Large concrete slabs, barriers, and pipes, generally several meters in length, provide the support for many of the voids. Small concrete rocks, typically 30 cm to 50 cm in diameter, fill in the the space around the larger concrete pieces. Rebar and other metal structures are scattered throughout the rubble pile. Robots are deployed from the perimeter either directly into subterranean voids or over top of the rubble pile to search for victims and map the area.

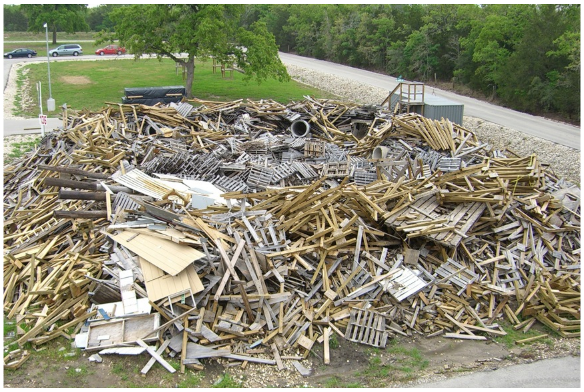


Figure 2: Wood rubble pile training scenario which simulates a fully collapsed wood structure.

The wood rubble pile scenario is shown in Fig. 2. This scenario simulates a fully collapsed wood structure with interior voids. The rubble pile consists of several meter length wood planks and wood pallets. Interior voids are created by using several meter sections of concrete piping. Robots are deployed from the perimeter of this pile by climbing, throwing, or launching into the central area to look for victims and to map the area.

The passenger trains scenario is depicted in Fig. 3. This scenario mimics the collision and partial derailment of passenger rail cars and industrial hazardous material tanker cars carrying an unknown substance. Robots are deployed from the perimeter of the wreck and circumnavigate the trains, tracks, and rubble to map the perimeter of the scene and determine the location of each car. The underside of the elevated car and the interior of the car on its side is explored for victims and to look for placards describing what hazardous material is onboard.

3.2 Collection Two

The second data collection effort was held on November 17-21, 2008 and focused on indoor US&R environments.

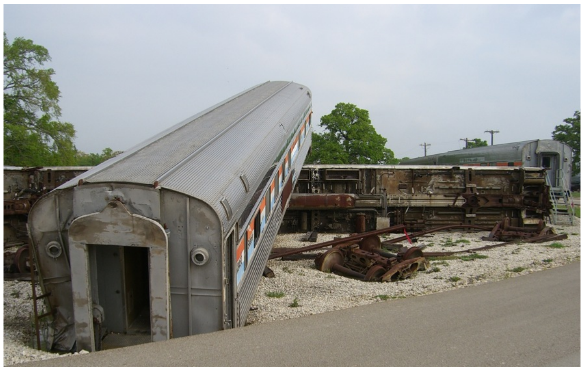


Figure 3: Passenger trains training scenario which simulates the collision and partial derailment of passenger rail cars.

Data for two training scenarios and two test methods was collected:

- Single Family Dwelling
- House of Pancakes
- Theater Maze
- Tube Maze

The single family dwelling scenario is depicted in Fig. 4. This scenario simulates a partially-collapsed single family home due to an earthquake. Entrances to the building and doorways between rooms are compromised, with many either fully or partially blocked. The floors are scattered with debris from the concrete structure and furniture. The ceiling is mostly collapsed in one room and there is a large breach in the floor in another room. There is also a basement accessible from the outside through a long set of stairs to a welled exit. Robots are deployed into the building to identify victims, hazards, and all entrances and exits to inform responders of the situation.

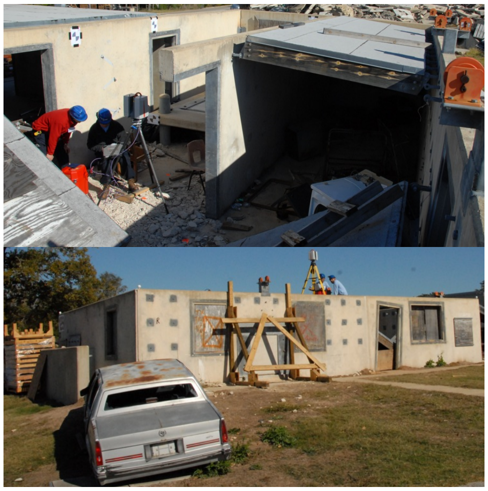


Figure 4: Single family dwelling training scenario simulating a partially collapsed home. The top picture shows the area inside one of the rooms of the building and the bottom picture shows the building from the outside.

The house of pancakes scenario is depicted in Fig. 5. This scenario mimics a partially collapsed concrete building of unknown use. The roof structure is collapsed on one side of the building causing the roof to angle downward such that it is almost in contact with the ground. The interior of the building contains various wood and concrete structures as well as office desks and tables. Robots are deployed into the building to search for victims and to map the environment.

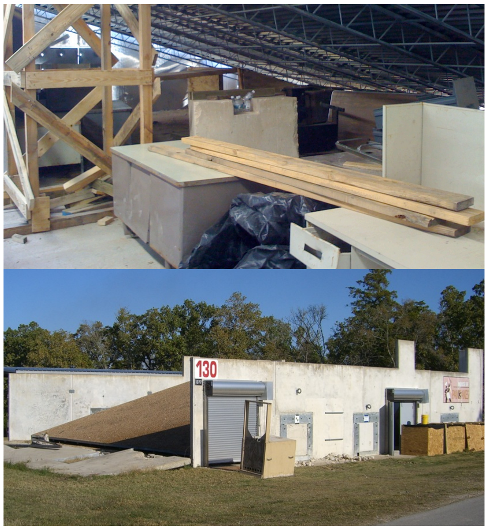


Figure 5: House of pancakes training scenario simulating a partially collapsed concrete building. The top picture shows the main area inside with the collapsed sloped roof and the bottom picture shows the building from the outside.

The theater maze test method is depicted in Fig. 6. This environment tests the ability of the robot and its operator to fully explore a complex unknown environment for victims and identify standard hazardous material placards without getting lost. The maze consists of rolling wooden floor planks on a slope and 2.44 m tall wooden walls.

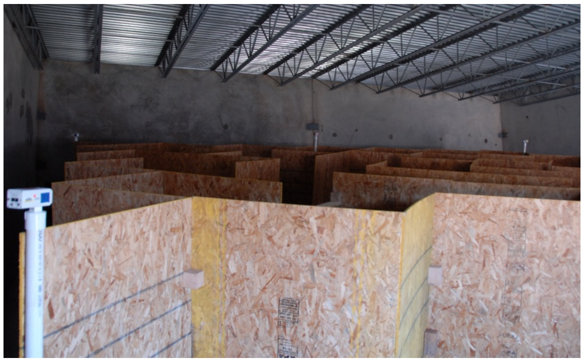


Figure 6: Theater maze test method which tests the ability of a robot to navigate a complex environment without getting lost.

The tube maze test method is depicted in Fig. 7. This

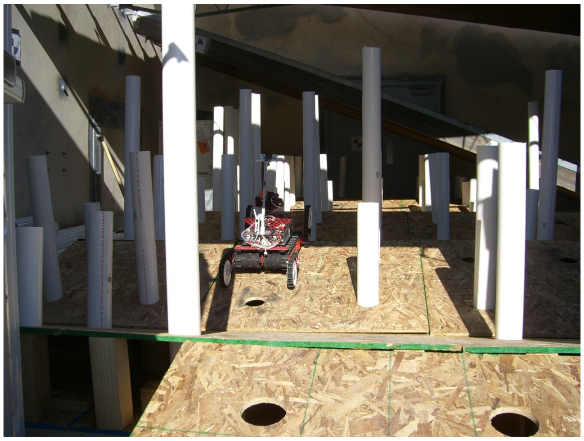


Figure 7: Tube maze test method which tests the mapping ability of US&R robots in an environment littered with occlusions.

environment tests the mapping and localization ability of robots in an environment full of occlusions. The flooring is made of angled sheets of wood. Rising up from the flooring are PVC pipes of varying height that provide occlusions for this environment.

4. DATA SETS

The 3D image data was collected using commercial laser scanners. A laser scanner is a 3D imaging device that uses a laser to measure the distance to an object. The laser beam is scanned both horizontally and vertically over time to image the operator-designated field of view. The distance, azimuth, and elevation information collected from each measurement in the scan is used to create high-resolution point clouds containing hundreds of thousands of points for a single scan.

Two different laser scanners were used in the two data collection efforts. The data for collection effort one was collected using a pulse-based time-of-flight laser scanner. The manufacturer specifies a range uncertainty of 7 mm and a point uncertainty of 12 mm at 100 m range for this instrument. The data for collection effort two was collected using a phase-based time-of-flight laser scanner. For this instrument, the manufacturer specifies a range uncertainty $\leq$ 6 mm for ranges up to 50 m. A point uncertainty was not specified for this instrument.

4.1 Sample Data and Metrics

Figures 8 and 9 show screen captures of scenes generated in point cloud software for the tube maze test method. Each point is colored based on the intensity of the laser return. Within the software, camera viewpoints can be changed to examine the 3D data from multiple viewing angles and measurements such as point-to-point distance can be readily determined.

Figure 8 shows an elevated view of the point cloud data for a single scan of the tube maze. Since laser scanners are line-of-sight instruments, a single scan is unable to capture the entire environment when there are occlusions in the scene. Occlusions cause ”shadows” of missing data where the laser scanner cannot sense. By design, the tube maze contains many occlusions and areas of missing data (shown in black) are prevalent in the individual scans.

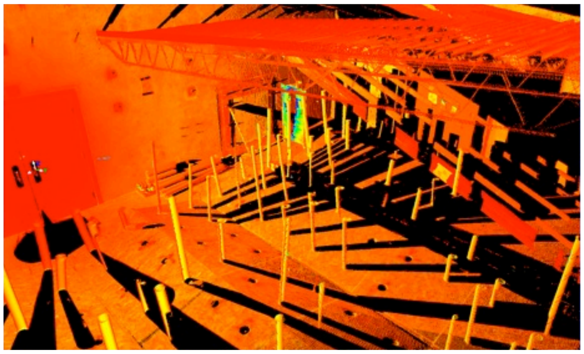


Figure 8: An elevated view of the point cloud data for a single scan of the tube maze test method. Occlusions cause ”shadows” of missing data.

To fill in the missing data, scans are taken from multiple locations around the scene. Individual scans are then merged through a process called registration to create complete point clouds of the scenes. While some of the scans required manual registration, most of the scans were registered using stationary targets placed in the scene to provide common points of reference to register the scans. The data was registered and segmented using commercial software tools. Figure 9 shows the complete registered point cloud data set for the tube maze test method from a similar viewpoint.

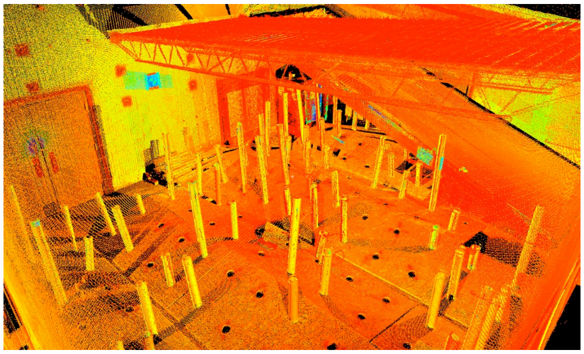


Figure 9: An elevated view of the fully registered point cloud data for the tube maze test method.

The number of points collected for each scenario is given in Table 1 and the number of scans is given in Table 2.

4.2 Data Availability

All of the 3D image data outlined in this paper is available free of charge to the public. Please contact the authors to obtain any data of interest. In the near future, the data and documentation of the file formats will be made available through the NIST website.

5. FUTURE WORK

Stepfield pallets are a fabricated and repeatable terrain for evaluating robot mobility [3]. As a first step towards developing terrain traversibility metrics, NIST researchers will use the ground truth data presented in this paper to investigate the design of a multi-unit stepfield approximation of

Table 1: The number of points for each scenario and test method data set from collection efforts one and two.

Collection	Scenario/Test Method	# Points (Millions)
1	Concrete Rubble Pile	5.77
	Wood Rubble Pile	7.75
	Passenger Trains	4.87
2	Single Family Dwelling	848.35
	House of Pancakes	2550.60
	Theater Maze	12.62
	Tube Maze	296.90

Table 2: The number of scans for each scenario and test method data set from collection efforts one and two.

Collection	Scenario/Test Method	# Scans
1	Concrete Rubble Pile	45
	Wood Rubble Pile	23
	Passenger Trains	41
2	Single Family Dwelling	26
	House of Pancakes	29
	Theater Maze	2
	Tube Maze	10

a representative segment of one of the Disaster City rubble piles. If this is achieved, the existing mobility metrics captured for the stepfields can be applied to predict how well a given mobility platform will perform in the rubble pile scenario and more generally, any terrain that can be modeled in this fashion.

The ground truth data collected in this work will also be used to explore methods for evaluating the quality of maps generated by the US&R robots which traversed the same scenario environments. While there is currently some work being investigated for evaluating 2D maps, there is little work being done for 3D maps.

Finally, the data will be used to support the modeling of the Disaster City scenarios for use in virtual training and testing environments such as USARSim.

6. ACKNOWLEDGMENTS

The authors would like to thank Adam Jacoff and Christopher Scrapper whose support made this work possible. Also, a hearty thanks to Gerry Cheok, Kamel Saidi, and Nicolas Philippon for their tireless work in helping plan and execute these data collection efforts.

7. REFERENCES

[1] ASTM. *ASTM E2544 - 09b Standard Terminology for Three-Dimensional (3D) Imaging Systems.* ASTM, 2009.

[2] S. Balakirsky, F. M. Proctor, C. J. Scrapper, and T. R. Kramer. An integrated control and simulation environment for mobile robot software development. In *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conferences*, August 2008.

[3] A. Jacoff, A. Downs, A. Virts, and E. Messina. Stepfield pallets: Repeatable terrain for evaluating robot mobility. In *Proceedings of the 2008 Performance Metrics for Intelligent Systems (PerMIS) Workshop*, 2008.

[4] A. S. Jacoff and E. R. Messina. DHS/NIST Response Robot Evaluation Exercises. In *Proceedings of the 2006 IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, January 2007.

[5] A. S. Jacoff and E. R. Messina. Urban search and rescue robot performance standards: Progress update. In *Proceedings of the 2007 SPIE Defense and Security Conference*, June 2007.

Performance Measurements for Evaluating Static and Dynamic Multiple Human Detection and Tracking Systems in Unstructured Environments

Barry Bodt,
Richard Camden
Army Research Laboratory
{babodt,rcamden}@arl.army.mil

Harry Scott, Adam Jacoff, Tsai Hong, Tommy Chang, Rick Norcross, Tony Downs, and Ann Virts
National Institute of Standards and Technology
{harry.scott, adam.jacoff, tsai.hong, tommy.chang, rick.norcross, tony.downs, ann.virts}@nist.gov

ABSTRACT

The Army Research Laboratory (ARL) Robotics Collaborative Technology Alliance (CTA) conducted an assessment and evaluation of multiple algorithms for real-time detection of pedestrians in Laser Detection and Ranging (LADAR) and video sensor data taken from a moving platform. The algorithms were developed by Robotics CTA members and then assessed in field experiments jointly conducted by the National Institute of Standards and Technology (NIST) and ARL. A robust, accurate and independent pedestrian tracking system was developed to provide ground truth. The ground truth was used to evaluate the CTA member algorithms for uncertainty and error in their results. A real-time display system was used to provide early detection of errors in data collection.

Categories and Subject Descriptors

B8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; C.4 [**Performance of Systems**]: Performance attributes.

General Terms

Tracking, Algorithms, Performance, Measurement, Experimentation

This paper is authored by employees of the United States Government and is in the public domain.
PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA. ACM 978-1-60558-747-9/09/09

Keywords

Unmanned ground vehicle, experimental design, ground truth, pedestrian tracking, metrics, perception, performance evaluation

1. INTRODUCTION

The ARL Robotics Collaborative Technology Alliance (CTA) conducted an assessment and evaluation of multiple algorithms for real-time detection of pedestrians in Laser Detection and Ranging (LADAR) and video sensor data taken from a moving platform in January 2009. In the assessment, the robot vehicle equipped with two pairs of stereo cameras, two sets of General Dynamics Robotic Systems (GDRS) LADAR and two sets of SICK[1] lasers was driven by an operator through a straight route of approximately 240 m containing various configurations of eight moving pedestrians, four mannequins, four barrels, four cones, two trucks, two crates, seven tripods and trees. In addition to the complexity of the environments, the variables included multiple robot vehicle speeds (30 km/h or 15 km/h) and pedestrian speeds (1.5 m/s or 3.0 m/s). The environment was intended to provide some Military Operations in Urban Terrain, or MOUT, characteristics.

The objective of the experiment was to capture the data necessary to evaluate the performance of each CTA team's algorithm, to provide data to support further development of algorithms, and to produce performance analyses based on the captured data to support obstacle avoidance planning. An Ultra

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST nor does it imply that the materials or equipment identified are necessarily the best for the purpose.

WideBand (UWB) system [1] employed by the National Institute of Standards and Technology (NIST) provided position tracking (≈20 cm uncertainty) of the moving and stationary humans, the robot vehicle, and other objects. Improved performance of the CTA tracking and recognition algorithms has called for improvements in the ground truth solution. Processing techniques were developed and implemented to produce higher quality tracking solutions than those provided by the raw data captured by the ultra wideband system. To address this, we developed a robust filter algorithm. To improve analysis of the performance of the CTA tracking systems, we also developed a temporally consistent algorithm for finding the correspondence between the ground truth data and the CTA tracking data. In addition, a display system was implemented to provide early detection of errors in data collection and to assist in data analysis.

The paper is organized as follows: Section 2 presents a detailed description of the solution based on the UWB system for capturing the ground truth data. Section 3 introduces the filter and interpolation algorithms for improving the quality of the UWB data. Section 4 describes the visualization system for providing early detection of errors. Section 5 presents the correspondence algorithm for data analysis. Sections 6 and 7 present the performance metrics and analysis for evaluating the CTA algorithms. Finally, Section 8 provides a summary and conclusion.

2. GROUND-TRUTH REFERENCE SYSTEM

2.1 Ground Truth Setup

NIST researchers have been working with an asset tracking system employing ultra wideband technology to capture 2-D location and path data for robots, vehicles, and personnel operating within scenarios set up to evaluate robotic perception systems. The goal is to capture quantitative performance data referenced to ground truth positions and time to help compare and improve sensors and algorithms in both indoor and outdoor scenarios.

The tracking system uses state of the art ultra wideband radio receivers posted around the perimeter of the scenario to track multiple static and dynamic targets with badge-size transmitters (see Figure 1). It works in open outdoor areas and indoor areas through certain types of walls, though overall accuracy can vary. NIST has performed system characterization tests in ideal conditions to determine the best possible 2D accuracy of the system, which is approximately 15 cm. We have used it to track vehicles and personnel throughout an area over 80 000 square meters with an average accuracy of approximately 20 cm and an update rate of approximately 50 Hz, which is sufficient for tracking vehicles at highway speeds. We have also used it to track robots through random mazes with plywood walls (non-line-of-sight) achieving similar accuracies. We have not been successful tracking through concrete walls, but have used additional receivers in hallways to compensate during indoor building deployments. The total number of dynamic and static transmitter tags used simultaneously thus far has been approximately 15 and 30 respectively for marking obstacles and known fiducial points to check accuracy. Setup time for a new site takes about 5 days. Returning to a previously setup site takes approximately two days for calibration prior to testing.

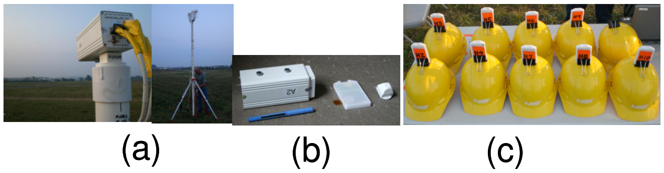


Figure 1. (a) shows a receiver deployed in the field atop a mast centered over a known fiducial marker. (b) shows the asset tracking system components, ultra-wideband radio frequency receiver (shown with integrated high-gain antenna), 1 W transmitter tag, and 30 mW transmitter tag. c) shows Several badge tags attached to helmets to track personnel in the scenario. Typically two tags are placed on moving vehicles to identify orientation.

Figure 2 shows a plot of the tracking results for a ground truth system coverage and accuracy test on the January course configured at NIST. Green and orange plots show the vehicle path and the other plots show pedestrian tracks.

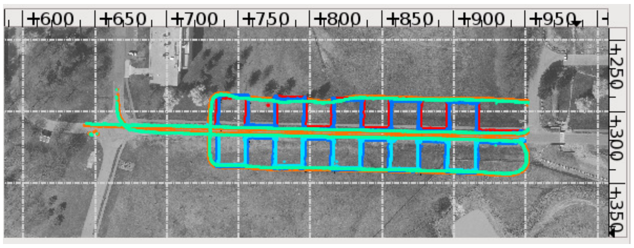


Figure 2. A calibration run with two transmitter tags mounted on a vehicle and two tags on each of two pedestrians to check coverage.

2.2 Filter and Interpolation Algorithms for the Ground-Truth Data.

The goal of the filter process is to remove outlier and error measurements from the ground-truth data. We identify outliers based on the maximum plausible speed of the tag. A polynomial least-squares algorithm filters the remaining data points. We then fit a spline through the filtered points to identify the tag's position as a function of time. We interpolate the trimmed, filtered, and splined data at timestamps obtained from the CTA performers' data. This interpolated ground-truth is later used to establish temporal constraints for correspondence.

The UWB position data contain anomalies that, while generally minor, diminish the usefulness of the data in subsequent evaluations and displays. The filter combines previous and subsequent readings to remove anomalies and identify a more accurate and timely position. For example, Figure 3 below is the UWB position data for tag A01D of Run3. The data show two significant anomalies: a gap in the center area and outlier points away from and along the track.

The filter has three components: trim, filter, and spline. The green dots represent the raw data that were trimmed as outliers. The red dots are the remaining raw data points. The white line is the result of the filter. And the blue points are the positions at the CTA timestamps based on a spline fit of the filtered points.

The filter's trim component removes outlier data points. Physical constraints limit the distance that an UWB tag can move between readings. The trim component computes the velocity between the current point and the last good point. The filter trims the current point when the velocity between these points is excessive. The filter checks subsequent points for a point that represents a reasonable velocity. The filter then uses that point as the last good point for subsequent evaluations. The filter passes the trimmed position list to the window component (see red in Figure 3).

The filter component is based on the Savitsky-Golay algorithm [2]. Savitsky applies a polynomial least-squares fit to a set of points before and after the current point. The filtered value is the sum of the products of the points with an array of coefficients determined by the order of the algorithm (generally 3) and the size of the point set. The Savitsky algorithm relies on evenly spaced data. The gaps in the trimmed data would cause the Savitsky algorithm to inappropriately shift the data points. To compensate, our algorithm fills gaps in the trimmed data with linearly interpolated data points before applying the Savitsky-Golay algorithm. Our algorithm discards the fill points prior to passing the data onto the spline component (see the white in Figure 3).

The filter's spline component is based on a cubic Hermite spline. The spline component identifies positions at a time of interest rather than at the time of data collection and allows researchers to determine the position of the UWB tag at times provided by the CTA systems (see the blue in Figure 3).

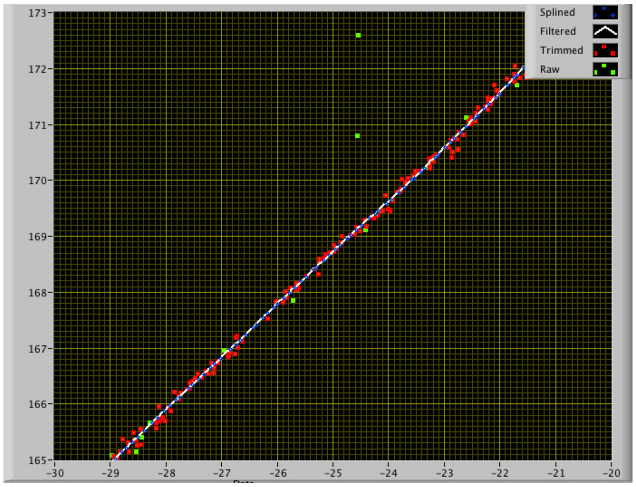


Figure 3. Green is the raw data. Red is the trimmed data. White is filtered data. Blue is the interpolation data

3. CTA REAL-TIME HUMAN DETECTION AND TRACKING ALGORITHMS

In this experiment, six algorithms were included from the CTA. Five use LADAR sensing and one uses a vision system to provide data for the algorithms. During each algorithm cycle, the algorithm reports information about the detected humans. The report includes the number of detections, their locations, strength of detections, the time of detection, as well as vehicle status such as its location, speed, orientation, etc. All detections from the same algorithm cycle have the same detection time. The reports are collected and saved into files, one per algorithm.

Reporting rate varies from algorithm to algorithm and may not be fixed within the algorithm itself. For example, an algorithm's cycle time may increase when the number of detections increases.

In general, there are two data sets: ground-truth data and detection data. Detection data are the locations and detection times of all humans reported by a CTA algorithm, whereas ground-truth data are the corresponding UWB locations for the same time.

Both the detection data and the ground-truth data are independently grouped with unique identifications (ID). For the ground-truth data, the group IDs are also referred to as the "tag ID". Different tag IDs always refer to different humans or physical objects.

For the detection data, the group IDs are referred to as "tracking ID". With perfect CTA system detection and tracking performance, the number of tracking IDs would be the same as the number of tag IDs. In reality, the detection data can have more than one tracking ID for the same human due to occlusion or imperfect tracking capability of the algorithms.

4. DATA VISUALIZATION FOR EARLY DETECTION OF ERRORS IN DATA COLLECTION.

Data visualization is important for verifying the integrity of both the ground-truth data and the outputs of the CTA algorithms prior to, and during, the data collection. Bad data could arise due to sensor malfunction or unforeseen circumstances prior to or during the data collection. Since data collection is expensive, time consuming, and labor intensive, it is advantageous to detect bad data as soon as possible and prevent waste of resources.

A software-based interactive viewer was developed for this purpose. The viewer uses various open source libraries and runs natively on Linux, Windows and Mac OS X. Figure 4 shows a typical screen-shot of the viewer displaying both the detection data from a CTA algorithm and the corresponding ground-truth data. An individual entity can be toggled on or off by clicking on its tag ID or tracking ID.

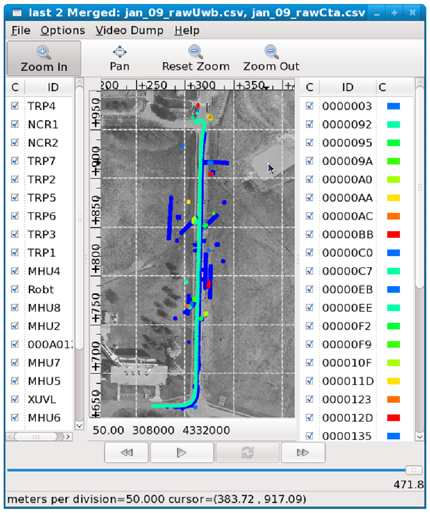


Figure 4. CTAviewer screenshot showing both detection data and ground-truth. The left panel lists the tag ID and the right panel lists the tracking ID. An individual entity can be toggled on/off by its ID.

Figure 5 and Figure 6 are examples of bad data. The plots show the locations of two ground-truth tags mounted on a moving vehicle. The two tags were separated by about 1 meter and the vehicle drove at a constant speed.

The viewer software allows us to quickly view and evaluate the data immediately after the each run and investigate the cause of any anomalies. Problems in ground-truth can often be eliminated by placing more UWB receivers in the problematic areas.

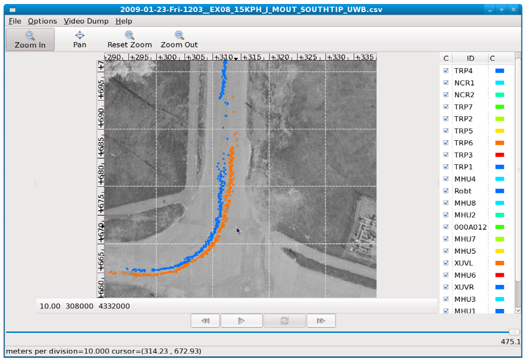


Figure 5. Correctable gaps in the ground-truth data.

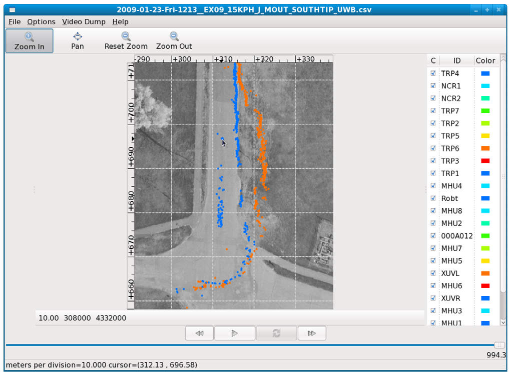


Figure 6. Uncorrectable severe distortion and gaps in the ground-truth data.

5. MAP BETWEEN GROUND-TRUTH AND DETECTION

All CTA teams output their results with time stamps which are used to synchronize to the UWB ground-truth data. Since each team has different output rates, the linear interpolation described in Section 2.2 is used to handle the different rates. All timestamps in the data collection systems come indirectly from a common clock source via a Network Time Protocol (NTP) server. All systems synchronize to the NTP server at the beginning of each run. A run lasts about 1 minute. This allows our data collection computers to stay synchronized to each other within 20 milliseconds.

All CTA algorithms output their results in a standard format, and are stored in Comma Separated Values (CSV) for viewing using the viewer described in section 3.

The requirements for the performance evaluation of the CTA systems include:

1. Timestamp correspondence between ground-truth and detection.

2. Object/human correspondences between ground-truth and detection.

3. Definition and computation of metrics and measurements for performance evaluation.

Establishing time correspondence between the ground truth and the detection system is important for resolving ambiguity that involves time. Finding a mapping between the ground truth objects and the objects detected by the CTA algorithms is crucial for evaluating the algorithms' performance. In Figure 7 using a nearest neighbor criterion, inside the red circle, the blue star T_3 will correspond to the yellow T_2 circle since the distance is less than the distance to the blue T_3 circle. When time correspondence is established, the star T_3 ground-truth will correspond to the ground truth represented by the blue T_3 circle. The time correspondence algorithm will be described in section 5.1. Before defining the metrics, it is necessary to have a good way of assigning detected objects to ground truth. The detail of the correspondence algorithm will be described in Section 5.2. The assessment of the performance of the CTA tracking systems required several measurements. These measurements will be described in Section 5.3.

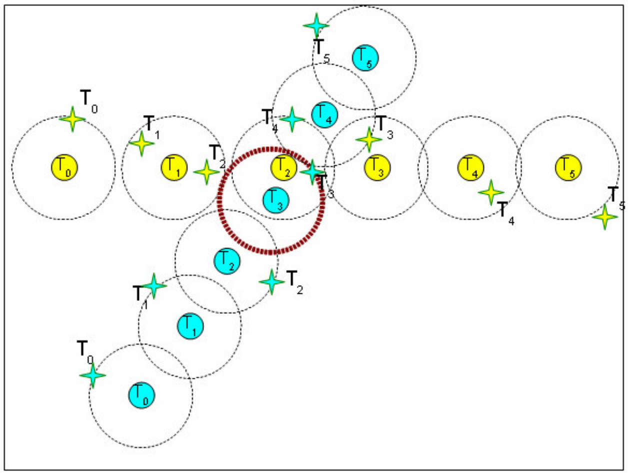


Figure 7. The CTA data are represented by star shapes and the ground-truth data are represented by shaded circles. T_n represents the time n. It is sufficient to use time to match the closest detection ground-truth pair. The outer circle around the ground-truth data indicates the threshold radius used for establishing a spatial constraint.

5.1 Establish Time Correspondences between Ground-Truth and Detection Tracking.

After the filtering and interpolation process, all ground-truth data are interpolated at timestamps compiled from all the detection data. Two matching files are generated for each team - one containing the ground truth and the other containing the detection data. These two matching files have the same number of entries. Each entry contains a timestamp and information about object locations detected at, or interpolated to, that timestamp. Since the timestamps are matched, there is a one-to-one correspondence among the timestamps in the entries between the two files.

5.2 Establish Object/Human Correspondences between Ground Truth and Detection Tracking

Previously [3], correspondence was determined solely based on spatial constraints. One such constraint is the distance between the ground-truth and the detection data. Although spatial constraints are essential, they alone can not resolve ambiguities that arise when close data are taken at different times. Such ambiguity and an unnecessary spatial search can be avoided when we take the temporal consistency into account

Several map correspondence algorithms [4][5][6][7] were investigated. The correspondence algorithm, adopted by Classifications of Events, Activities and Relationships (CLEAR)[8] evaluation workshop group, was implemented. Using matched-pair data from Section 5.1, the algorithm computed the averaged error distances over the cluster with respect to each object. Clusters were associated with ground-truth objects based on a minimum average distance subject to meeting a 3 m proximity threshold based on our experiments. In addition, we used velocity to differentiate stationary from moving objects. Clusters were then labeled as a human (moving), mannequin (stationary human), misclassification objects (moving or stationary), or false positive. The results of this correspondence procedure will be used for computing several measurements for the analysis of the human detection in the following section.

5.3 Post Process the Data Acquisition files for Human Detection Analysis

In order to analyze the CTA tracking algorithms correctly and accurately, post processing of the data is necessary. CTA algorithms differed in cycle time ranging from 7 Hz to 20 Hz. At the end of each algorithm cycle, each algorithm reported detection information such as positions and velocities of the humans. The underlying assumptions for the outputs of the algorithms included the following:

- Only obstacles seen and classified as human were reported.
- Unique identification numbers were assigned to individual algorithm detections within a run.
- Algorithms demonstrated tracking of an individual by maintaining the same ID in successive frames.
- Algorithms also reported velocity of the detected humans.

Since we only instrumented the ground-truth data in the 300 m x 150 m test area, all detections are excluded if they occurred outside the test area. The correspondence algorithm described in Section 5.2 found the correspondence between the detections and the ground-truth based on location and time stamp. Detections were compared with all the ground-truth objects on the course. Absolute error distances were computed, summed, and averaged over the cluster with respect to each course object. The absolute velocity error between detection and ground truth objects was also computed and averaged over the cluster. Clusters were associated with a ground-truth object based on minimum average distance subject to meeting a 3 m proximity threshold. Using velocity to establish stationary/moving objects, clusters were then classified as a human, mannequin, misclassification, or false positive. Other values were reported in the post processing. For example, reports included the distance from the moving vehicle at the time of first detection for individual detections within the common ID cluster, the shortest distances and velocities, and dispersion measures for distance and velocity.

6. PERFORMANCE METRICS

Post processing of the data above results in a spreadsheet for each algorithm with metrics for analysis. A record is formed for each algorithm-reported human. Each algorithm assigns an identifier to an entity on the course classified by the algorithm to be a human. All information related to that algorithm identification is condensed to a single record. This record may hold information from many cycles of the algorithm. Post processing determines whether that entity is, in truth, a human or mannequin (true positive), another known course entity not human or mannequin (misclassification), or an unknown course feature with no associated ground-truth (false positive). Distinctions are also made between moving and stationary entities and various classes of nonhuman entities (e.g., barrels, cones, crates). Field notes describe test conditions under which the data were collected, absolute and relative positioning of the robot platform and detected entities recorded at the time detections first occurred for an identification, time and cycle number indicators of the persistence of detection, or the accuracy of the algorithm classification decision.

7. EXPERIMENTS AND PERFORMANCE EVALUATION

The purpose of this section is to outline the experiment and to illustrate the importance of the ground truth system in assessment of the algorithms. A complete analysis is not given. The principal experiment consisted of thirty-two runs conducted at the south end of Center Drive on the NIST campus (Figure 9). An autonomous vehicle platform, with sensors and algorithms on board as discussed above, was driven south to north over an approximately 240 m run. Scripted scenes with human motion, mannequins, and course clutter were sensed and interpreted and reported by the algorithms in real time. Eight humans were present in each run, four to either side of the road. Four moved in a manner parallel to the road, three at 45 degrees toward the road, and one perpendicular toward the road. Three parallel runs were receding from the platform and one was approaching. Among the 45 degree runs, all were approaching the road, but only the run to the right was approaching the vehicle.

Movements of the humans were choreographed and timed to ensure that regardless of test conditions the scene sensed remained consistent across runs. See Figure 8. All humans were upright in the principal experiment. Excursion runs not reported here explored other postures and group movement patterns.

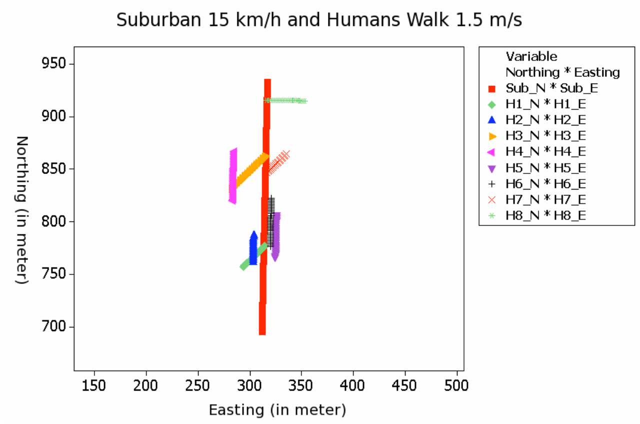


Figure 8. Human paths relative to platform route. The units are in meters.

Test conditions were formed based on three factors: platform speed, human speed and course clutter. The platform was driven at either 15 km/h or 30 km/h, humans moved at 1.5 m/s or 3.0 m/s, and the course was cluttered to approximate MOUT complexity or was open except for the human movers. The test conditions were allotted equally in accordance with a 2^3 factorial design with 4 replications per condition. Under the MOUT conditions only, 8 mannequins, 4 barrels, 4 cones, 2 crates, and 2 trucks were included on the course. Seven NIST tripods were on the course for all 32 runs. Figure 9 shows a view of the course during a MOUT run.

The assessment of algorithms focuses on the questions of what an algorithm saw, when it saw it, and how long the sighting persisted. These questions will be pursued for each algorithm and in the context of the experimental conditions under which the data were collected. The ground-truth system allows definitive answers to these questions. We share some preliminary high-level results to illustrate performance.

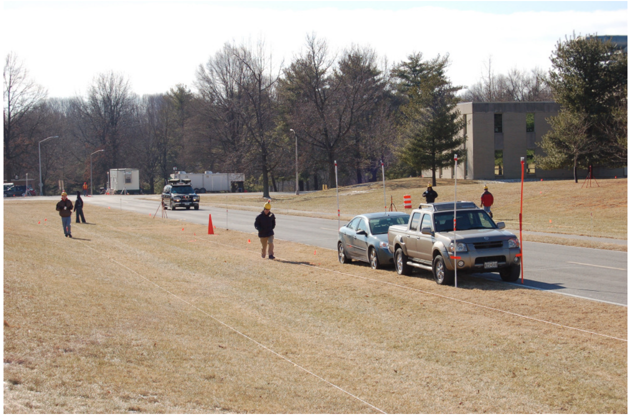


Figure 9. Right side of course during a MOUT run.

Table 1 summarizes the performance of the six algorithms in terms of detections, misclassifications, and false positives over the complete set of 32 runs. Entries are percentages except for the false positive entries, which report the number per run. Note that true positives are in bold, and false positives are in italics. All other entries show the algorithm misclassification of other course entities as human. At a high level, this table addresses what was seen.

Table 1. Algorithm performance expressed in terms of the percentage of course entities detected and the number of false positives per run.

	CTA Algorithm					
Object Type	Alg 1	Alg 2	Alg 3	Alg 4	Alg 5	Alg 6
Human(%)	**97.3**	**90.8**	**98.4**	**98.0**	**89.5**	**85.7**
Mann.(%)	**10.2**	**-**	**97.7**	**98.4**	**91.4**	**62.5**
Cones(%)	0.0	-	4.7	0.0	65.6	0.0
Barrels(%)	14.1	-	54.7	70.3	89.1	0.0
Crates(%)	46.9	-	100.0	90.6	100.0	50.0
Trucks(%)	25.0	-	100.0	25.0	100.0	75.0
Tripods(%)	1.3	46.7	53.6	60.7	58.9	29.8
False Positives	*29.8*	*77.9*	*155*	*37.3*	*29.8*	*1.3*

Performance varies widely across algorithms. While some demonstrate a high probability of detection, misclassification of other course entities is clearly a problem. Moreover, the number of false positives recorded, if not addressed, ultimately would provide a greater challenge for dynamic planning in an autonomous mode.

In Figure 10, a boxplot (with mean) is shown for the distance between the platform and the target entity at the time of first detection. This distance is as perceived by the algorithm, but generally this does not vary greatly with the actual ground truth. Detections (green), misclassifications (yellow), and false positives (red) are shown. There are differences according to the type of obstacle. Tripods and trucks (Trks), for example, tend to be recognized in the neighborhood of 30 m away; whereas humans are detected on average at more than 50 m away. Confidence in this graph is based on the ground-truth system. A similar graph exists with actual distances based on the ground-truth, but a distance for false positives requires the algorithm-produced values. In this fashion, the question of when entities were seen is addressed.

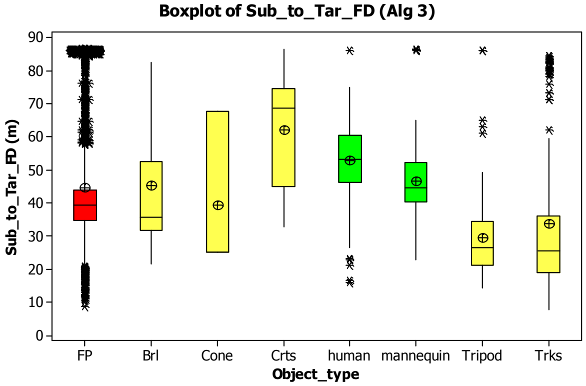


Figure 10. Boxplots of distance from platform to targets detected by the algorithms for different object types.

Figure 11 shows a boxplot for the duration of time all entities of a certain type were tracked. Ideally, humans and mannequins would be tracked persistently; whereas, other entities would not.

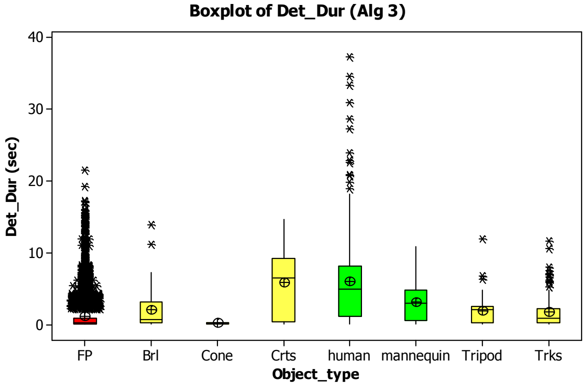


Figure 11. Boxplots of duration of tracking for different object types.

From Table 1 we learn that high detection rates are accompanied by higher than desired misclassification rates and numbers of false positives. The intent of examining data in Figure 11 is to determine if persistent tracking requirements might greatly reduce false alarms and misclassifications while retaining a high level of detection. In this case, at least 75 % of the false positives fall below the 25th percentile for humans and mannequins detected, suggesting persistent tracking may reduce false positives. However, tracking of misclassified entities would not be greatly influenced.

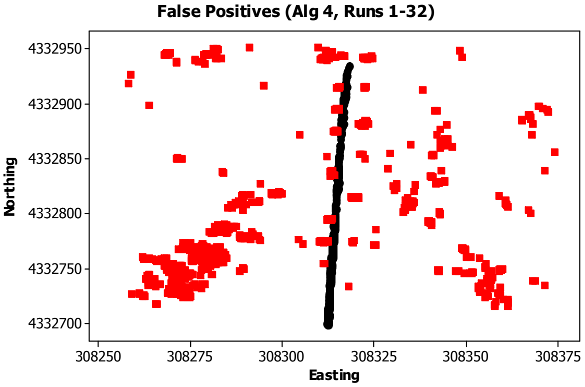


Figure 12. Scatterplot of false positive locations for Alg4.

Ground-truth allows a definitive decision on false positives. Figure 12 shows the false positives recorded for ALG4 over the 32 runs of the experiment. The path of the vehicle (black) and the false positive locations (red) are shown. As part of the analysis, the cause of false positives (e.g., bushes, trees, high grass) will be pursued.

One of the most advantageous features in the measurement technology employed here is the time sequenced display of detection. For individual runs, it is necessary to drill down into the data to investigate anomalies more carefully. A static display in Figure 13 shows a run for ALG4. Labels over the points for the path of the vehicle indicate when a specific moving human was detected during the run. The moving human is also plotted. In this instance, there are replicates for some of the moving humans (e.g., MUH4). This tells us that multiple unique identifications were assigned to this human. For some reason, the algorithm judged the human to be a different entity at different time. One possibility is occlusion, as might be the case here when MUH4 was obscured from view by crates during a MOUT run. Although informative, the static display does not reveal the same detail as movies of the run as it unfolds. The CTA Viewer illustrated in Figure 4 is critical to detailed analysis.

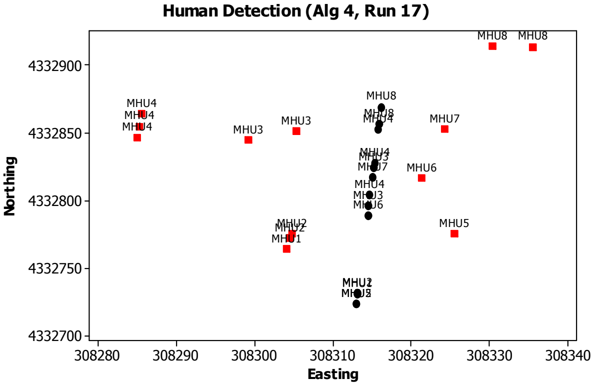


Figure 13. Human detection for run 17 using ALG4.

8. CONCLUSION

We presented details of several components of a system for determining performance of sensors and perception algorithms tasked with detection, tracking, and classification of moving and fixed objects, including pedestrians, around a moving robot vehicle. We presented a filter algorithm developed to improve ground-truth data for analysis. In addition, we developed a group-based correspondence between ground-truth and detection for data analysis and performance evaluation.

From an analysis perspective, the advances in measurement technology of good ground truth data improve the assessment process markedly. The ground truth precision provides an objective evaluation of the results reported by the algorithms. It makes possible the exact tracking of moving entities on the course, essential given the planned assessment of the “detection and tracking” purposes of the algorithms. This was previously not possible. The CTA viewer has proven to not only be a useful tool in visual analytics, but has also provided an instant check during the conduct of the experiment as to whether or not data are being collected and whether systems are in good calibration.

9. ACKNOWLEDGMENTS

The authors would like to thank the CTA partners for their research and willingness to conform to the reporting format, and to thank Jeff Jones and his team at General Dynamic Robotics Systems (GDRS) in Westminster for their help with data collection.

10. REFERENCES

[1] Robert Fontana, Recent System Applications of Short-Pulse Ultra-Wideband (UWB) Technology (Invited Paper), IEEE Microwave Theory & Tech., Vol. 52, No. 9, Sept. 2004.

[2] Abraham Savitzky and Marcel Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures". Analytical Chemistry 36 (8): 1627–1639, 1964.

[3] Barry Bodt and Rickard Camden, “Detecting and Tracking Moving Humans form a Moving Vehicle,”, SPIE, Orlando, Florida, 2008.

[4] Keni Bernardin, A. Elbs, Rainer Stiefelhagen, “Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment”. The Sixth IEEE International Workshop on Visual Surveillance, VS 2006, Graz, Austria, 2006.

[5] Kevin Smith, Sileye Ba, Jean-Marc Odobez and Daniel Gatica-Perez, “Evaluating Multi-Object Tracking”, Workshop on Empirical Evaluation Methods in Computer Vision (EEMCVF) San Diego, CA, 2005.

[6] Sarma Pingali and Jakub Segen, "Performance Evaluation of People Tracking Systems," wacv, pp.33, Third IEEE Workshop on Applications of Computer Vision (WACV '96), 1996

[7] Faisal Bashir and Fatih Porikli., "Performance Evaluation of Object Detection and Tracking Systems", IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), June 2006.

[8] Keni Bernardin, “ CLEAR 2007 Evaluation Plan 3D Person Tracking Task,”, Classification of Event, Activities and Relationships Evaluation and Workshop (CLEAR) 2007.

[9] Richard Camden and Barry Bodt, “Safe Operations Experiment Report,” ARL-TR-3773, U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, April 2006.

[10] Elias Rigas, Barry Bodt and Richard Camden, “Detection, tracking, and avoidance of moving objects from a moving autonomous vehicle,” Proc. SPIE 6561 (2007).

Mathematical Metrology for Evaluating a 6DOF Visual Servoing System

Mili Shah
Loyola Univ. Maryland
Baltimore, Maryland
mishah@loyola.edu

Tommy Chang,Tsai Hong
National Inst. of Standards and Technology
Gaithersburg, Maryland
{tchang,hongt}@nist.gov

Roger Eastman
Loyola Univ. Maryland
Baltimore, Maryland
reastman@loyola.edu

ABSTRACT

In this paper we develop a homogeneous matrix transformation to fit two streams of dynamic six degree of freedom (6DOF) data for evaluating perception systems using ground truth. In particular, we compare object position and orientation results from a 6DOF laser tracker that we consider to be ground truth with results from a real-time visual servoing system from the Purdue Robot Vision Lab. A problem that arises when comparing these two data streams is that they are not necessarily in the same coordinate system. Therefore, a method to transform one coordinate system to the other is needed. We solve this problem by developing an optimization problem that minimizes the space between each coordinate system. In other words, we construct a rotation and translation which best transforms one coordinate space to the other.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Performance attributes; B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; G.1.6 [**Optimization**]: Global optimization; I.4.8 [**Scene Analysis**]: Motion, Tracking; I.5.4 [**Applications**]: Computer Vision

General Terms

Computer Vision, Laser Tracker, Dynamic 6DOF metrology, Performance Evaluation

1. INTRODUCTION

In previous work [2] we reported on experiments in the evaluation of the performance of a real-time visual servoing system using a highly accurate, dynamic, six degree of freedom (6DOF) laser tracker. The purpose of the experiments was to demonstrate a method for evaluating real-time 6DOF dimensional measurements of an object or assembly component under moderately constrained motion. By taking geometrically calibrated, time-synchronized data streams simultaneously from the 6DOF servoing sensor system and the laser tracker, the 6DOF system data can be evaluated against the laser data serving as conventional ground truth. In this paper we report on improved techniques for post-experiment and geometric calibration and evaluation of the experimental data.

Reliable, accurate real-time systems for 6DOF perception would have applications in advanced manufacturing robotics and automation, as they would enable greater interaction with objects in motion and more flexible robotic workcells. However, despite considerable advances in real-time vision and in laboratory demonstrations [7,16,17], these systems have not yet been widely commercialized and this would be assisted by reference metrology systems for empirical performance evaluation. Reference systems would include a standard sensor system for ground truth along with appropriate metrics for the comparison of test systems with the reference system. Standards and test procedures for dimensional metrology are well-established and highly accurate for static measurements, with coordinate measuring machines and laser trackers giving position measurements to microns. However, the theory, technology, and test procedures are not well established for dynamic dimensional measurements in uncontrolled environments.

To assist in establishing these test procedures, the questions addressed in this work focus on calibrating and comparing two 6DOF data streams. We assume the two vector data streams include position as X, Y, Z, pose as roll, pitch and yaw, and that the two data streams have been time-synchonized so we have correspondence between individual vectors in each data stream. But, we do not assume accurate geometric calibration of coordinate systems between the two data streams. During our initial experiments, accurate calibration proved difficult so we looked for a post-experiment calibration approach that would compute an accurate transformation between two coordinate systems, taking into account all information in the 6DOF data. Once the two data streams have been calibrated, we wish to compare the two for the magnitude and nature of the differences in order to characterize the 6DOF system under test.

The real-time visual servoing implementation used in this study was developed at the Purdue Robot Vision Lab[1] us-

PerMIS '09, September 21-23, 2009, Gaithersburg, MD, USA
Copyright 2009 ACM 978-1-60558-747-9/09/09 ...$10.00.

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST nor does it imply that the materials or equipment identified are necessarily the best for the purpose.

ing a subsumptive, hierarchical, and distributed vision-based architecture for smart robotics [3,6,16,17]. This is a robust, advanced dynamic visual servoing implementation with a high-level of fault tolerance to non-cooperative conditions such as severe occlusions and sudden illumination changes. The Purdue system combines a ceiling mounted camera with a trinocular system mounted on the robot end-effector, and uses position based visual servoing (PBVS). The work in this paper is aimed at the evaluation of sensors for PBVS, in which the servoing system senses the position and orientation of the part in 3D coordinates, as opposed to image based visual servoing (IBVS), in which the servoing system senses the position and orientation of the part in 2D image coordinates.

2. PREVIOUS WORK

While pose estimation and visual servoing receive attention in the literature, evaluation of visual servoing usually appears as a secondary element to the presentation of new servoing approaches or algorithms. Many papers that present a new approach include an empirical evaluation, but since the paper emphasizes the development of the new approach, the evaluation section is often brief. Two papers that do focus on the evaluation of visual servoing algorithms are [1, 5]. In [5], there is a sensitivity analysis and simulation to compute the contribution of image measurement errors to the calculated pose and control trajectory for PBVS and hybrid visual servoing. In [1], there is a modular analysis of the elements of a visual servoing systems with the intention of supporting a design and evaluation framework, with an emphasis on the control subsystem. The paper considers many aspects of performance analysis for static and dynamic cases, as well as accuracy and timing issues.

Most evaluation papers consider static pose only [6, 8] and not 6DOF sensor measurements under motion. References [8, 9] use Monte Carlo simulation for the evaluation of pose algorithm accuracy under noise and object orientations. In those articles, results are given for pose estimation for a complex industrial part and the error from unidentified ground truth is plotted as position or orientation error vs. the rotation of the object. The key result is to note that the error as a function of part rotation varies considerably, spiking at ambiguous orientations of the object. Two papers that do consider dynamic pose are [7, 11]. In [11] disassembly used car parts video sequences are used for tests of a model-based algorithm with four parameter variations to analyze the relative contributions of subcomponents such as the edge detection operator or search technique. The results are given as deviations from the results of the one parameter set that successfully maintained track through the video sequences, but the nature and quality of this retrospective ground truth is not described in the article. In [11] three tracking approaches for 6DOF pose estimation and grasping of hand-held objects are evaluated using ground truth from an unidentified infrared marker tracking system good to 1.5 m in position but with no rotation accuracy or measurements per second cited. The three approaches run at between 8 Hz and 25 Hz. The article gives results in graphs that compare ground truth position and orientation data to robot end-effector position and tracked position, but no quantitative or summary statistics are given for the graphed data.

The metrics used to evaluate pose estimation and visual servoing systems vary. They include the mean and standard deviation of a measure of error in world coordinates, including individual differences for each coordinate, a norm for position and orientation separately, and rarely a combined norm for all 6 degrees of freedom. The orientation can be compared in roll-pitch-yaw, quaternion, or angle-axis representations. In experiments without ground truth in world coordinates, or for IBVS in which pose in world coordinates is not computed, errors are computed in the image domain. In some visual servoing evaluations, the metric is the number of cases successfully completed during the experiments. In physical experiments in the evaluation of pose estimation or visual servoing, a mechanism must be used to generate motion, frequently a robot arm [3, 4, 11]. [11] uses an arm to move a camera towards a car battery through a known trajectory linear in both translation and angle, and repeats the motion 80 times to judge repeatability of the tracking algorithm.

3. VISUAL SERVOING EXPERIMENTS

3.1 Purdue Data

The Purdue system produces a 6DOF pose at the rate of 30 Hz. The output consists of 3 translations and 3 rotational angles, all relative to the robot base frame. The object whose, pose is measured by the Purdue system, is a typical engine cover about 0.5 m in width and 0.25 m in height. Figure 1 defines the object frame.

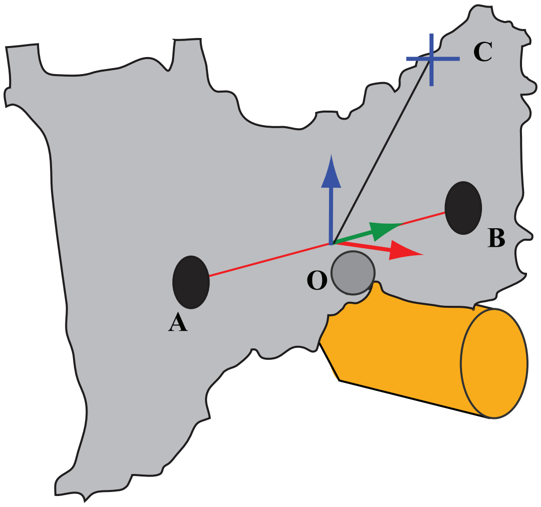


Figure 1: Engine cover and the object frame: A,B and C are coplanar in the YZ plane. O is centered between A and B. OB is the Y axis, while the X-axis is in the direction of the cross product of OB and OC. The Z-axis is given by the cross product of axes X and Y.

3.2 Laser Tracker Data

The laser tracker (LT) measures the 3D locations of a smart track sensor (STS), which measures its own orientation. Together, the two measurements give a complete 6DOF pose of the STS at a rate of up to 150 Hz.

In our experiments, the STS is rigidly attached beneath the engine cover (object) as shown in Figure 2. The laser

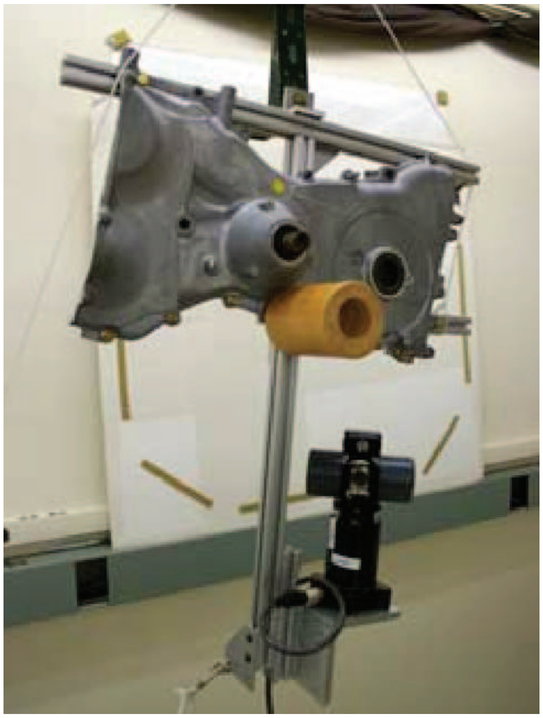


Figure 2: Engine cover (object) and the STS

tracker measures the position and orientation of the STS, while the Purdue system measures the position and orientation of the engine cover (object). However, since the STS and the engine cover (object) are fixed rigidly to each other, the laser tracker can be utilized to compute the transformation between the two. A point of concern is that the Purdue data is in the coordinate system of the robot base whereas the laser tracker has its own coordinate system. Calibrating these two coordinate systems can be a daunting task. Therefore, the objective of this paper is to use the data to construct the best transformation of the robot base coordinate system into the laser tracker coordinate system. The methodology of this process is shown in Section 5.

3.3 Synchronization of the two systems

In order to achieve a matched-pairs design, we take simultaneous measurements and thus minimize the difference in system outputs due to independent measurements taken at different times and different rates. Another advantage of taking simultaneous measurements is that we do not need to know the object motion.

Synchronization can be easily achieved through a common external signal to trigger data acquisition. We use a 30 Hz square wave signal as the Purdue system requires a steady 30 Hz data stream (limited by the cameras' frame rate).

Although a common external trigger signal is used to trigger the data acquisition of both systems, the Purdue system does not latch data instantly. This is because the cameras in the Purdue system do not use a fixed shutter/exposure time. After a trigger signal is received, the cameras open their shutters for some amount of time to collect light. In general, the amount of time changes from frame to frame, depending on the lighting condition at the time. During this exposure/integration time, motion blur can happen. Although the integration time is small, it can be a source of uncertainty in determining the exact pose of the object. Large motion blur will increase pose uncertainty.

In order to be able to uniquely identify and track each trigger signal, the data collection software from each system maintains its own sequence counter and tags each count with a time-stamp having microsecond resolution. Both data collection software modules get their timestamps indirectly from a common clock source via an Network Time Protocol (NTP) server. However, instead of running an NTP client, which attempts to model the clock drift over a long period of time, we simply have the data collection computers synchronize the NTP sever every 10 s. We find this setup allows our data collection computers to stay synchronized to each other within 3 ms. In general, the clock circuits in today's consumer computers are precise but temperature dependent.

3.4 Experimental Setups

We conducted two sets of experiments, one with the object stationary and one with the object moving with a simple linear velocity.

3.4.1 Stationary Tests

The stationary tests allowed us to evaluate the basic performance of both systems and assure that the laser tracker was performing to specification after shipping. The object was placed in four positions and data were collected for 15 s to 30 s each.

3.4.2 Linear Motion Tests

In the linear motion tests, the object was moved about 1.5 m left to right. For each trial, the motion was repeated 30 times as the object moved.

4. CALIBRATION

In order to compare data streams collected from the Purdue system with data streams collected from the laser tracker system, which we consider to be ground truth, both systems must first be placed in the same coordinate system. In other words, a homogeneous matrix that transforms the Purdue data into the coordinate system of the laser tracker system data is needed. We define ${}_{X}\mathbf{H}_{Y}$ as the homogeneous transformation from the coordinate system of Y to X. In other words, ${}_{X}\mathbf{H}_{Y}$ defines the 6DOF pose of Y in X coordinates. Therefore in this paper, we are searching for ${}_{LT}\mathbf{H}_{RB}$ where LT is the output of the laser tracker system and RB is the output of the Purdue system. In [2], a description of the methodology for the output of both the Purdue system and the laser tracker system is provided. A review is given in Section 3. Here, we will give an overview of the necessary components (Figure 3).

The Purdue system provides ${}_{RB}\mathbf{H}_{O}$, where RB denotes the robot base and O denotes the object of interest. Similarly, the laser tracker system provides ${}_{LT}\mathbf{H}_{STS}$. However, ${}_{LT}\mathbf{H}_{O}$ is what is needed. This can be calculated by noting that

$${}_{LT}\mathbf{H}_{O} = {}_{LT}\mathbf{H}_{STS} \times {}_{STS}\mathbf{H}_{O} \tag{1}$$

and

$${}_{STS}\mathbf{H}_{O} = {}_{STS}\mathbf{H}_{LT} \times {}_{LT}\mathbf{H}_{O} \tag{2}$$

is a fixed value and thus only one coordinate frame is needed to construct it. ${}_{STS}\mathbf{H}_{LT} = ({}_{LT}\mathbf{H}_{STS})^{-1}$ and ${}_{LT}\mathbf{H}_{O}$ is constructed by using the laser tracker along with a spherically

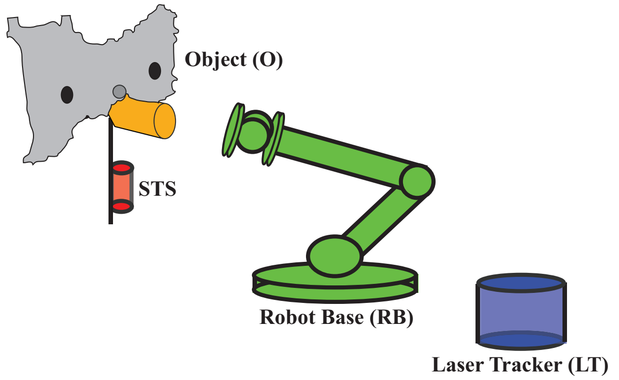


Figure 3: The necessary components of the Purdue data stream and the laser tracker data stream.

mounted reflector (SMR) to calculate the Cartesian coordinate position of three features on the object. These three features are enough information to identify the object's reference frame [2].

The output for the laser tracker system is ${}_{\mathrm{LT}}\mathbf{H}_{\mathrm{O}}$ whereas the output for the Purdue system is ${}_{\mathrm{RB}}\mathbf{H}_{\mathrm{O}}$. Therefore, to be able to compare the two outputs ${}_{\mathrm{LT}}\mathbf{H}_{\mathrm{RB}}$ is needed. The following section describes a mathematical method that constructs ${}_{\mathrm{LT}}\mathbf{H}_{\mathrm{RB}}$ by forming the best homogeneous matrix that fits the data ${}_{\mathrm{RB}}\mathbf{H}_{\mathrm{O}}$ from the Purdue system to the ${}_{\mathrm{LT}}\mathbf{H}_{\mathrm{O}}$ of the laser tracker system.

5. MATHEMATICAL ANALYSIS

5.1 Homogeneous Matrix

In the previous section, an overview of how the data streams are constructed from both the Purdue system and the laser tracker system is given. The output of the Purdue system is given as a series of homogeneous matrices

$$\left({}_{\mathrm{RB}}\mathbf{H}_{\mathrm{O}}\right)_i = \begin{bmatrix} \widehat{\mathbf{R}}_i & \widehat{t}_i \\ 0 & 1 \end{bmatrix} \tag{3}$$

for time steps $i = 0, 1, , n-1$. Similarly, the output of the laser tracker system is a series of homogeneous matrices

$$\left({}_{\mathrm{LT}}\mathbf{H}_{\mathrm{O}}\right)_i = \begin{bmatrix} \mathbf{R}_i & t_i \\ 0 & 1 \end{bmatrix} \tag{4}$$

for time steps $i = 0, 1, , n-1$. We are interested in finding a rotation $\mathbf{R}$ and translation $\mathbf{t}$ that best transforms the coordinate system of the Purdue data into the coordinate system of the laser tracker data. Specifically, we want to construct the homogeneous matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

that solves

$$\min_{\mathbf{H}} \|\mathbf{HP} - \mathbf{L}\|^2$$

where

$$\mathbf{P} = \left[\left({}_{\mathrm{RB}}\mathbf{H}_{\mathrm{O}}\right)_0 \quad \left({}_{\mathrm{RB}}\mathbf{H}_{\mathrm{O}}\right)_1 \quad \dots \quad \left({}_{\mathrm{RB}}\mathbf{H}_{\mathrm{O}}\right)_{n-1} \right]$$

and

$$\mathbf{L} = \left[\left({}_{\mathrm{LT}}\mathbf{H}_{\mathrm{O}}\right)_0 \quad \left({}_{\mathrm{LT}}\mathbf{H}_{\mathrm{O}}\right)_1 \quad \dots \quad \left({}_{\mathrm{LT}}\mathbf{H}_{\mathrm{O}}\right)_{n-1} \right].$$

Shah develops an algorithm for constructing such an $\mathbf{H}$ in [10]. Specifically, the best rotation has to first be constructed as

$$\mathbf{R} = \mathbf{VDU}^T$$

where the singular value decomposition of

$$\mathbf{X}\widehat{\mathbf{X}}^T = \mathbf{USV}^T$$

with

$$\begin{aligned} \mathbf{X} &= \left[\mathbf{R}_0 \quad \mathbf{t}_0 \quad \dots \quad \mathbf{R}_{n-1} \quad \mathbf{t}_{n-1} \right] \\ \widehat{\mathbf{X}} &= \left[\widehat{\mathbf{R}}_0 \quad \widehat{\mathbf{t}}_0 \quad \dots \quad \widehat{\mathbf{R}}_{n-1} \quad \widehat{\mathbf{t}}_{n-1} \right] \end{aligned}$$

and

$$\mathbf{t}_i = t_i - t \quad \text{with} \quad t = \frac{1}{n}\sum_{i=0}^{n-1} t_i \tag{5}$$

$$\widehat{\mathbf{t}}_i = \widehat{t}_i - \widehat{t} \quad \text{with} \quad \widehat{t} = \frac{1}{n}\sum_{i=0}^{n-1} \widehat{t}_i. \tag{6}$$

Also

$$\mathbf{D} = \begin{cases} \operatorname{diag}(1,1,1) & \text{if } \det(\mathbf{VU}^T) = 1, \\ \operatorname{diag}(1,1,-1) & \text{if } \det(\mathbf{VU}^T) = -1. \end{cases}$$

Once the rotation $\mathbf{R}$ is found, the translation $\mathbf{t}$ can be constructed by setting

$$\mathbf{t} = \widehat{t} - \mathbf{R}t$$

where t and $\widehat{t}$ are defined in (5) and (6), respectively.

5.2 Error Metrics

Given a general homogeneous matrix $\mathbf{H}$ – made up of a rotation $\mathbf{R}$ and translation $\mathbf{t}$ – a series of metrics is now offered to compare how well $\mathbf{H}$ transforms a given data stream into another [10].

To see how well a given rotation $\mathbf{R}$ transforms a single rotation $\mathbf{R}_i$ from the laser tracker data stream to a rotation $\widehat{\mathbf{R}}_i$ from the Purdue data stream, evaluate

$$\begin{aligned} \|\mathbf{RR}_i - \widehat{\mathbf{R}}_i\|^2 &= \|\mathbf{RR}_i\|^2 - 2\operatorname{tr}\left(\mathbf{RR}_i\widehat{\mathbf{R}}_i^T\right) + \|\widehat{\mathbf{R}}_i\|^2 \\ &= 6 - 2(1 + 2\cos\theta) \end{aligned}$$

where $\{1, \cos\theta \pm i \sin\theta\}$ are the eigenvalues of $\mathbf{RR}_i\widehat{\mathbf{R}}_i^T$. Therefore,

$$0 \le \|\mathbf{RR}_i - \widehat{\mathbf{R}}_i\|^2 \le 8,$$

since $-1 \le \cos\theta \le 1$. Moreover,

$$0 \le 1 - \frac{1}{8}\|\mathbf{RR}_i - \widehat{\mathbf{R}}_i\|^2 \le 1.$$

defines a metric between 0 and 1 where 1 denotes a perfect fit.

A similar metric can be constructed to compare a given translation t_i from the laser tracker data stream to a translation $\widehat{t}_i$ from the Purdue data stream. In this case we want to see how close $\mathbf{R}t_i + \mathbf{t}$ is to $\widehat{t}_i$. Thus, we consider the dot product between these two normalized vectors. In other words, we evaluate

$$0 \le \frac{(\mathbf{R}t_i + \mathbf{t})^T\,\widehat{t}_i}{\|\mathbf{R}t_i + \mathbf{t}\|\|\widehat{t}_i\|} \le 1$$

Once again this defines a metric with values between 0 and 1 where 1 denotes a perfect fit. One should note that this metric loses valuable information regarding the scaling of the problem. For example, if the vectors $\mathbf{R}t_i + \mathbf{t}$ and $\widehat{t_i}$ point in the same direction (not necessarily the same magnitude), then the metric would give an accuracy reading of 1, though the vectors may not be equal. However, this metric is problem independent allowing one to compare two different problem setups. Another metric that can be used (but is problem dependent) is to look at

$$\|\mathbf{R}t_i + \mathbf{t} - \widehat{t_i}\|.$$

However, this metric does not have a defined upper bound. Instead, one could compare the magnitude of this metric with the magnitude of the data used in order to calculate the accuracy of the algorithm.

In the next section, experiments will be performed to see how well the homogeneous matrix constructed in 5.1 performs using the metrics just defined.

6. EXPERIMENTS

The algorithm in 5.1 that constructed the best homogeneous matrix $\mathbf{H}$ to fit two streams of 6DOF data was applied to data streams that were collected from the Purdue system and a laser tracker system at Purdue University in April of 2008 [2]. These data streams were obtained from two experiments (see Section 3).

6.1 Stationary

In the stationary experiment, the object was placed in four positions for 15 s to 30 s each. The mean distance for each position was in the 3500 mm to 4100 mm range with a standard deviation of 0.006 to 0.008 for the STS/LT system. For the Purdue Line tracker system each position was in the 2600 mm to 2700 mm range with a standard deviation of 0.560 to 0.630 standard deviation. More details can be found in [2]. Overall, the laser tracker system is two orders of magnitude more accurate than the Purdue system.

The homogeneous matrix calculated from these stationary data streams is

$$\mathbf{H}_{\text{Stat}} = \begin{bmatrix} -0.79 & -0.61 & -0.11 & 715.94 \\ 0.60 & -0.79 & 0.07 & 2228.30 \\ -0.13 & -0.01 & 0.99 & -1133.76 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We calculated the accuracy of this homogeneous matrix using the metrics provided in the previous section. Not surprisingly, we have near 100 % accuracy for this homogeneous matrix for both the rotation and translation as can be seen in Figure 4. In addition, the translational error is around 12 mm – a two order decrease in magnitude compared to the data position.

6.2 Linear Motion

In the linear motion experiment, the object was moved 1.5 m to the left and right. This motion was repeated 30 times for each trial and quickly returned back to the starting position. It should be noted that this backward sweep was ignored in the data collection for both systems.

The homogeneous matrix calculated from these linear motion data streams is

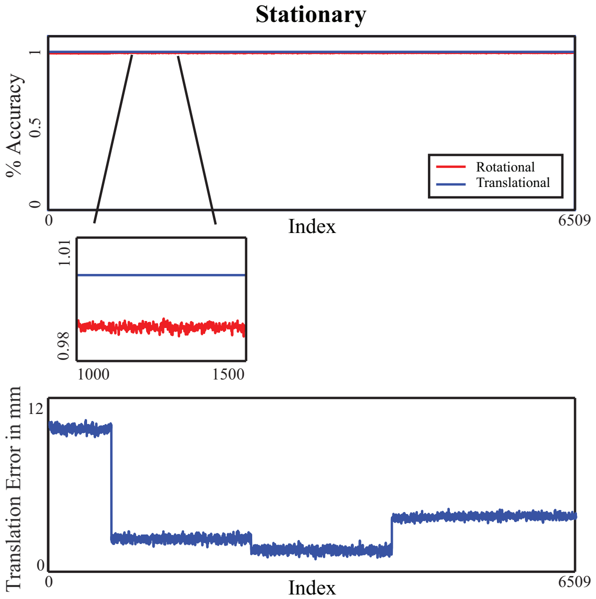


Figure 4: Error metrics from the stationary experiment where the object was placed in four positions for 15 s to 30 s each.

$$\mathbf{H}_{\text{Move}} = \begin{bmatrix} -0.79 & -0.61 & -0.08 & 666.20 \\ 0.61 & -0.79 & 0.04 & 2271.00 \\ -0.09 & -0.01 & 1.00 & -1238.45 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which is not very different from the stationary homogeneous matrix $\mathbf{H}_{\text{Stat}}$. This should be expected since this experiment should not have much noise introduced from the simple linear motion. Moreover, the minimal noise results in $\mathbf{H}_{\text{Move}}$ having near 100 % accuracy for both the rotation and translation as can be seen in Figure 5. In addition, the translational error is only around 10 mm.

7. CONCLUSIONS

In this paper, we presented improved techniques for the calibration of two 6DOF data streams. Previously, calibration was done by hand which was prone to errors. Here, the data was used to mathematically find the best fit between two given 6DOF data streams. Specifically, we constructed the homogeneous matrix that best transformed the coordinate system of one of the two data streams into the other. Moreover, metrics were offered to evaluate the effectiveness of this transformation.

We tested this method on two data sets collected at Purdue University. The first consisted of the object being placed in four positions for 15 s to 30 s and the second consisted of the object moving in a linear motion. We found that the homogeneous matrix fit the data almost perfectly for these two systems.

8. ACKNOWLEDGMENTS

We thank Mike Shneier and Elena Messina for their fruitful discussions. We thank Avi Kak, Johnny Park, and Ger-

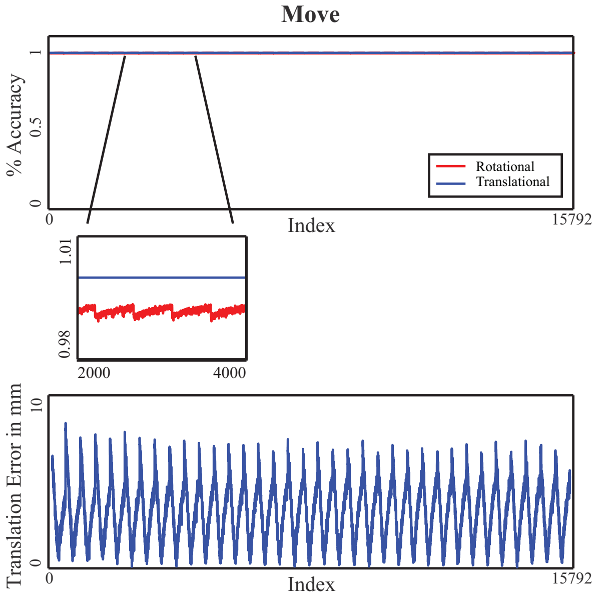


Figure 5: Error metrics from the move experiment where the object was moved 1.5 m to the left and right.

man Holguin from Purdue University for working with us on the real-time data collection using their vision system, robot arm, and rail. In addition, we would like to thank Jane Shi from General Motors, Frank Maslar from Ford, and Kam Lau from Automated Precision Inc. for their collaboration.

9. REFERENCES

[1] M. Bachiller, J. A. Cerrada, and C. Cerrada. A modular scheme for controller design and performance evaluation in 3d visual servoing. *Journal of Intelligent and Robotics Systems*, 36(3):235–264, 2003.

[2] T. Chang, T. Hong, M. Shneier, G. Holguin, J. Park, and R. Eastman. Dynamic 6dof metrology for evaluating a visual servoing system. In *Performance Metrics for Intelligent Systems (PerMIS) Workshop*, pages 173–180, 2008.

[3] K. Deguchi. A direct interpretation of dynamic images with camera and object motions for vision guided robot control. *International Journal of Computer Vision*, 37(1):7–20, 2000.

[4] B. Espiau. Effect of camera calibration errors on visual servoing in robotics. In *International Symposium on Experimental Robotics*, pages 182–192, 1993.

[5] V. Kyrki, D. Kragic, and H. I. Christensen. Measurement errors in visual servoing. *Robotics and Autonomous Systems*, 54(10):815–827, 2006.

[6] C. B. Madsen. Viewpoint variation in the noise sensitivity of pose estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 41–46, 1996.

[7] P. Preisig and D. Kragic. Robust statistics for 3d object tracking. In *International Conference on Robotics and Automation*, pages 2403–2408. IEEE, 2006.

[8] G. J. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. In *ISSAC*, pages 215–223, 2003.

[9] A. H. Rivera-Ríos, F.-L. Shih, and M. M. Marefat. Stereo camera pose determination with error reduction and tolerance satisfaction for dimensional measurements. In *International Conference on Robotics and Automation*, pages 423–428, 2005.

[10] M. I. Shah. Six degree of freedom point correspondences. Technical report, Department of Mathematical Sciences, Loyola College in Maryland, 2009.

[11] M. Tonko and H.-H. Nagel. Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments. *International Journal of Computer Vision*, 37(1):99–118, 2000.

Ontology Formalisms: What is Appropriate for Different Applications?

Craig Schlenoff
National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
301-975-3456

craig@schlenoff.com

ABSTRACT

Ontologies can take many forms. There are ontologies that are extremely formal (e.g., using first order logic), and there are ontologies that are less formally defined (e.g., ontologies in the relational databases or dictionaries). Nonetheless, all of these can be considered ontologies and are appropriate in different situations.

In this paper, I will present a view of levels of ontology formalizations and then describe three efforts that have applied ontologies to solve real-world problems. I will show where each of these efforts fall on the formalization spectrum and show why that level of formalization is appropriate for that application.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods Language – *predicate logic, relation systems, representation languages, representations*

General Terms

Design, Experimentation, Standardization, Languages, Theory

Keywords

Ontologies, Formalization, Robotic, Knowledge Representation

1. INTRODUCTION

Ontologies can take many forms. There are ontologies that are extremely formal (e.g., using first order logic), and there are ontologies that are less formally defined (e.g., ontologies in the relational databases or dictionaries). Nonetheless, all of these can be considered ontologies and are appropriate in different situations.

Similarly, ontologies can play different roles. They can be used for common access to information, for search, as exchange languages and for reasoning.

In this paper, I will present one view of different levels of ontology formalizations and then describe some efforts that have applied ontologies to solve real-world problems. I will then show where each of these efforts fall on the formalization spectrum and show why that level of formalization is appropriate for that application. Section 2 describes the formalization scale that will be using for this paper and Section 3 gives an overview of how ontologies have been used in real-world applications. Section 4 describe the details of three projects that have used ontologies and how they fit into the classifications described in Sections 2 and 3. Section 5 concludes the paper.

2. LEVELS OF ONTOLOGY FORMALISM

For the purpose of this paper, I will loosely define an ontology as a knowledge representation that can be captured at different levels of formality, ranging from terms in a glossary or dictionary up to formal logic-based descriptions. Admittedly, this definition of an ontology is much broader than the commonly-accepted view. Often, people think of an ontology as a more formal representation; often one that can be reasoned over in an automated fashion.

To describe the level of formalisms of an ontology, I will use the scale in Figure 1. This scale shows examples of ontologies listed from least formal (left side of the figure), to more formal (right side of the figure). The black diagonal line in the middle of figure shows the point at which ontologies can be reasoned over. One can run a reasoning engine over everything to the right of the line but cannot over the formalisms to the left of the line. This figure was not created by the author; it is often used in the literature but the author was unable to find the origin of it.

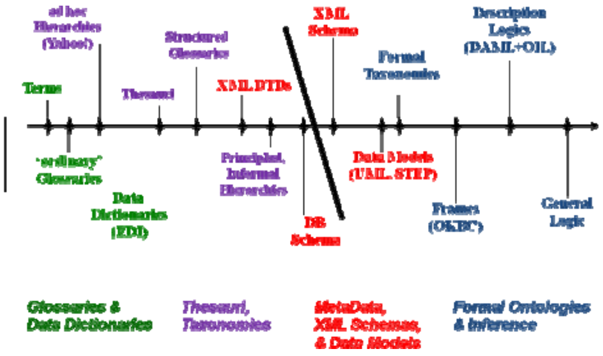


Figure 1: Levels of Ontology Formalism

The items in green can be thought of as standard glossaries or dictionaries, similar to the ones that you may have in your house. The items in purple are thesauri, taxonomies, and hierarchies. In this realm, one is starting to organize and categorize information.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
ACM 978-1-60558-747-9/09/09

These structures start to exhibit some superclass/subclass types of relationships, and can be used for application such as navigating web pages. The red items start providing a lot more structure to data, and provide a much richer set of relationships, such as part-of, contains, spatial relations, etc. These structures are often used as specification for software, exchange languages, and ontology-based search. The items in blue can be thought of as formal ontologies and are often represented in logic-based languages such as the Knowledge Interchange Format (KIF) or description logics. The advantage of these types of formalisms is that they allow for inferencing. This allows one to discover additional information that is not formally represented and also allows one to identify inconsistencies in the knowledge that is represented. For the remainder of this paper, I will be using this formalization scale to characterize existing efforts in ontology development for robotics and related applications.

3. APPLYING ONTOLOGY: THE BIG PICTURE

Over the past two decades, ontologies have found a role in many different applications. Four ways in which ontologies have been used include:

- Common access to information
- Ontology-based search
- Exchange language
- Reasoning

Each of these is discussed in more detail below.

3.1 Common Access to Information

It is often the case that multiple applications need access to the same information. This type of information could be a material database, a specification of a part to be manufactured, or terrain characteristics of an environment that an autonomous vehicle must traverse. Instead of requiring an application to encode this information in its own internal format, an ontology can provide this information in a neutral format that these different applications can reference. By not duplicating this information is numerous different applications, the ontology can allow the information to be represented only once, providing only a single source when information needs to be updated and ensuring that there is consistency between the information in different applications. In addition, the ontology provides a common set of vocabulary that all of the applications that access the ontology can reference. This will ensure that when information exchange between the applications needs to occur, mappings between concepts can be easily done based on the vocabulary in the ontology. The Road Network Database and the Intelligent Systems Ontology, discussed later in the paper, are examples of ontologies developed for common access to information.

3.2 Ontology-Based Search

It is not uncommon that a given concept can have two terms that correspond to it. For example, when a person goes to a web site and wants to buy a helmet that will protect their head at a construction site, they may type in either “hard hat”, “protective helmet”, “hard helmet”, or possibly other terms. Depending what the person types is, often different results will be displayed. This is because search engines often work on the term that is entered as opposed to the concept that is intended.

Ontologies can help to address this issue by representing concepts by what they mean (as opposed to the terms that are used to represent them) and then mapping search terms and underlying information in product databases to those ontological concepts. This is not only true with products… it can also be done with web pages or any other item that needs to be searched.

There are no specific examples of this type of ontology application in the paper but please contact the author if you would like to learn more about how this was applied in private industry.

3.3 Ontologies as an Exchange Language

Information often needs to be shared among different applications. This information is usually generated in one application and then needs to be sent to another application. An example of this is in the manufacturing domain where a person may use a process planning system to create a part in one application and then needs to send that information to a scheduling or production planning system to allow the part to be made. The problems with point-to-point translators between each pair of application are well documented, and these result in a very large amount of translators that need to be developed. Also, as a new version of the application is released, all of the translators that are written either to or from that application need to be updated.

Ontologies have shown to be valuable in serving as a neutral representation to allow for the exchange of information between different applications. The ontology provides a common superset of all of the information structures that need to be exchanged between the applications. By having this common interchange structure, a given application would only have to write a translator to and from the ontology and then would be able to exchange information with any other application that has done the same.

STEP [1] (STandard for the Exchange of Product model data) is perhaps the most widely used ontology for this purpose. In this paper, I describe the Process Specification Language (PSL) which is a more formal ontology that is used to exchange process data among applications.

3.4 Ontologies for Reasoning

When represented formally, ontologies have the ability to reason over information and provide additional information that was not previously formally represented. For example, when an autonomous vehicle is driving down a road and is presented with multiple paths, each of which that has an obstacle in its way, the ontology can reason about the expected damage that could occur by hitting each of the obstacles based on their known characteristics and those of the vehicle. Then a proposed path can be presented to a planner to determine how the vehicle should proceed. An ontology for navigation planning is discussed later in this paper which shows how ontologies can be used for this purpose.

4. ONTOLOGY EXAMPLES

In this section, I will describe existing and past efforts that have used ontologies for real-world applications. For each effort, I will characterize it with respect to its level of formality as described in Section 2 and what role it is playing as described in Section 3. I will start with ontologies that are considered to be less formal and then proceed to more formal ontologies.

4.1 The Road Network Database

For an autonomous vehicle to navigate a road network, it must be aware of and must respond appropriately to any object it encounters. This includes other vehicles, pedestrians, debris, construction, accidents, emergency vehicles … and the roadway itself. The road network must be described such that an autonomous vehicle knows, with great precision and accuracy, where the road lies, rules dictating the traversal of intersections, lane markings, road barriers, road surface characteristics, and other relevant information.

The purpose of this section is to provide an overview of the Road Network Database [2], which is to provide the data structures necessary to capture all of the information necessary about road networks so that a planner or control system on an autonomous vehicle can plan routes along the roadway at any level of abstraction. At one extreme, the database should provide structures to represent information so that a low-level planner can develop detailed trajectories to navigate a vehicle over the span of a few meters. At the other extreme, the database should provide structures to represent information so that a high-level planner can plan a course across a country. Each level of planning requires data at different levels of abstraction, and as such, the Road Network Database must accommodate these requirements. In this section, I explore the contents of the Road Network Database and describe why it was represented in a database format as opposed to a more formal ontology.

The fundamental components of the Road Network Database are described below. This is not an exhaustive list, but instead is meant to give the reader an idea of the type of structures that are represented in the database.

- **Junctions** – A junction is a generic term referring to two or more paths of transportation that come together or diverge, or a controlled point in a roadway. Examples of roadway paths that could cause a junction are lanes splits, forks in the road, merges, and intersections.
- **Intersections** - Intersections are a type of junction in which two or more separate roads come together.
- **Lane Junctions** - A lane junction is a location in a junction in which two or more lanes of traffic overlap.
- **Road** – A road is a stretch of travel lanes in which the name of the travel lanes does not change. An example is “Main Street” or “Route 95.”
- **Road Segment** - A road segment is a uni-directional stretch of roadway bounded by intersections. A road segment is roughly analogous to a “block”.
- **Road Element** - A road element is a uni-directional stretch of roadway bounded by any type of junction. Unlike road segments, road elements can be bounded by merging lanes, forks in the road,
- **Lane Cluster** - A lane cluster is a set of uni-directional lanes (with respect to flow of traffic) in which no physical attribute of those lanes change over the span of the lane segment. Unlike a road element, lane clusters are not required to be bounded by junctions.
- **Lane** - A lane is a single pathway of travel that is bounded by explicit or implicit lane marking.
- **Lane Segment** - A lane segment is the most elemental portion of a road network captured by the database structure. Lane segments can be either straight line or constant curvature arcs. One or more lane segments compose a lane
- **Junction Lane Segments** - A junction lane segment is a constant curvature path through a portion of a lane junction.

As stated earlier, the data structures are designed to accommodate a control system that may contain planners with various levels of abstraction. The planners, their descriptions, and the data structures which best correspond to their level of responsibility are shown in Table 1.

Table 1: Planner to Data Structure Mapping

Planner Name	Planner Description	Appropriate Data Structures
Destination Planner	Plans the sequence of route segments to get to commanded destination goal. Outputs MapQuest[1]-like directions Plans on the order of 1 to 2 hrs into the future	Roads Road Segments Intersections
Drive Behavior Planner	Develops low-level behaviors for negotiating intersections and deciding when to change lanes. Plans on the order of 100 secs into the future. Plans up to 500 m	Lane Clusters Lanes Intersection
Elemental Maneuver Planner	Carries out real-time maneuvers to slow down, stop, speed up, and change lateral position. Plans on the order of 10 secs into the future Plans up to 50 m distances	Lanes Lane Segments

[1] The name of commercial products or vendors does not imply NIST endorsement or that this product is necessarily the best for the purpose.

This information is represented in a relational database. An example of the detailed information that was represented for a road segment can be seen in Table 2. The corresponding picture of what a road segment may look like is shown in Figure 2.

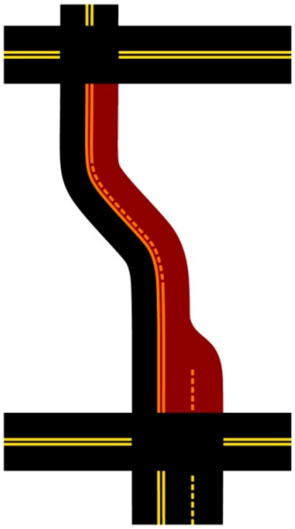


Figure 2: Sample Road Segment

A road segment is a uni-directional stretch of roadway bounded by intersections. A road segment is composed of one or more road elements and zero or more junctions. There are one or more road segments in a road. Unlike road elements, road segments are only bounded by intersection, not any type of junction. A road segment within a road must always be rendered in the same direction as the road. Road segments are used in the planning and control system to provide MapQuest-like directions to the vehicle to allow for route planning.

This Road Network Database is represented as a database schema (on the left side of the formalization figure shown in Section 2). The reason why a more informal representation was chosen was because the database was meant to serve for common access to information (as described in Section 3). It was not anticipated that any reasoning would need to be performed on the data structures so a more formal type of representation (e.g., logic) was not needed. Conversely, since the database was expected to provide common access to information, more informal types of representations (glossaries, data dictionaries, informal hierarchies) were not used since they did not provide the level of specificity needed and provide too high a level of ambiguity in the meaning of the terms that were represented.

Table 2: Road Segment Database Representation

Attribute	Data Type	Value Restriction	Point To	Description
ID	Integer	Any whole number greater or equal to one		A unique identifier for this entry in this table
World_ID	Integer		World.ID	A pointer to an element in the World table that indicates with which world this entry is associated. A road segment may only be associated with a single world. See 4.5.1. for information about worlds.
Description	Text			A textual description of this field for human understanding
Road_ID	Integer		Road.ID	A pointer to the element in the Road table in which the road segment is a part of.
Start_Point_Adjacent_Intersection_ID	Integer		Intersection.ID	A pointer to the element in the Intersection table which precedes the road segment.
End_Point_Adjacent_Intersection_ID	Integer		Intersection.ID	A pointer to the element in the Intersection table which follows the road segment.
Segment_Length	Double			Measured in meters. The length of the road segment measured from center point to center point. This should be derived from the length of the road elements which compose it.
Road_Segment_Class	Integer		RoadSegmentClass.ID	A pointer to an element in the RoadSegmentClassLookup table which contains the class of road segment which applies to this road segment.

4.2 Ontologies for Autonomous Navigation

The field of autonomous vehicles has reached a level of maturity such that it could greatly benefit from leveraging the latest technologies in the area of reasoning over knowledge representations and ontologies.[2] The use of ontologies and automated inference is a natural fit for representing and reasoning about world models (the internal knowledge representation) for autonomous vehicles. The goal for the effort described in this section is to apply ontologies to improve the capabilities and performance of on-board route

[2] The 2004 AAAI Spring Symposium series includes a workshop on this the topic: “Knowledge Representation and Ontology in Autonomous Systems”. See: http://www.aaai.org/Symposia/Spring/2004/sssparticipation-04.pdf

planning for autonomous vehicles. More specifically, to apply ontologies to determine the extent to which a given object is an obstacle to a given vehicle in a given situation [3].

There are many potential benefits of introducing an ontology (or set of ontologies) into an autonomous vehicle's knowledge base. One is the potential for reuse and modularity. For example, a general theory of obstacles could apply to a broad range of autonomous vehicles. In addition, ontologies provide a mechanism to allow for a more centralized approach to represent and reason about environmental information. Different modules in an autonomous vehicle would query the ontology, rather than having the information scattered among the modules. This has a corresponding benefit in cheaper and more reliable maintenance. Finally, there is the potential for increased flexibility of response for the autonomous vehicle. Methods that rely on pre-classification of certain kinds of terrain in terms of their traversability [4;5] are important, but do not support reasoning about objects in a more dynamic context.

I start with the simple scenario illustrated in Figure 3. Our vehicle (labeled OV) is in the left lane of a four-lane, two-way, undivided highway. An object is detected in our lane. The goal is to formulate an optimal route plan that takes into account the potential damage from a collision with the object. The main role of the ontology component is [initially] to provide assessments of collision damage. I will take into account not only damage to the vehicle, but also damage to the payload and to the object, itself. This information is used to plan a route that either goes around the object, or collides with it.

A number of parameters may be varied in this scenario. These include: the type of vehicle being controlled, the speed at which the vehicle is traveling, the payload being carried, and type of object in the path that may be an obstacle. For example, if the object is a newspaper in the middle of the roadway, then the ontology component will conclude that no damage will occur and the planner will conclude that the best course of action is to maintain the current lane (because changing lanes always accumulates additional risk over maintaining your lane).

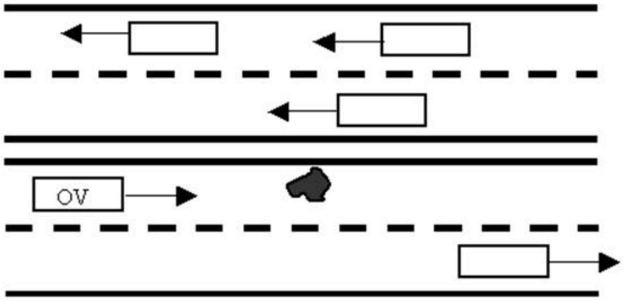


Figure 3. Simple Driving Scenario

However, if the object were a large cinder block, significant damage would be likely and the final route should be quite different. The ontology component is equipped with knowledge about many kinds of vehicles, objects, and the kind of damage that can arise from different collisions. This is used to determine the damage that would be caused by a collision.
The ontology includes objects, vehicles and situations with associated inference rules. Specifically, the ontology contains different types of objects that one expect to encounter in various environments, along with their pertinent characteristics and relationships to other objects. Initially the effort is focusing on on-road driving, so categories of objects such as other vehicles, pedestrians, animals, debris, speed bumps, etc. are represented. Each one of the objects that fall under these categories has a set of characteristics that describe them and help us to understand the damage that may be caused by colliding with them. For example, a certain type of debris may have a set of dimensions, a weight, a density, a velocity, etc. The rules determine the 'degree of obstacleness', which is ultimately expressed in terms of a cost.

The ontology and its associated reasoning engine provides as an output, a damage assessment in the event of a collision between our vehicle and a given object based upon:

- The type of autonomous vehicle;
- The type of object being collided with;
- The closing speed of our vehicle with the object;
- The integrity of our vehicle, i.e., what damage has already occurred to our vehicle, if any.

Based on this information, the ontology provides a damage classification pertaining to:

- The vehicle's integrity (initially only assigning damage to the bumper, wheels, and overall vehicle, but will eventually include other components of the vehicle).
- The obstacle's integrity
- The vehicle payload's integrity

In order to provide the damage classifications, the expressiveness of the ontology must be such that it represents concepts such as:

- The type of vehicle that is being autonomously controlled and its pertinent characteristics;
- The objects that are being encountered in the environment and their pertinent characteristics;
- The payloads that the vehicle is carrying and their pertinent characteristics;
- Severity classifications of damage;
- Damage types;
- Terrain information (initially fixed as paved roads);
- Collisions (e.g., a certain type of vehicle with a certain type of object).

For the initial work, the levels of collision damage shown in Table 3 are assumed.

Table 3: Levels of Collision Damage

	Vehicle	Object	Payload
None	No damage to vehicle	No damage to object	No damage to payload
Minor	Damage to vehicle will not affect vehicle performance	Damage to object will not affect object overall integrity	Damage to vehicle will not affect payload
Moderate	Moderate probability of vehicle damage, maintenance required	Damage to object will affect object integrity, but will not result in object destruction	Moderate probability of payload loss
Severe	Major loss of functionality/ integrity of vehicle likely	Major destruction of object	Major payload loss
Catastrophic	Vehicle loss	Object destruction	Payload loss

There are many approaches that could be used to estimate the actual collision damage. These include:

- Numerical simulation tools which model the physics of weight, materials, shapes, density, momentum etc. to compute impact damage;
- Probabilistic models;
- Fuzzy logic;
- Symbolic logic.

Not one of these techniques is likely to be adequate in all circumstances. The current work focuses on the symbolic logic approach. The hypothesis is that even when logic-based inference is not sufficient, the core ontology of objects and characteristics will remain useful as a conceptualization and vocabulary for expressing rules and procedures for estimating damage.

For the initial experiments, a small ontology was constructed using OilEd [6]. Using a description logic [7] tool has two advantages for us. First, the classifier detects logical errors in the ontology, which greatly increases confidence that the ontology is correct. Second, it is very fast at doing inference. This is important because the planner needs to query the ontology component up to a few hundred times a second to get damage estimates for the many nodes being explored in the search space.

A class called *Situation* was defined which has various characteristics or attributes, each modeled by functional relations with *Situation* as the domain. The key characteristics of a *Situation* that will determine the damage classification are the vehicle, the payload and the object with which the vehicle may collide. These functional relations are called *hasVehicle*, *hasPayload*, and *hasPotentialObstacle*, respectively. Attributes were also used to define the damage categories in Table 3. For example, the class *VehicleIntegrityMinor* is defined to be the class of all *Situations* such that the value of the functional relation *hasVehicleIntegrity* attribute is *Minor*.

A simple ontology of physical objects was constructed that including various types of vehicles and other objects such as bricks, newspapers etc. that may be in the vehicle's environment. These objects have characteristics such as weight, speed, density, etc. that are important in determining the damage category. Initially, some qualitative categories for measuring these characteristics were created, such as low, medium and high for weight, or density.

Finally some axioms were created which specify how to classify a given situation in terms of the categories in Table 3. Here is a simple example:

A *Situation* such that

- The value of the *hasPotentialObstacle* relation is restricted to be of type *SmallDenseObject.*
  &
- The value of the *hasVehicle* relation is restricted to be of type *Car*

is a subclass of *VehicleIntegrityModerate*.

Some fictitious situations were created to test these axioms. For example, the situation whose *hasPotentialObstacle* relation is a brick, and whose *hasVehicle* relation is a Toyota Corolla will be classified by this rule under *VehicleIntegrityModerate.* This is inferable because a brick is a *SmallDenseObject* (by virtue of its weight and size), and a Toyota Corolla is a subclass of *Car*.

This Autonomous Navigation Ontology is represented in description logic (near the right side of the formalization figure shown in Section 2). The reason why a more formal representation was chosen was because the ontology was developed to allow reasoning, which requires that the underlying representation be more formal. The effort clearly falls into the "Ontology for Reasoning" section described in Section 3. Full first order logic could have been chosen in this effort, but it was felt that it was overkill for the fairly simple examples that were anticipated.

4.3 The Process Specification Language

The Process Specification Language (PSL) [8] is addressing the software interoperability issue by creating a neutral, standard language for process specification to serve as an interlingua to integrate multiple process-related applications throughout the manufacturing life cycle. This interchange language is unique due to the formal semantic definitions (the ontology) that underlie the language. Because of these explicit and unambiguous definitions, information exchange can be achieved without relying on hidden assumptions or subjective mappings.

Existing approaches to process modeling lack an adequate specification of the semantics of the process terminology, which leads to inconsistent interpretations and uses of the information. Analysis is hindered because models tend to be unique to their applications and are rarely reused. Obstacles to interoperability arise from the fact that the legacy systems that support the functions in many enterprises were created independently, and do not share the same semantics for the terminology of their process models.

For example, consider Figure 4 in which two existing process planning applications are attempting to exchange data. Intuitively the applications can share concepts; for example, both *material* in Application A and *workpiece* in Application B correspond to a common concept of *work-in-progress*. However, without explicit definitions for the terms, it is difficult to see how concepts in each application correspond to each other. Both Application A and B have the term *resource*, but in each application this term has a different meaning. Simply sharing terminology is insufficient to support interoperability -- the applications must share their semantics.

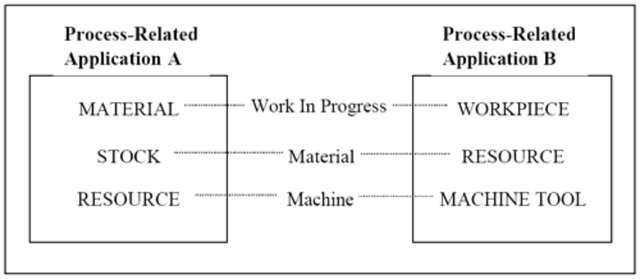


Figure 4: The Need For Semantics

A rigorous foundation for process design, analysis, and execution therefore requires a formal specification of the semantics of process models. One approach to generating this specification is through the use of ontologies. A major goal of PSL is to reduce the number of translators to *O(n)* for *n* different ontologies, since it would only require translators from a native ontology into the interchange ontology.

Within this work, the term “ontology” refers to a set of sentences in first-order logic, comprising a set of foundational theories and sets of definitions written using the foundational theories. In providing such an ontology, one must specify three notions:

- Language
- Model theory
- Proof theory (axioms and definitions)

A language is a set of symbols (lexicon) and a specification of how these symbols can be combined to make well-formed formulae (grammar/syntax). The lexicon consists of logical symbols (such as connectives, variables, and quantifiers) and non-logical symbols. For PSL, the non-logical part of the lexicon consists of expressions (constants, function symbols, and predicates) that refer to everything needed to describe processes.

The underlying language used for PSL is KIF[9] (Knowledge Interchange Format). Briefly stated, KIF is a formal language developed for the exchange of knowledge among disparate computer programs. KIF provides the level of rigor necessary to define concepts in the ontology unambiguously, a necessary characteristic to exchange manufacturing process information using the PSL Ontology.

The primary component of PSL is its terminology for classes of processes and relations for processes and resources, along with definitions of these classes and relations. Such a lexicon of terminology along with some specification of the meaning of terms in the lexicon constitutes what this effort is calling an ontology.

The model theory of PSL provides a rigorous mathematical characterization of the semantics of the terminology of PSL. The objective is to identify each term with an element of some mathematical structure, such as a set or a set with additional structure (e.g. a complete partial order); the underlying theory of the mathematical structure then becomes available as a basis for reasoning about the terms of the language and their relationships.

The proof theory of PSL provides axioms for the interpretation of terms in the ontology. It is useful to distinguish two types of sentences in this set of axioms: core theories and definitions. A core theory is a set of distinguished predicates, function symbols, and individual constants, together with some axiomatization. Distinguished predicates are those for which there are no definitions; the intended interpretations of these predicates are defined using the axioms in the core theories. For these terms, one needs to describe the set of models corresponding to the intuitions that one has for them. Axioms are then written that are sound and complete with respect to the set of models. That is, every interpretation that is consistent with the axioms is a model in the set, and any model in the set is an interpretation consistent with the axioms. These axioms constitute the foundational theories of the ontology. The set of models form the semantics (or model theory) of the ontology.

All other terms in the ontology are given definitions using the set of primitive terms. These definitions are known as conservative definitions since they do not add to the expressive power of the core theories, that is, anything that can be deduced with the definitions, can be deduced using the core theories alone. All definitions in an ontology are specified using the core theories; any terminology that does not have a definition is axiomatized in some core theory. Since all other terms are defined using these primitives, the set of models for them can be defined using the models of the core theories for the primitives. One can therefore assign semantics to the definitions using the classes of models that have already been specified for the core theories.

The challenge is that some framework is needed for making explicit the meaning of the terminology for many ontologies that reside only in people's heads. Any ideas that are implicit are a possible source of ambiguity and confusion. For PSL, the model theory provides a rigorous mathematical characterization of process information and the axioms give precise expression to the basic logical properties of that information in the PSL language. So when one speaks about semantics for PSL, it is in reference to the axiomatization of core theories and definitions for the PSL terminology.

The focus of the ontology is not only on the terms, but also on their definitions. An infinite set of terms can be included in the ontology, but they can only be shared if everyone agrees on their definitions. It is the definitions that are being shared, *not simply* the terms. A simple definition with the PSL ontology is shown below:

Definition An activity is-occurring-at a timepoint p if and only if p is betweenEq the activity's begin and end points.

```
(defrelation is-occurring-at (?a ?p) :=
   (exists (?occ)
   (and  (occurrence ?occ ?a)
         (betweenEq (beginof ?occ) ?p (endof ?occ)))))
```

A simple axiom within the PSL ontology is shown below:

Axiom. An object can participate in an activity only at those timepoints at which both the object exists and the activity is occurring.

```
(forall (?x ?a ?t)
   (=>   (participates-in ?x ?a ?t)
         (and       (exists-at ?x ?t)
                    (is-occurring-at ?a ?t))))
```

This Process Specification Language is represented in full first order logic (all the way to the right side of the formalization figure shown in Section 2). The reason for this is two-fold:

1. Precise semantics are needed to ensure that complete and unambiguous information exchange occurs between two applications,
2. Reasoning must be performed over the concepts in the ontology to ensure that mappings between the applications ontology and PSL are complete and correct.

The effort clearly falls into the “Ontology as an Exchange Language” section described in Section 3.

5. CONCLUSION

In this paper, different levels of ontology formalization are discussed and an overview of the ways in which ontologies have been used in practice over the past couple decades is described. Three examples are also provided of very different ontologies that have been developed to solve real-world problems in the autonomous vehicle and manufacturing systems integration domains. It is also explained why the formalisms that were used were the most appropriate for their intended purpose.

There are many other ontology efforts which could have been used as examples, including an ontology for searching products on private company’s web site, an ontology for classifying robot capabilities to allow a first responder to find the best robot for a disaster site, and an ontology that was developed to classify autonomous vehicle behaviors so that the right behavior can be chosen when confronted with specific environmental conditions. The three that were chosen were done so because they provide a good spectrum of the types of formalism that can be used when developing an ontology. If the reader is interested in hearing about these efforts, please don’t hesitate to contact the author.

6. REFERENCES

[1] S. Brooks and R. Greenway, "Using STEP to integrate design features with manufacturing features," in *Computers in Engineering Conference* New York, NY: 1995, pp. 579-586.

[2] C. Schlenoff, S. Balakirsky, T. Barbera, C. Scrapper, J. Ajot, E. Hui, and M. Paredes, "The NIST Road Network Database: Version 1.0," National Institute of Standards and Technology (NIST) Internal Report 7136,2004.

[3] C. Schlenoff, S. Balakirsky, M. Uschold, R. Provine, and S. Smith, "Using Ontologies to Aid in Navigation Planning in Autonomous Vehicles," *Knowledge Engineering Review*, vol. 18, no. 3, pp. 243-255, 2004.

[4] J. J. Donlon and K. D. Forbus, "Using a Geographic Information System for Qualitative Spatial REasoning about Trafficability," in *Proceedings of the Qualitative Reasoning Workshop* Loch Awe, Scotland: 2003.

[5] R. M. Malyyankar, "Creating a Navigation Ontology," in *Proceedings of the Workshop on Ontology Management, AAA!-99* Orlando, FL: 1999.

[6] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens, "OilEd: a reason-able ontology for the semantic web," in *Proc. of the Joint German Austrian Conference on AI, number 2174 in Lecture Notes In Artificial Intelligence* Springer-Verlag, 2001, pp. 396-408.

[7] F. Baader, D. McGuinness, D. Nardi, and F. Patel-Schnedier, *Description Logic Handbook: Theory, Implementation and Application* Cambridge University Press, 2002.

[8] C. Schlenoff, M. Gruninger, F. Tissot, J. Valois, J. Lubell, and J. Lee, "The Process Specification Language (PSL) Overview and Version 1.0 Specification," in *NISTIR 6459, National Institute of Standards and Technology* 2000.

[9] M. Genesereth and R. Fikes, "Knowledge Interchange Format," in *Stanford Logic Report Logic-92-1* Stanford University: 1992.

Universal Core Semantic Layer: A Roadmap to Semantic Interoperability

Lowell Vizenor
National Center for Ontological Research
University at Buffalo
701 Ellicott Street, Buffalo NY, 14214
lowell.vizenor@ctg.com

Barry Smith
National Center for Ontological Research
University at Buffalo
701 Ellicott Street, Buffalo NY, 14214
phismith@buffalo.edu

ABSTRACT

The Universal Core (UCore) is a central element of the US National Information Sharing Strategy that is supported by four Federal Government Departments (Defense, Energy, Justice, Homeland), by the Intelligence Community, and by a number of other national and international institutions. The goal of the UCore initiative is to foster information sharing by means of an XML schema providing consensus representations for four groups of universally understood terms under the headings who, what, when, and where. We here describe a project to create an ontology-based supporting layer for UCore, entitled 'Universal Core Semantic Layer' (UCore SL), and show how UCore SL is being used to further UCore's information sharing goals.

Categories and Subject Descriptors

D.3.1 [**Formal Definitions and Theory**]: Semantics

General Terms

Design, Reliability, Standardization, Languages, Verification.

Keywords

Ontology, Data Integration, Semantic Technology, OWL DL, Universal Core.

1. INTRODUCTION

The use of intelligent agents to assist the warfighter in obtaining accurate and relevant information about the battlespace and thereby reduce the overall impact of the fog a war is a vision that has yet to be fully realized. If intelligent agents are to scan a sea of information (in real time) to prevent mistakes, execute rules, discover anomalies and improve decisions, then it will be necessary to build logically and ontologically sound information artifacts that can truly support these next-generation goals. In this paper, we discuss a project to create an ontology-based supporting layer to further information sharing capabilities and act as a foundation to future applications that place actionable intelligence in the hands of commanders and intelligence analysts.

1.1 The Universal Core

The Universal Core (UCore) [1] is a US Federal Government information sharing initiative that is supported by the US Departments of Defense, Energy, Justice, and Homeland Security, by the Intelligence Community, and by a large number of other national and international agencies. UCore supports the principles of the Department of Defense (DoD) and Intelligence Community (IC) Data Strategies by defining a small set of common data elements that are implemented in a lightweight information exchange schema that is shared across multiple agencies.

The prime focus of the UCore initiative is messaging. UCore is designed to promote information sharing across multiple message domains by means of a simple XML message format built on a taxonomical structure comprising four groups of terms under the headings who, what, when, and where. Table 1, below, represents the taxonomy as released in UCore Version 2.0, which is the version upon which we focus in what follows. Table 2 represents the relations contained within the UCore 2.0 XML Schema.

UCore works by requiring message-creators to construct for each message a digest, a summary built out of a restricted vocabulary of UCore terms, and to link elements from the message payload to this digest. Developers of information systems are encouraged to use these terms wherever practical in order to realize the goal of facilitating automated sharing of information within and across agencies. To reap maximal benefit from its messaging resources, participants in the UCore initiative offer validation processes and tools intended to promote machine understanding of message content, and thereby prospectively enabling multiple different types of information retrieval, reasoning and consistency checking.

The UCore taxonomy consists of terms (such as 'Person' or 'Organization') which are universally understood in the sense that they require no domain-specific expertise for their understanding. The taxonomy can thereby be shared by many different types of users, and thus provides the opportunity for interoperability over many different sorts of domain-specific exchanges. As M. Daconta expresses it:

> if I have a UCore-wrapped National Information Exchange Model [NIEM] message from Immigration and Customs Enforcement about illegal immigrants wounded during criminal activity and I have a UCore-wrapped Health and Human Service Department message on visitors to emergency rooms, I have enabled immediate cross-domain search. … UCore is a process of extracting cross-domain commonality from your message flows, thereby massively broadening the possible adoption and use of your shared information. In information sharing, adoption by consumers is the key value metric [2].

Table 1. UCore 2.0 Taxonomy

uc:Entity		uc:Event	
uc:Cargo	uc:LivingThing	uc:AlertEvent	uc:LawEnforcementEvent
uc:CollectionOfThings	uc:Animal	uc:CommunicationEvent	uc:MigrationEvent
uc:CyberAgent	uc:Person	uc:CriminalEvent	uc:MilitaryEvent
uc:Document	uc:MicroOrganism	uc:CyberSpaceEvent	uc:NaturalEvent
uc:Environment	uc:Plant	uc:DisasterEvent	uc:ObservationEvent
uc:Equipment	uc:Organization	uc:EconomicEvent	uc:PlannedEvent
uc:Facility	uc:PoliticalEntity	uc:EmergencyEvent	uc:PoliticalEvent
uc:FinancialInstrument	uc:Sensor	uc:EnvironmentalEvent	uc:PublicHealthEvent
uc:GeographicFeature	uc:Vehicle	uc:EvacuationEvent	uc:SecurityEvent
uc:GroupOfOrganizations	uc:Aircraft	uc:ExerciseEvent	uc:SocialEvent
uc:GroupOfPersons	uc:GroundVehicle	uc:FinancialEvent	uc:TerroristEvent
uc:InformationSource	uc:Spacecraft	uc:HazardousEvent	uc:TransportationEvent
uc:Infrastructure	uc:Watercraft	uc:HumanitarianAssistanceEvent	uc:WeatherEvent
		uc:InfrastructureEvent	

1.2 UCore and the Army Net-Centric Data Strategy

UCore is designed not only to support messaging and the retrieval and analysis of message content. It is built also in such a way as to support interoperability of information systems of a variety of different types. That is, it is built in such a way as to serve as a basis for the construction of more inclusive artifacts that will serve as interoperability corridors tailored to the needs of groups of specialist users.

Against this background, the Army Net-Centric Data Strategy Center of Excellence is supporting experiments to use UCore as the basis for fostering the interoperability of information artifacts created by Communities of Interest (COIs) in the Command and Control (C2) and other domains. The idea is that such COIs will create new vocabularies tailored to meet their unique requirements and thus go beyond the narrow set of UCore terms. UCore thereby serves as a vehicle which will maintain a joint community perspective by providing an evolving resource of common terms with shared definitions and associated logical resources. The long term goal is that these common terms will create a common reference platform allowing data from diverse COIs to be understood by systems across the DoD and IC. This approach is designed to allow a level of information sharing between unanticipated users and systems and to reduce the time and cost to implement information sharing across the DoD and IC enterprise, while allowing COIs to focus on their community specific needs.

To achieve these ends, UCore will need to accommodate new requirements from its partner agencies, while at the same time remaining faithful to its key principle of providing a small set of essential terms and relations. The latter will however need to be expanded to some degree in order to include those universally understandable terms not so far included. UCore has accordingly established a Configuration Control Board (CCB), whose role is to manage change and versioning in such a way that UCore artifacts remain useable throughout the change lifecycle.

2. UNIVERSAL CORE SEMANTIC LAYER

We describe in what follows an initiative on the part of the Army Net-Centric Data Strategy (ANCDS) [3] Center of Excellence to create an analogous logical infrastructure in support of the UCore endeavor focusing especially on the application of UCore in the creation of domain and COI-specific extensions. The role of logical core is played in this case by the UCore Semantic Layer (UCore SL), version 1.0 which was released on June 15, 2009. UCore SL is the product of work by researchers from the National Center for Ontological Research (NCOR) in Buffalo, New York, with considerable input from the intelligence community under the sponsorship of the Office of the Director of National Intelligence (ODNI) CIO.

Where UCore 2.0 is an XML artifact in which definitions are logically unarticulated, UCore SL is an OWL DL artifact based on logically articulated definitions. UCore SL is designed to work behind the scenes in UCore 2.0 application environments as a logical supplement to the UCore messaging standard. It thus supports UCore's goals by providing additional resources on the side of logical structure. UCore SL offers the entirety of the content UCore-2.0, both taxonomy and relations, in a form which satisfies the needs of users needing enhanced logical resources. It provides for logical decomposition of terms and definitions, genuine reasoning based on the logical content of these definitions, and thereby also enhanced support for the creation of consistent extension modules. UCore SL is being used as a tool for validation of UCore itself and for the generation of proposals for changes and additions both to UCore 2.0 and its extensions. It also serves accessibility of UCore message content to W3C-standard OWL-DL technology.

Where UCore 2.0 provides through its XML framework and controlled vocabulary for syntactic interoperability, UCore SL offers a logically organized vocabulary of terms, relations and definitions which can serve the semantic interoperability of UCore message content.

UCore SL is already helping to provide semantic interoperability in the results of work sponsored by the ANCDS COE on Biometrics and C2 Ontologies carried out by NCOR researchers in Buffalo. We are currently evaluating the ability of UCore SL to provide more powerful reasoning and message-checking capabilities as compared with UCore 2.0 without the added logical support. We are also testing the capacities of UCore SL to provide facilities for enhanced data sharing by helping to ensure that extension modules created by different domains or COIs, for example within the C2 framework, are created in a logically consistent fashion on the basis of logically sound and easily understood definitions.

3. UCORE SL ENHANCHMENTS TO UCORE 2.0

3.1 Mapping UCore SL to UCore 2.0

The UCore SL Taxonomy (version 1.0) consists of 144 terms organized into an is-a (subclass) hierarchy, of which the top two terms are sl:Entity and sl:Event (see table below), corresponding roughly to the continuant and occurrent terms standardly used in upper-level ontologies such as BFO. The UCore SL taxonomy comprehends the entirety of the UCore 2.0 taxonomy in the sense that each one of the 55 terms in the UCore 2.0 taxonomy is mapped to a corresponding UCore SL term. As a result, it is possible to translate UCore 2.0 into UCore SL in order to take advantage of the latter's enhanced logical resources.

UCore SL contains 16 relations, with definitions relying on those provided in BFO [4]. 12 UCore SL relations have counterparts in UCore 2.0. In keeping with the W3C recommended best practice for reuse of OWL resources, ucore:DistinctFrom and ucore:SameAs are not mapped to corresponding UCore SL relations but rather to owl:differentFrom and owl:sameAs respectively. Four other UCore SL relations taken over from BFO do not correspond to any UCore 2.0 relations but are included in order to ensure logical decomposability of definitions. These are: inheres_in, part_of, participates_in and agent_in.

We use the OWL import mechanism to import the UCore 2.0 Taxonomy into the UCore SL taxonomy but not conversely. The import mechanism is uni-directional, which means that the UCore SL ontology contains the content of the UCore 2.0 Taxonomy but not vice versa. In other words, the UCore 2.0 Taxonomy can be used without any reference to UCore SL.

The formal mechanism used to map a UCore SL term to a UCore 2.0 term is the OWL property, owl:equivalentClass [5]. For example, UCore SL asserts that sl:Group is equivalent to uc:CollectionOfThings.(Terms prefixed with 'sl:' are UCore SL terms and terms prefixed with 'uc:' are UCore 2.0 terms.) In other words, every instance of sl:Group is an instance of uc:CollectionOfThings and vice versa. These equivalence statements are the logical crosswalk between UCore SL and UCore 2.0 that make it possible to enhance UCore 2.0 with the logical resources of UCore SL.

Table 2. UCore 2.0 and UCore SL Relations

UCore 2.0 Relations	UCore SL Relations
rdfs:subClassOf	rdfs:subClassOf
ucore:AffiliatedWith	slr:affiliated_with
ucore:CauseOf	slr:cause_of
ucore:Controls	slr:controls
ucore:DistinctFrom	owl:differentFrom
ucore:EmployedBy	slr:employed_by
ucore:HasDestinationOf	slr:has_destination_of
ucore:HasFamilialRelationTo	slr:has_familial_relation_to
ucore:HasOriginOf	slr:has_origin_of
ucore:InvolvedIn	slr:involved_in
ucore:LocatedAt	slr:located_at
ucore:OccursAt	slr:occurs_at
ucore:SameAs	owl:sameAs
ucore:SubordinateTo	slr:subordinate_to
ucore:WorksAt	slr:works_at
	slr:agent_in
	slr:inheres_in
	slr:part_of
	slr:participates_in

The UCore 2.0 Taxonomy is relatively flat and semantically weak. For example, it contains no disjointness axioms, something which is essential to indentifying inconsistencies in an ontology. Two classes are declared to be disjoint if they share no instances in common. For example, Person and MilitaryEvent are in theory disjoint classes. That said, the UCore 2.0 Taxonomy lacks the logical resources to detect a case where something is asserted to be both an instance of Person and MiltaryEvent. The only way to detect this error would be through manual review. The problem is that information artifacts are large and complex and no amount of manual review will catch all the errors. This is why building systems that support automated reasoning are so important.

In UCore SL the following equivalence statements are made:

sl:Group ≡ uc:CollectionOfThings

sl:GroupOfOrganizations ≡ uc:GroupOfOrganizations

sl:GroupOfPersons ≡ uc:GroupOfPersons.

Also, sl:GroupOfOrganizations and sl:GroupOfPersons are subclasses of sl:Group and are disjoint with one another. A few consequences of this are: 1) every instance of sl:GroupOfPersons is an instance of sl:Group, 2) every instance of sl:GroupOfOrganizations is an instance of sl:Group, and 3) no instance of sl:GroupOfPersons is an instance of sl:GroupOfOrganizations. On the UCore 2.0 side of things, uc:CollectionOfThings, uc:GroupOfOrganizations and

Table 3. UCore SL Taxonomy

- sl:Entity
  - sl:InformationContentEntity
    - sl:Analysis
    - sl:Objective
    - sl:ObjectiveSpecification
    - sl:Opinion
    - sl:Plan
    - sl:TaskSpecification
  - sl:PhysicalEntity
    - sl:Agent
    - sl:Artifact
      - sl:ArtificialAgent
      - sl:Equipment
      - sl:Facility
      - sl:Sensor
    - sl:Environment
    - sl:GeographicFeature
    - sl:GeospatialBoundary
    - sl:GeospatialRegion
      - sl:AdministrativeDivision
      - sl:ControlFeature
      - sl:CoverageFeature
      - sl:GeopoliticalEntity
      - sl:Route
      - sl:Track
    - sl:Group
      - sl:GroupOfOrganizations
      - sl:GroupOfPersons
    - sl:InformationBearingEntity
      - sl:Database
      - sl:Datafile
      - sl:Document
      - sl:Program
      - sl:Website
    - sl:Infrastructure
    - sl:Materiel
      - sl:Consumable
    - sl:Organization
      - sl:Government
    - sl:PhysicalObject
      - sl:LivingThing
        - sl:Animal
          - sl:Person
        - sl:InfectiousOrganism
        - sl:MicroOrganism
        - sl:Plant
      - sl:Vehicle
    - sl:SpaceRegion
  - sl:Property
    - sl:Capability
    - sl:PhysicalProperty
      - sl:AtmosphericProperty
      - sl:GeographicProperty
      - sl:OceanographicProperty
      - sl:SpaceEnvironmentProperty
    - sl:Role
      - sl:AffiliationRole
      - sl:AgentRole
      - sl:CargoRole
      - sl:ControlFeatureRole
      - sl:ControlledSubstanceRole
      - sl:InformationSourceRole
      - sl:MaterielRole
      - sl:WaypointRole
- sl:Event
  - sl:Act
    - sl:ActOfCommunication
    - sl:ActOfHumanitarianAssistance
    - sl:ActOfObservation
    - sl:CriminalAct
    - sl:ImmigrationEvent
    - sl:LawEnforcementEvent
    - sl:TerroristAct
  - sl:CyberSpaceEvent
  - sl:Danger
  - sl:Disaster
  - sl:EconomicEvent
    - sl:FinancialEvent
  - sl:EnvironmentalEvent
  - sl:Epidemic
  - sl:EvacuationEvent
  - sl:HazardousEvent
  - sl:Incident
  - sl:InfrastructureEvent
  - sl:MigrationEvent
  - sl:MilitaryEvent
  - sl:MissileLaunchEvent
  - sl:NaturalEvent
    - sl:AtmosphericEvent
    - sl:GeographicEvent
  - sl:NaturalEvent (cont.)
    - sl:OceanographicEvent
    - sl:SpaceEnvironmentEvent
  - sl:PlannedEvent
  - sl:PoliticalEvent
  - sl:PublicHealthEvent
  - sl:SecurityEvent
    - sl:NationalSpecialSecurityEvent
  - sl:SocialEvent
  - sl:StructuralCollapse
  - sl:Task
  - sl:TransportationEvent

uc:GroupOfPersons are all sibling classes (i.e. no one class is a subclass of any of the others classes) and no one class is declared to be disjoint with any of the others. So, the fact that something is an instance of uc:GroupOfPersons does not entail that it is also an instance of uc:Group and 2) the fact that something is an instance of both uc:GroupOfPersons and an instance of GroupOfOrganizations does not entail a contradiction.

This situation is improved by the fact that every term in the UCore 2.0 Taxonomy is mapped to a corresponding term in UCore SL. Once the OWL DL reasoner is run, the hierarchical structure and disjointness axioms in UCore SL are transferred over to UCore 2.0. Figure 1 shows the derived subclass statements. The bi-directional arrows indicate that the two classes are equivalent to one another and the uni-directional arrows indicate that one class is a subclass of another class. Post-inference, uc:GroupOfOrganizations and uc:GroupOfPersons are now subclasses of uc:CollectionOfThings. Although not depicted in Figure 1, the reasoner would also detect cases where something was declared to be both an instance of uc:GroupOfOrganizations and uc:GroupOfPersons.

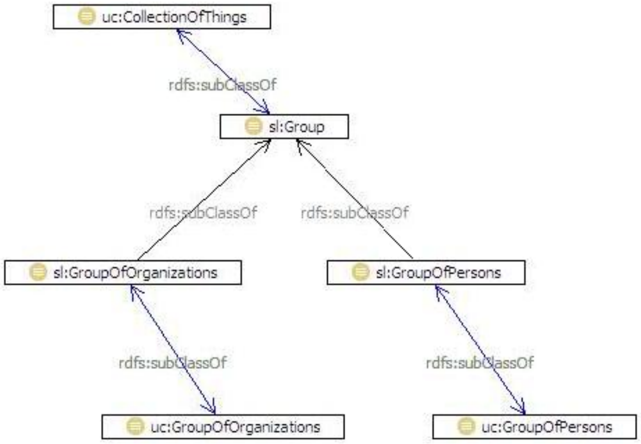


Figure 1. Inferred Subclass Statements

3.2 Extensions of UCore

The purpose of UCore SL is to create a logical infrastructure in support of the UCore endeavor focusing especially on the application of UCore in the creation of domain and COI-specific extensions such as Command and Control (C2), Global Force Management and Cyberspace Operations. To this end, UCore SL can act an overarching framework that can constrain and guide the development of these various domain and COI-specific extensions of UCore.

The mechanism for validating these extensions of UCore is similar to the mechanism described above to map the UCore 2.0 Taxonomy to UCore SL. For example, in order for a C2 ontology to extend UCore SL, the C2 ontology would need to 1) import UCore SL and 2) subsume (directly or indirectly) every C2 term under a UCore SL term. At this point, it is possible to run an inference engine to identify inconsistencies between the C2 ontology and UCore SL. In some cases, it will be necessary to validate multiple domain or COI specific ontologies against one another. Again, UCore SL can facilitate such efforts by providing the logical resources necessary for identifying inconsistencies across ontologies. It should be noted, however, that UCore SL is still in the early phases of development and in future releases more machine-readable definitions (i.e. owl restrictions) will be added in addition to a number of constraints that will check for bad practices in ontology design (e.g. multiple inheritance, use of plurals for class names, classes with only one child, etc.)

Besides providing support for automated reasoning, another way that UCore SL is intended to support extensions of UCore 2.0 is through the use of logical definitions. Every UCore SL term is defined in terms of necessary and sufficient conditions following the Aristotelian schema, which defines each child term ‘A’ in terms of its immediate parent ‘B’ together with the differentia ‘C’ which determines what it is about the Bs which makes them As (as in: a human =def. an animal that is rational). One reason for this is that is instills discipline the Taxonomy development process. Another reason is that it specifies the intended semantics of the term in a human readable format that may serve as a guide for application developers who wish to implement this information artifact in one form or another. The implementation of UCore SL in OWL DL, for instance, is one such case, but there are any number of other ontology and data modeling languages in which one may need to implement UCore SL. To this end, the logical definitions (expressed in controlled way) serve as an application neutral expression of the intended semantics.

3.3 Using UCore SL to Support Reason with UCore Messages

As summarized in [6], we are developing a system which will allow software agents to better understand and reason with UCore-2.0 messaging content in an approach based once again on the logical resources provided by UCore SL. The underlying idea is to treat the XML-labels used in UCore 2.0 messages as annotations for particulars (for instance individual agents) about which these messages contain information. Some particulars are referred to in these messages directly (for instance the military unit that has been given an order to move from place A to place B); others are particulars that must exist for the messages to be correctly interpretable by software agents and whose existence can thereby be indirectly inferred. To make such inferences XML-labels are mapped to ontologies based on UCore SL. Depending on the quality of the mappings and the quality of the associated ontologies, more and better inferences can be made about the portion of reality described in the messages.

We are working on a method to quantify the quality of these mappings and the ontologies in such a way that we can demonstrate that one ontology is to be preferred over another or that one mapping to an ontology is to be preferred over another mapping. By using such quantified measures, we can engineer an evolutionary improvement of ontology resources which can be used across the entire domain of messaging in areas such as C2, where tight integration of messages deriving from disparate sources is required.

3.4 Use of Logical Definitions

The UCore 2.0 definitions are derived primarily from the Concise Oxford English Dictionary (OED), which, while helpful to human users, unfortunately only goes part of the way in specifying the intended meaning of the terms in a fashion useful to computers. A further problem with this approach is that there are numerous cases where the provided definition is not in agreement with UCore's own is-a hierarchy. An example is uc:Animal:

> A non-human organism which feeds on organic matter, has specialized sense organs and nervous system, and is able to move about and to respond rapidly to stimuli. (Derived from OED)

Given that uc:Person is a subclass of uc:Animal, this definition entails that a uc:Person is a non-human organism. In this case, the UCore Configuration Control Board (CCB) has agreed to remove "non-human" from the definition of Human.

Other examples of UCore 2.0 definitions are:

- uc:GroupOfPersons =def A number of people located, gathered, or classed together. (Derived from OED)
- uc:Organization =def An organized body of people with a particular purpose, e.g. a business or government department. (Verbatim from OED)
- uc:PoliticalEntity =def An organized governing body with politcal responsibility in a given geographic region. (Derived from OED)

The definition of 'Organization' does not make it clear whether or not organizations are groups of persons. The definition of 'PoliticalEntity' suggests that it should be a subclass of 'Organization', but this is not reflected in the UCore 2.0 Taxonomy.

UCore SL, in contrast, utilizes the structure of the Taxonomy in the formulation of its definitions. Examples from UCore SL are:

1. sl:Government =def. An Organization with political responsibility for governing in a specified GeospatialRegion.
2. sl:Organization =def. An Agent that has (1) members which are Agents, (2) one or more Objectives, and (3) MemberRoles (and other AffiliateRoles) which are realized in the pursuit of the Objective or Objectives
3. sl:GroupOfPersons =def. A Group that includes only Persons.

The fact that sl:Government is a subclass of sl:Organization is reflected in both the definition and the Taxonomy (see table 3).

The UCore SL team recommended seven improvements to the UCore taxonomy:

1. AlertEvent should be a subclass of CommunicationEvent,
2. WeatherEvent should be a subclass of NaturalEvent,
3. ExerciseEvent should be a subclass of PlannedEvent,
4. FinancialEvent should be a subclass of EconomicEvent,
5. FinancialInstrument should be a subclass of Document,
6. CyberAgent should be a subclass of Agent. But the taxonomy does not contain the term Agent. The taxonomy needs to include Agent,
7. PoliticalEntity should be a subclass of Organization.

These changes should provide clarity to developers seeking to determine which term to use from the taxonomy. The added hierarchical structure effectively groups related terms together, providing developers with a clearer understanding of the available choices.

4. CONCLUSION

UCore SL, an ontology-based supporting layer for UCore, is designed to work behind the scenes in UCore 2.0 application environments as a logical supplement to the UCore messaging standard. UCore SL builds upon previous work in the biomedical domain on creating consistent extensions on the basis of a common core ontology in order to serve interoperability. UCore SL provides the logical resources for the UCore initiative to do this work.

UCore SL is currently in the beta phase of development with several current and potential users who are testing it in their application environments and providing valuable feedback in order to help improve future versions of UCore-SL. In order for UCore SL to succeed, it is necessary to develop a vital user community around UCore SL, one where multiple extension ontologies are subjected to rigorous logical analysis and testing, linked together in a computable way and used to annotate large quantities of data.

5. ACKNOWLEDGMENTS

Thanks are due to Deborah Nichols, Randy Dipert, Ron Rudnicki, and James Schoening.

6. REFERENCES

[1] http://ucore.gov/.

[2] http://www.gcn.com/Articles/2009/06/15/Reality-Check-commentary-UCore-info-sharing.aspx.

[3] Army Net-Centric Data Strategy (ANCDS). http://data.army.mil/

[4] B. Smith, W. Ceusters, B. Klagges, et al., "Relations in biomedical ontologies," Genome Biology, vol. 6, 2006, R46, 2005.

[5] http://www.w3.org/TR/owl-ref/.

[6] S. Manzoor, W. Ceusters, B. Smith. "Referent Tracking for Command & Control Messaging Systems," *submitted to* Ontology for the Intelligence Community, Fairfax, Virginia, October 21-22, 2009.

Performance Measures of Agility for Mobile Robots

Alan Bowling and Shih-Chien Teng
Department of Mechanical and Aerospace Engineering
University of Texas at Arlington, USA
bowling@uta.edu, shih.teng@mavs.uta.edu

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Modeling techniques

General Terms

DESIGN, PERFORMANCE

Keywords

Performance measures

1. INTRODUCTION

Agility refers to a system's ability to change some aspect of its current situation. For example, consider *agile manufacturing* which concerns the ability to *change a particular manufacturing setup* from one product to another without much downtime [3, 8, 25]. In aeronautics, several scalar metrics have been proposed for *aircraft agility* which together measure an aircraft's ability to *change its speed and direction* [21, 1, 2].

In robotics, it is difficult to find a definition of agility or any scalar metrics for measuring it, although several works refer to it. This is true for all robots and is especially true for legged robots. The proposed definitions are often more implicit than explicit, but all discuss the robot's ability to change some aspect of its situation or state. In examining the field of legged robots, it is claimed that the robot's agility stems from its mechanical design, control, or both, as summarized in the following two paragraphs.

Design: Several works claim agility by design, including [17] because the multi-legged, segmented robot can change its overall body configuration allowing it to follow the terrain's contours. An innovative leg design is discussed in [19] which is attributed to the hexapod robot's agility discussed in terms of its speed and its ability to climb obstacles. Similar work discusses the effect of actuation efficiency on agility in terms a multi-legged robot's speed and ability to climb over obstacles [18]. In [16] an actuation scheme that allows a biped to perform acrobatic movements, exemplifying agility, is explored. In [10] an analysis of the robots ability to change the configuration of its legs was considered an indicator of agility. A quadruped climbing robot in [24] is claimed to be agile because of its ability to climb and change direction. A turning criteria, or direction change, at a fairly high speed is used in [15] for examining the agility of a wall climbing robot. The design of a climbing robot in [22] also mentions agility where the design is evaluated mainly in terms of its speed. Other design studies discuss several issues, speed being a central one, but it is unclear how agility was quantified [12, 11]. The work in [14] analyzes acceleration capability as a measure of agility, similar to what is used herein, except that a polytope characterization is used.

Control: In [7] agility in a hexapod micro-robot is obtained through fine tuning the control to increase its speed and ability to change direction. A review of monopod robots [23] mentions agility only in the abstract, but it has been claimed that monopods are agile because their hopping motion allows them to change directions at each ground contact and to perform acrobatic movements [20]. A controller for executing a self-righting motion, which could be considered acrobatic or agile, was developed in [9]. In [13] a genetic algorithm was used to evolve a controller to execute *dynamic maneuvers* consisting of high-speed turns and a running jump.

Nearly all of the works discussed in the "Design" and "Control" sections above include the word "agile" or "agility" somewhere in the paper, but only [14] presents a characterization to describe it. Herein the definition of agility proposed in [4] is used which measures a legged robot's ability to *change its velocity.* Acceleration is defined as the rate of change of velocity, thus the robot's ability to accelerate itself can be used to measure agility. Agile robots can change their velocity abruptly by generating high levels of acceleration.

However, it is often difficult to intuitively understand agility in terms of acceleration capability. Thus the effort here is to build a bridge between acceleration capability analysis and more common measures of motion ability for mobile systems in order to show the correlation between them. The metric considered is the *time to maximum velocity.* The study is performed using an analysis of acceleration capability, referred to as the dynamic capability equations (DCEs), along with simulation techniques in order to determine values for these metrics. A hexapod and the tripod gait are used to illustrate these ideas. This system is chosen because it is simple to generate joint trajectories resulting in the desired locomotion. However, there are some interesting challenges

PERMIS'09 September 21-23, 2009, Gaithersburg, Maryland, USA
Copyright 2009 ACM 978-1-58113-000-0/00/0004 ...$10.00.

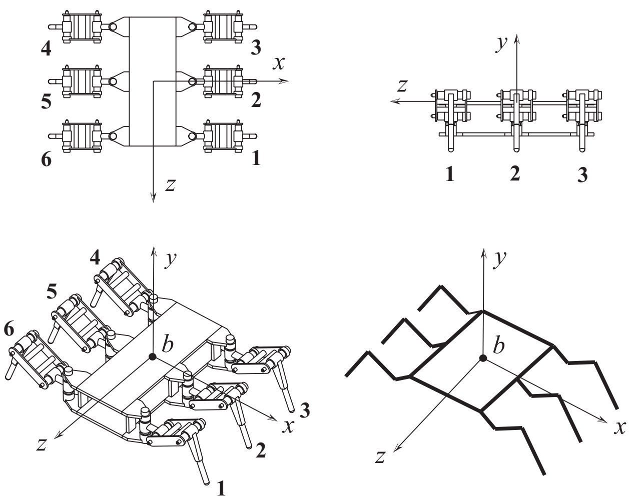


Figure 1: The Hexapod.

involved in actually developing these simulations. These difficulties give one an appreciation for using a performance metric or characterization, such as the DCEs, to aid in the design as opposed to evaluating the metrics using a simulation.

2. THE SCENARIO

The goal of this work is to examine the effects of acceleration capability on locomotion. This effect is investigated in simulation using a model of the hexapod shown in Fig. 1 as a test subject. In these test cases, the hexapod uses the tripod gait to locomote, shown in Fig. 2. This gait is accomplished by placing the odd numbered feet on the ground while lifting the even numbered feet in the air. The odd tripod is then swept forward which moves the body forward. The even tripod is placed on the ground and the odd tripod is lifted off of the ground. The even tripod then sweeps backward and the cycle is repeated. The same cycle can be used for walking straight or turning.

The hexapod's acceleration capability is affected by several different factors including its inertial properties, configuration, ground contact forces, and actuator torque capacity. In order to examine how these factors affect its locomotion, it is necessary to simulate the applied forces. This means that a controller which can generate the tripod gait has to be developed, and the contact forces have to be applied to the dynamic model in a manner that is physically meaningful. These developments created some interesting challenges in assessing how acceleration capability affects the turning radius and achievement of the robot's top speed. The next sections discuss the elements of the simulation in detail.

3. THE SIMULATION

A hybrid dynamic simulation approach was used to deal with the intermittent contact between the hexapod's feet and the ground. The equations of motion for the hexapod can be expressed as

$$A(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{b}(\dot{\mathbf{q}},\mathbf{q}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\Gamma} = \mathsf{G}^T \boldsymbol{\Upsilon} \qquad (1)$$

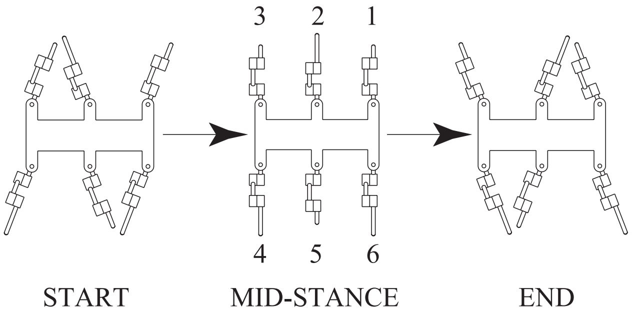


Figure 2: The Tripod Gait.

where $\mathbf{q} \in \mathbb{R}^{24}$ contains the generalized coordinates, and $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are its time derivatives of velocity and acceleration. The vectors $\mathbf{b}(\dot{\mathbf{q}},\mathbf{q}) \in \mathbb{R}^{24}$, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^{24}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{24}$, $\mathbf{F} \in \mathbb{R}^{c}$, and $\boldsymbol{\Upsilon} \in \mathbb{R}^{18}$ contain velocity, gravity, actuator torque, external, and actuator forces. The matrices $A(\mathbf{q}) \in \mathbb{R}^{24\times 24}$, $\mathsf{G} \in \mathbb{R}^{24\times 18}$, $J_c \in \mathbb{R}^{24\times c}$ are the mass, actuator transmission, and contact Jacobian matrices; c is the number of contact constraints.

3.1 Contact Forces

When a foot touches the ground, no-slip, no rebound, point contact is assumed. Thus when a foot touches the ground its velocity is set equal to zero. This can be accomplished in several ways, but the details of what was done here are given in [5]. The important thing to note is that (1) does not include contact constraints. Thus they will have to be included in some manner if (1) is used for purposes other than numerical integration.

3.2 Actuator Torques

The computed torque method was used to execute the tripod gait. Since a turning motion is required, the goal is to avoid calculation of the inverse kinematics associated with a turn. Thus an operational space approach was followed. However, the relations in (1) represent an under actuated system. Since the contact constraints are applied during the numerical integration process, a control command which accounts for the effect of the contact constraints is needed.

The constraints are found using a Jacobian of the form

$$\boldsymbol{\vartheta} = \left[\boldsymbol{v}_b^T \ \ \boldsymbol{\omega}_b^T \ \ \boldsymbol{v}_1^T \ \ \cdots \ \ \boldsymbol{v}_6^T \right]^T = J\,\dot{\mathbf{q}} \qquad (2)$$

where $\boldsymbol{v}_b$ and $\boldsymbol{\omega}_b$ are the translational and rotational velocities of the torso and $\boldsymbol{v}_i$ are the translational velocities of the contact point on each foot. Without a loss of generality, assume that the odd tripod is in contact with the ground:

$$\boldsymbol{\vartheta}_f = \left[\boldsymbol{v}_b^T \ \ \boldsymbol{\omega}_b^T \ \ \boldsymbol{v}_2^T \ \ \boldsymbol{v}_4^T \ \ \boldsymbol{v}_6^T \right]^T = J_f\,\dot{\mathbf{q}} \qquad (3)$$

$$\mathbf{0} = \boldsymbol{\vartheta}_c = \left[\boldsymbol{v}_1^T \ \ \boldsymbol{v}_3^T \ \ \boldsymbol{v}_5^T \right]^T = J_c\,\dot{\mathbf{q}} = Q\,R\,E^T\,\dot{\mathbf{q}} \quad (4)$$

$$= Q\,[\,R_c \ \ R_f\,]\,E^T\dot{\mathbf{q}} = Q\,(R_c\,\dot{\mathbf{q}}_c + R_f\,\dot{\mathbf{q}}_f) \quad (5)$$

$$\mathbf{0} = J_c\,\ddot{\mathbf{q}} + \dot{J}_c\,\dot{\mathbf{q}} \qquad (6)$$

where $Q \in \mathbb{R}^{24\times 24}$ and $R \in \mathbb{R}^{24\times c}$ are obtained from the QR decomposition of J_c, and $R_c \in \mathbb{R}^{c\times c}$ is upper triangular and full rank. Using (5) yields

$$\dot{\mathbf{q}} \;=\; E\begin{bmatrix} -R_c^{-1}R_f \\ \mathrm{I} \end{bmatrix}\dot{\mathbf{q}}_f \;=\; P\,\dot{\mathbf{q}}_f \tag{7}$$

$$\ddot{\mathbf{q}} \;=\; P\,\ddot{\mathbf{q}}_f + E\begin{bmatrix} -R_c^{-1}Q^T\dot{J}_c\;\dot{\mathbf{q}} \\ 0 \end{bmatrix} = P\ddot{\mathbf{q}}_f + \dot{P}\dot{J}_c\dot{\mathbf{q}} \tag{8}$$

Since the feet stick to the ground and do not rebound we know that $\boldsymbol{\vartheta}_c = \dot{\boldsymbol{\vartheta}}_c = \mathbf{0}$ so $\boldsymbol{\vartheta}_f$ is used to define the operational space:

$$\boldsymbol{\vartheta}_f \;=\; J_f\;\dot{\mathbf{q}} \;=\; J_f P\;\dot{\mathbf{q}}_f \tag{9}$$

$$\dot{\boldsymbol{\vartheta}}_f \;=\; J_f\;\ddot{\mathbf{q}} + \dot{J}_f\dot{\mathbf{q}} \;=\; J_f P\ddot{\mathbf{q}}_f + J_f\dot{P}\dot{J}_c\dot{\mathbf{q}} + \dot{J}_f\dot{\mathbf{q}}\;. \tag{10}$$

All of the velocity relationships above have a dual involving static forces which can be expressed as

$$\left(P^TP\right)^{-1}P\;\dot{\mathbf{q}} = \dot{\mathbf{q}}_f \quad\longrightarrow\quad \boldsymbol{\Gamma} = P^T\left(P^TP\right)^{-T}\boldsymbol{\Gamma}_f \tag{11}$$

and from (1)

$$\left(\mathsf{G}\mathsf{G}^T\right)^{-1}\mathsf{G}\,\boldsymbol{\Gamma} \;=\; \boldsymbol{\Upsilon}\;. \tag{12}$$

The relations in (7)-(12) are used to transform the computed torque method into operational space as

$$\dot{\boldsymbol{\vartheta}}_d + K_v\left(\boldsymbol{\vartheta}_d - \boldsymbol{\vartheta}_f\right) + K_p\left(\mathbf{x}_d - \mathbf{x}\right) = \boldsymbol{\Gamma}_f^* \tag{13}$$

$$P^T\left(AP\left((J_fP)^{-1}\left(\boldsymbol{\Gamma}_f^* - J_f\dot{P}\dot{J}_c\dot{\mathbf{q}} - \dot{J}_f\dot{\mathbf{q}}\right)\right) + \mathbf{b} + \mathbf{g}\right) = \boldsymbol{\Gamma}_f \tag{14}$$

$$\left(\mathsf{G}\mathsf{G}^T\right)^{-1}\mathsf{G}P\left(P^TP\right)^{-T}\boldsymbol{\Gamma}_f \;=\; \boldsymbol{\Upsilon} \tag{15}$$

where $\dot{\boldsymbol{\vartheta}}_d$, $\boldsymbol{\vartheta}_d$, and $\mathbf{x}_d$ represent the desired trajectories for the torso and the feet; see (3). The terms K_p and K_v are diagonal gain matrices. Note that $\mathbf{x}$ contains a set of Euler angles to represent the arbitrary orientation of the torso. It is possible to find other forms of (15), but the one given is less likely to involve singular matrix products.

The different postures in the tripod gait are considered as repeating via points for the desired trajectory [6]. The via points are connected using cubic splines with zero velocity at the start and end of each segment. The turning motion is obtained by specifying the desired orientation of the torso.

3.3 Performance Analysis

The DCEs are used to analyze the robot's performance. A recent paper gave an extensive discussion of the DCEs for legged robots in contact with the environment [4]. The resulting characterization describes how well a legged robot can use ground contact to accelerate itself. The performance analysis is too involved to fully present here so only the highlights will be given.

The goal is to determine to compare the torque required to achieve desired motions, with the amount of torque available. In these simulations the available torque is determined by the motors and has the form of a set of inequalities

$$\boldsymbol{\Upsilon}_{lo} \;\leq\; \boldsymbol{\Upsilon} \;\leq\; \boldsymbol{\Upsilon}_{hi}\;. \tag{16}$$

In this work, each element in $\boldsymbol{\Upsilon}_{lo}$ and $\boldsymbol{\Upsilon}_{hi}$ is a constant. After the contact constraints are applied to the equations of motion in (1), they are substituted into (16) to obtain the governing equations for the performance analysis. Further information on this analysis is provided in [4].

The analysis is carried out in terms of *balanced quantities* defined as

$$\dot{\boldsymbol{v}}_b^T\dot{\boldsymbol{v}}_b \;=\; |\dot{v}_b|^2 \qquad\qquad \dot{\boldsymbol{\omega}}_b^T\dot{\boldsymbol{\omega}}_b \;=\; |\dot{\omega}_b|^2 \tag{17}$$

where $|\dot{v}_b|$ is the balanced translational acceleration of the torso. The leftmost relation in (17) represents a sphere with a radius of $|\dot{v}_b|$. Note the difference between the radius $|\dot{v}_b|$ and the magnitude of the vector $\dot{\boldsymbol{v}}_b$ which is $\|\dot{\boldsymbol{v}}_b\|$. The idea is to find the maximum values of the balanced quantities subject to the (1) and (16). For a given configuration and contact state, an analytic solution to this problem exists, and it is the result of the DCE analysis referred to as the dynamic capability hypersurface.

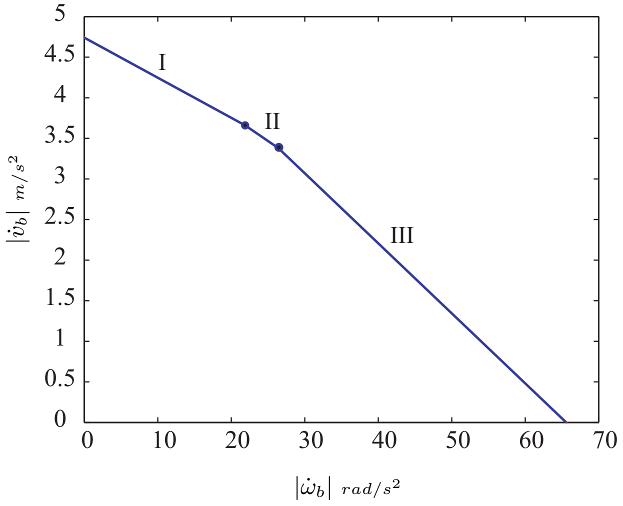


Figure 3: Performance Curve.

A curve from the hypersurface is shown in Fig. 3. It shows the combinations of translational and rotational acceleration of the torso that is guaranteed to be achievable in and about any direction.

A key feature of this analysis is that it can provide the acceleration capability in a particular direction, by defining balanced quantities such as

$$\dot{v}_{b_1}^2 \;=\; |\dot{v}_{b_1}|^2\;. \tag{18}$$

This information is used in the example of Sec. 4.

4. TIME TO VELOCITY

In the automobile industry it is common to quote the time it takes to accelerate from 0mph to 60mph as a measure of performance. This type of measure is also examined for a hexapod with a slightly different interpretation. These test case were run for a set of trajectories which implement the tripod gait. The trajectories for the different phases of the gait were encoded as cubic splines with a start and final time to completion. Therefore the amount of time to complete a cycle of the tripod gait was specified.

A fast and a slow tripod gait were used with prescribed cycle times of 0.39s and 1.2s. There were also two actuator types used, a weak and strong one whose peak torques

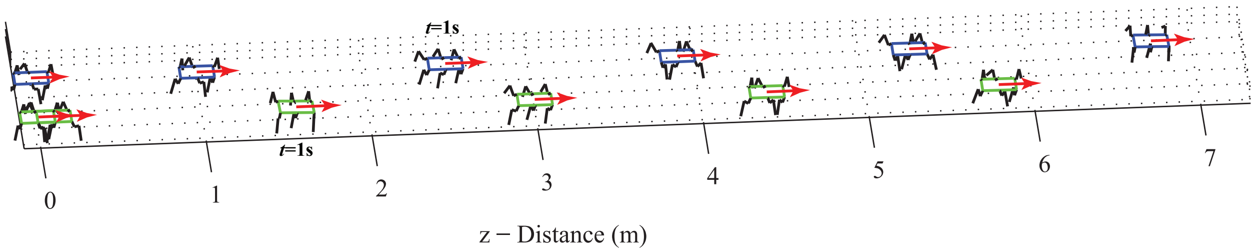


Figure 4: Low Acceleration Capability Hexapods in a 2.5 **Second Race. Motion capture is taken at** 0.5**s intervals. Both hexapods have a maximum torque of** 500**Nm. The top and bottom hexapods use the fast and slow trajectories. The top hexapod's actuators saturate, limiting its top speed. The time label at** $t = 1$**s shows the relative position of the hexapods. The length of the arrow extending from each hexapod indicates its acceleration capability in the forward direction; both hexapods have about the same acceleration capability.**

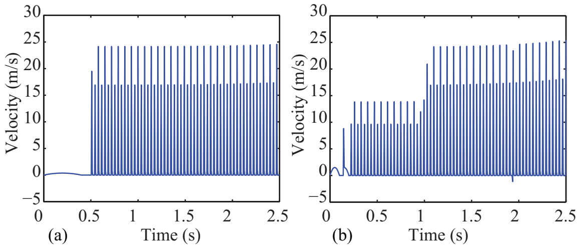


Figure 5: Velocity in the Forward Direction with Peak Torque 500**Nm. (a) Hexapod executing the slow trajectories, and (b) the fast trajectories.**

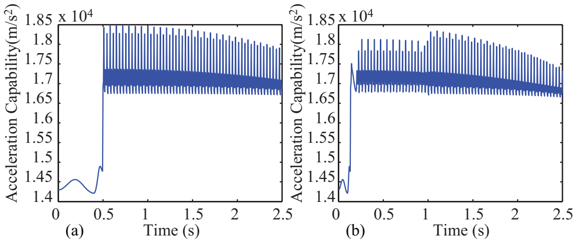


Figure 6: Acceleration Capability in the Forward Direction with Peak Torque 500**Nm. (a) Hexapod executing the slow trajectories, and (b) the fast trajectories.**

were 500Nm and 1000Nm. If the controller commands more torque than is available, the actuator is clipped to the maximum torque, as would occur in the real world. This clipping prevents the robot from asking for an infinite amount of torque, but also limits its acceleration capability and therefore its speed.

This is illustrated in Fig. 4 where both hexapods have actuators with a peak torque of 500Nm, but the top hexapod is commanded to perform the fast trajectories while the lower one performs the slow ones; the hexapods are identical. The top hexapod moves faster than the lower one but it reaches a maximum speed of about 25m/s as shown in Fig. 5b. The fast trajectories push the hexapod to move faster than this, but its torque limitations prevent it.

Also notice in Fig. 5 that the speed varies greatly during the tripod gait, so even if the hexapod has a large amount of acceleration capability, as shown in Fig. 6, it cannot utilize all of it all of the time due to the trajectories. Both hexapods have roughly the same amount of acceleration capability in the forward direction, as indicated by the arrows in Fig. 4 and in Fig. 6. The top hexapod has a little less because it achieves higher velocities which create velocity forces that require torque to overcome.

The lower hexapod gets of to a slow start, in Fig. 5a, but quickly approaches its maximum velocity. If it had not had this difficulty starting off, it would have kept pace with the top hexapod. However, notice that the top hexapod also had some difficulty getting started in Fig. 5b.

In Fig. 7 the peak torque of the top hexapod is increased to 1000Nm. The bottom hexapod in Figs. 4 and 7 are identical. The increase in peak torque gives the top hexapod more acceleration capability thereby allowing it to achieve the higher speeds required to execute the fast trajectories. Comparing Figs. 4 and 7 notice that the top hexapod moves farther in the 2.5 second run time. Fig. 8b shows that the top hexapod did achieve a higher velocity 35m/s, as compared to the top speed of 25m/s in Fig. 5b. Indeed its acceleration capability in the forward direction was also larger, as shown in Fig. 9b, than in the earlier race, Fig. 6b.

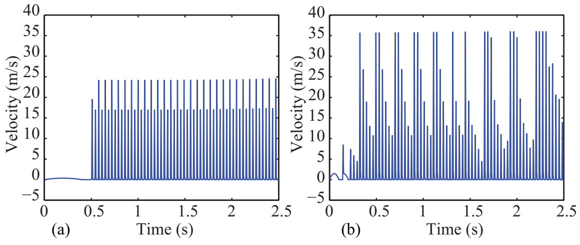


Figure 8: Velocity in the Forward Direction. (a) Hexapod executing the slow trajectories with peak torque of 500**Nm, and (b) the fast trajectories with peak torque** 1000**Nm.**

These two cases show how increasing the acceleration ca-

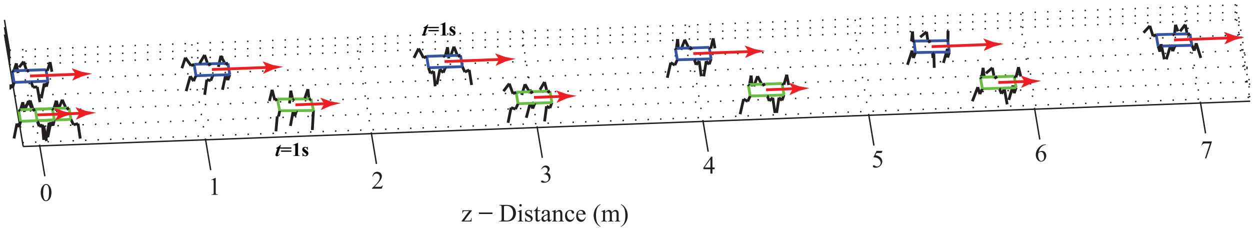


Figure 7: High vs. Low Acceleration Capability Hexapod in a 2.5 Second Race. Motion capture is taken at 0.5s intervals. The top hexapod has maximum torque of 1000Nm while the bottom one has 500Nm. The higher acceleration capability allows the top robot to move farther in 2.5 seconds than in the previous race. The time label at $t = 1$s shows the relative position of the hexapods. The length of the arrow extending from each hexapod indicates its acceleration capability in the forward direction; the longer arrows show that the fast hexapod has more acceleration capability.

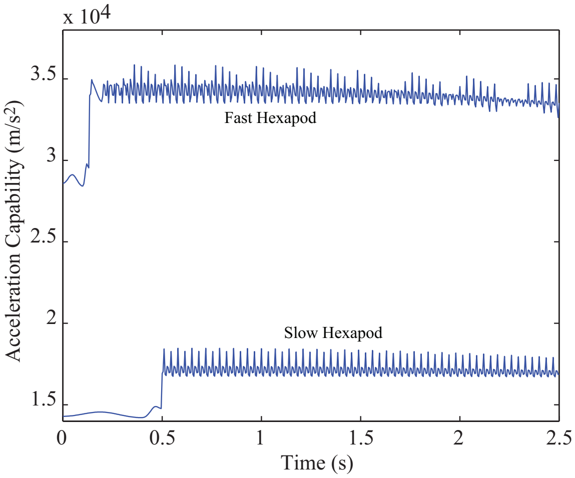


Figure 9: Acceleration Capability in the Forward Direction. (a) Hexapod executing the slow trajectories with peak torque of 500Nm, and (b) the fast trajectories with peak torque 1000Nm.

pability can improve the performance of a hexapod. The original hexapod can accelerate from 0-25m/s in 2.5s, but after an increase in acceleration capability it could accelerate from 0-35 in 2.5s. Here the increase in acceleration capability is due to increased actuator size. It is also possible that the pattern of locomotion could be changed to increase its acceleration capability to achieve the same effect.

What should be noticeable is the number of factors that come into play in the simulation which affect the results. Clearly a change in gait pattern would affect the performance. A performance metric which requires a simulation to evaluate it would have to be run each time any of these factors changes in order to evaluate the performance. This is not a simple task when considering legged robots. It is actually simpler and faster to use the analytic solution encoded in the DCE to design in the desired performance. A simulation will most likely be needed in the end to check the final conceptual design, but it need not slow the design process down when a performance measure is available.

5. CONCLUSIONS

This paper showed that acceleration capability has a significant impact on a commonly used performance metric, the time to a particular speed, which allows it to be used in design in lieu of a simulation. This is advantageous because of the difficulty involved in simulating the locomotion of legged robots. There are several factors which will affect the simulation results, which are more easily addressed using a performance characterization like the DCE. Future work involves examining more agile motions like turning in order to consider rotational performance.

6. REFERENCES

[1] C. W. Alcorn, M. A. Croom, M. S. Francis, and H. Ross. The x-31 aircraft: Advances in aircraft agility and performance. *Progress in Aerospace Sciences*, 32(4):377–413, August 1996.

[2] G. Avanzini, G. de Matteis, and L. M. de Socio. Analysis of aircraft agility on maximum performance maneuvers. *Journal of Aircraft*, 35(4):529–535, July-August 1998.

[3] J. Barata and L. M. Camarinha-Matos. Coalitions of manufacturing components for shop floor agility - the cobasa architecture. *International Journal of Networking and Virtual Organizations*, 2(1):50–77, 2003.

[4] A. Bowling. Dynamic performance, mobility, and agility of multi-legged robots. *ASME Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 128(4):765–777, December 2006.

[5] A. Bowling, D. M. Flickinger, and S. Harmeyer. Energetically consistent simulation of simultaneous impacts and contacts in multibody systems with friction. *Multibody System Dynamics*, 22(1):27–45, August 2009.

[6] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Publishing Company, second edition, 1989.

[7] F. De Ambroggi, L. Fortuna, and G. Muscato. Plif: Piezo light intelligent flea - new micro-robots controlled by self-learning techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1767 – 1772, 1997.

[8] I. H. Garbie, H. R. Parsaei, and H. R. Leep. A novel approach for measuring agility in manufacturing firms. *International Journal of Computer Applications in Technology*, 32(2):95–103, September 2008.

[9] A. Greenfield, U. Saranli, and A. A. Rizzi. Solving models of controlled dynamic planar rigid-body systems with frictional contact. *The International Journal of Robotics Research*, 24(11):911–931, November 2005.

[10] B. Han, Q. Luo, Q. Wang, and X. Zhao. A research on hexapod walking bio-robot's working space and flexibility. In *2006 IEEE International Conference on Robotics and Biomimetics, ROBIO 2006*, pages 813 – 817, 2006.

[11] T. Ho, S. Choi, and S. Lee. Development of a biomimetic quadruped robot. *Journal of Bionic Engineering*, 4(4):193 – 199, 2007.

[12] B. Kennedy, H. Agazarian, Y. Cheng, M. Garrett, G. Hickey, T. Huntsberger, L. Magnone, C. Mahoney, A. Meyer, and J. Knight. Lemur: Legged excursion mechanical utility rover. *Autonomous Robots*, 11(3):201 – 205, 2001.

[13] D. P. Krasny and D. E. Orin. Evolution of dynamic maneuvers in a 3d galloping quadruped robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1084–1089, 2006. Orlando, Florida, USA.

[14] J. Lee, H. Shim, and D. Kim. Mobility and agility analysis of walking robot. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 1349 – 1354, Piscataway, NJ 08855-1331, United States, 2008.

[15] M. P. Murphy and M. Sitti. Waalbot: An agile small-scale wall-climbing robot utilizing dry elastomer adhesives. *IEEE/ASME Transactions on Mechatronics*, 12(3):330 – 338, 2007.

[16] A. A. F. Nassiraei, S. Masakado, T. Matsuo, T. Sonoda, I. Takahira, H. Fukushima, M. Murata, K. Ichikawa, K. Ishii, and T. Miki. Development of an artistic robot "jumping joe". In *IEEE International Conference on Intelligent Robots and Systems*, pages 1720 – 1725, 2006.

[17] A. Preumont, P. Alexandre, I. Doroftei, and F. Goffin. Conceptual walking vehicle for planetary exploration. *Mechatronics*, 7(3):287 – 296, 1997.

[18] R. Quinn, G. Nelson, R. Bachmann, and R. Ritzmann. Toward mission capable legged robots through biological inspiration. *Autonomous Robots*, 11(3):215 – 220, 2001.

[19] R. D. Quinn, G. M. Nelson, R. J. Bachmann, D. A. Kingsley, J. T. Offi, T. J. Allen, and R. E. Ritzmann. Parallel complementary strategies for implementing biological principles into mobile robots. *International Journal of Robotics Research*, 22(3-4):169 – 186, 2003.

[20] M. H. Raibert. Hopping in legged systems-modeling and simulation for the two-dimensional one-legged case. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-14(3):451–463, 1984.

[21] G. W. Ryan, III and D. R. Downing. Evaluation of several agility metrics for fighter aircraft using optimal trajectory analysis. *Journal of Aircraft*, 32(4):732–738, July-August 1995.

[22] A. Saunders, D. Goldman, R. Full, and M. Buehler. The rise climbing robot: Body and leg design. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6230 I, page 623017(13 pages), 2006.

[23] A. Sayyad, B. Seth, and P. Seshu. Single-legged hopping robotics research - a review. *Robotica*, 25(5):587 – 613, 2007.

[24] O. Unver, A. Uneri, A. Aydemir, and M. Sitti. Geckobot: A gecko inspired climbing robot using elastomer adhesives. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2006, pages 2329 – 2335, 2006.

[25] W.-P. Wang. Toward developing agility evaluation of mass customization systems using 2-tuple linguistic computing. *Expert Systems with Applications*, 36(2 Part 2):3439–3447, March 2009.

Measuring Robot Performance in Real-time for NASA Robotic Reconnaissance Operations

Debra Schreckenghost
TRACLabs, Inc
1012 Hercules, Houston, TX 77058
ghost@ieee.org

Terrence Fong
NASA Ames Research Center
Moffett Field, CA 94035
terry.fong@nasa.gov

Tod Milam
TRACLabs, Inc
1012 Hercules, Houston, TX 77058
tmilam@traclabs.com

Hans Utz
Research Institute for Advanced
Computer Science
Moffett Field, CA 94035
hans.utz@nasa.gov

ABSTRACT

Technical advances since Apollo make it possible to perform robotic reconnaissance to gain a better understanding of lunar sites prior to human exploration. NASA is conducting analog field tests to investigate these operations concepts with advanced robots and simulated flight operations. We have developed robot performance monitoring software for use during robotic reconnaissance operations. We measure robot performance by monitoring robot data in real-time and computing robot performance metrics from that data. Metrics are computed for two regimes of flight operations – remote supervision of autonomous robot operations and debrief support after a flight operations shift. In this paper we describe our performance monitoring software, define the metrics we compute, discuss how these metrics are used in flight operations, and summarize results from recent field tests.

Categories and Subject Descriptors

C.4.3 [**Performance of Systems**]: *Measurement techniques, Performance attributes, Reliability, availability, and serviceability*; I1.2.9 [**Artificial Intelligence**]: Robotics - *Commercial robots and applications, Operator interfaces*

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Operational performance metrics, robotic recon, space robotics.

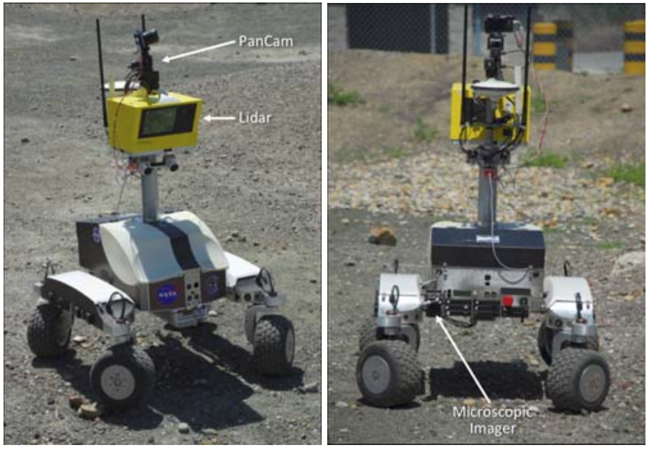
1. INTRODUCTION

NASA’s plan to return humans to the moon raises questions about how to conduct lunar exploration missions to best utilize limited crew resources. Technical advances since Apollo enable the use of robotic systems to complement the human mission. Improved instrumentation makes it possible to perform robotic reconnaissance, or “recon”, to gain a better understanding of lunar sites prior to human exploration. NASA is conducting analog field tests to investigate these operations concepts with advanced robots and simulated flight operations [1].

We define robotic recon as operating a planetary rover under ground, or non-EVA astronaut, control to scout planned sorties prior to EVA activity. Scouting is an essential phase of field work, particularly for geology. Robot instruments provide measurements of the surface and subsurface at resolutions and from viewpoints not achievable from orbit. This surface-level data can then be used to select locations for field work and prioritize targets to improve crew productivity. Robotic recon can be done months in advance, or be part of a continuing planning process during human missions. [1].

We first began studying robotic recon during the June 2008 NASA Human-Robotic System (HRS) project analog field test conducted at the Moses Lake Sand Dunes, WA. During this test, an experimental ground control team located at the NASA Johnson Space Center (JSC) used a K10 planetary robot (Figure 1) to remotely scout a portion of the sand dunes. The data collected during the reconnaissance was then used to develop a plan for crew EVA in the same area. Lessons learned at Moses Lake were subsequently used to improve and validate robotic recon systems during Operational Readiness Tests (ORT) at the NASA Ames Research Center (ARC) in November 2008 and June 2009. Most recently, we conducted an experimental assessment of robotic recon as part of the 2009 Desert Research and Technology Studies (D-RATS) analog field campaign at Black Point Lava Flow, AZ.

Figure 1. The K10 planetary rover is equipped with three



instruments for robotic recon: a panoramic imager (PanCam), a 3D scanning lidar, and a terrain facing microscopic imager.

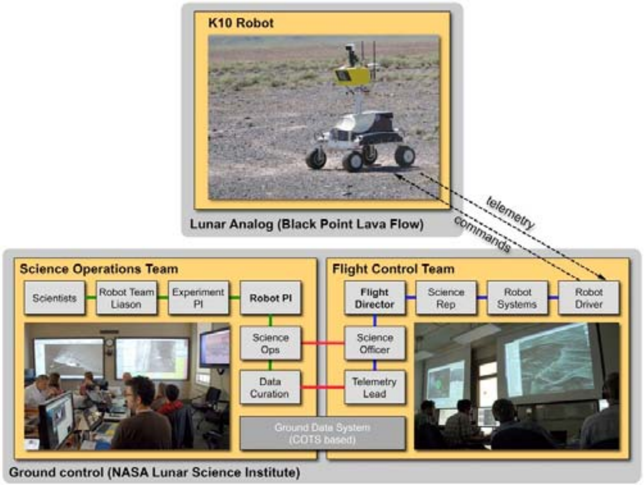


Figure 2. An experimental ground control for robotic recon includes a science operations team and a flight control team. The roles and protocols used by this ground control are a hybrid of Apollo, Shuttle, Space Station, and MER concepts.

For these ORTs and field tests, ground control (Figure 2) was conducted concurrent with robot surface operations. The Science Team builds robot plans to visit features of interest and take instrument readings to determine what features would benefit most from astronaut Extra-Vehicular Activity (EVA). During robotic recon operations, the ground control team reviews robot plans, then uplinks them to the robot. Similar to planetary rovers like JPL's Mars Exploration Rover (MER), the K10 robot immediately begins to execute the uplinked plan. But, unlike MER, ground control supervises the robot as it executes the plan. They continuously monitor robot data in real-time and can intervene using tele-operations if opportunities or problems arise.

To help improve the efficiency and effectiveness of robotic recon operations, we have developed real-time robot performance monitoring software. In the following, we describe our approach, define the performance metrics we compute, discuss how these metrics are used in robot operations, and summarize results from recent field tests.

2. RELATED WORK

The planetary surface environments in which recon robots must operate are highly variable. To maximize the value of the reconnaissance, the robot may be required to make observations in areas that are difficult to access or traverse. For example during the June 2009 ORT at NASA Ames Research Center, the K10 robot operated near its safety limits to access desired geologic features on steep slopes. When communication latencies and bandwidth permit, tele-operating the robot in these conditions can be preferable to autonomous operations. Thus, robotic recon includes autonomous robotic activities interleaved with scheduled tele-operation.

For this operations model, human-robot interaction metrics such as interaction efficiency [2] and neglect tolerance [3] are relevant. Interaction efficiency is improved by minimizing human interaction time. Neglect tolerance is a function of neglect time, the average time a robot can be ignored while keeping performance above some acceptable level. Larger neglect tolerance indicates greater independence in robot operations. We apply these metrics a bit differently for robotic recon, however. Central to these differences is the distinction between *planned* and *unplanned* human interaction.

For lunar reconnaissance operations, minimizing all human interaction time may *not* translate to more efficient robot operations. In fact, it may often be more time and resource efficient to tele-operate the robot in difficult terrain than to operate autonomously. Thus, our objective is to minimize the time spent on *unplanned interventions* (such as anomaly handling).

To do this, we make use of *Mean Time to Intervene* (MTTI) [4], which is the average time spent handling anomalies that interrupt planned robot tasks. Scheduled tele-operations are not included in MTTI. We also compute *Mean Time Between Interventions* (MTBI) as the average time between unplanned interventions. Similar to neglect time, larger MTBI indicates improved human-robot performance for reconnaissance.

We measure the robot's productivity as a function of the time the robot spends on reconnaissance tasks (called *productive time*). We compare productive time to time spent on other tasks (called *overhead time*), such as waiting for a reconnaissance plan or handling problems. One metric we use is *Work Efficiency Index* (WEI) [5], which is the ratio of productive time to overhead time. We also compute the *Percentage of Time on Task* as the productive time normalized to the total elapsed time in the shift.

Performance measures used for the MER robots include the total traverse distance over the lifetime of the robot [6]. Maximizing traverse distance for such exploration is desirable as an indicator of increased productive lifespan for the robot. But longer distances traveled or drive times do not necessarily correspond to better performance during robotic recon. Specifically, some plans may have short traverses interleaved with many data collection tasks. In such cases the robot may be performing optimally even with shorter drive times and less distance traveled.

During the November 2008 ORT, for example, we observed the robot actually traveled greater distances when it was performing less well. The longest distance traveled over a shift corresponds to more human time spent unexpectedly tele-operating the robot out of difficult terrain. Comparing total distance traveled to the distance traveled performing planned tasks reveals this important distinction. Thus for robotic recon, a better indicator of good performance is the minimum distance traveled outside the plan instead of the maximum distance traveled.

Performance measures are more effective for time and resource management if good estimates of expected task times and plan durations are available. These estimates define expected baseline performance useful in interpreting measures computed in real-time. Such information can be used to assess how efficiently the robot accomplished tasks under nominal circumstances and to determine what it is feasible to get done in the time remaining under contingency conditions. Algorithms for estimating task times can benefit from work on metrics for diagnostic and prognostic technologies [7, 8]. Such algorithms address detecting problems that can affect the quality of observations, which could be useful in pruning abnormal observations from computation of typical task times.

3. ROBOT PERFORMANCE MEASURES

3.1 Software Approach

We measure robot performance during reconnaissance operations by monitoring robot data in real-time and computing robot performance metrics from that data. Robot performance is computed remotely for use by flight operations personnel. The same robot data stream used for flight operations are used to compute performance of the robot in real-time. This includes detecting event signatures in data that affect robot performance (e.g., robot in motion). Metric values are displayed on web-based dashboards and plots for use by flight operations. Figure 3 illustrates the key components of the performance monitoring software.

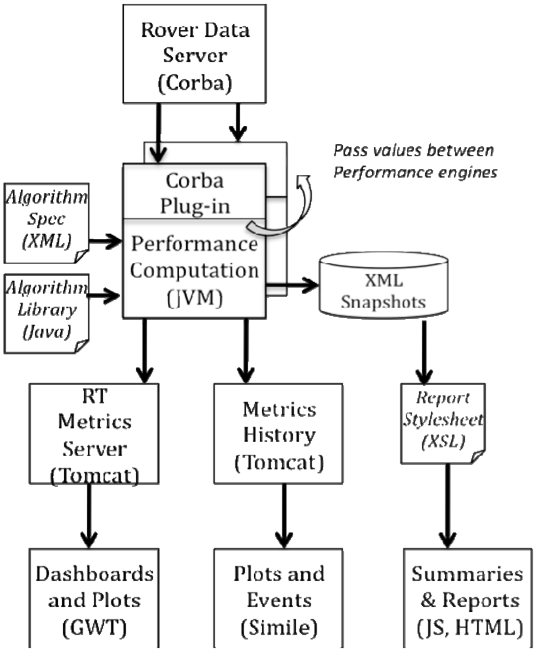

Rover Data Server (Corba)
Pass values between Performance engines
Algorithm Spec (XML)
Corba Plug-in
Performance Computation (JVM)
Algorithm Library (Java)
XML Snapshots
RT Metrics Server (Tomcat)
Metrics History (Tomcat)
Report Stylesheet (XSL)
Dashboards and Plots (GWT)
Plots and Events (Simile)
Summaries & Reports (JS, HTML)

Figure 3. Components of Performance Monitoring Software

Metric algorithms are encoded as Java objects. At run-time algorithms are selected for execution and associated with robot data using configuration files. An instance of the algorithm's class is created for each connection to a robot data item, permitting algorithms to be reused across multiple metrics. Complex algorithms are composed by connecting sequences of simpler algorithms (i.e., the output of one algorithm provides input to another algorithm). Computed metrics are distributed via a real-time data server and displayed as dashboards in a web page. Performance history is provided in two ways: plots of metric values computed over time and debrief reports of summary metrics computed over a shift.

We evaluated our performance monitoring software during the June 2008 HRS field test at Moses Lake Sand Dunes, WA, to assess the feasibility of computing robot performance metrics in real time [9]. The objectives of this evaluation were to identify meaningful robot metrics, to assess whether these metrics can be computed using existing robot data, and to determine the impacts of remoteness on the computation of robot metrics on Earth. Subsequently we supported two ORTs at NASA ARC - one in November 2008 and another in early June 2009. During these ORTs we observed how performance metrics are used in operations. We also assessed how the human team design and protocols impact robot performance, such as robot utilization and robot wait time.

Based on the results from these prior tests, we deployed and tested a revised system during the robotic recon portion of the 2009 D-RATS field test at Black Point Lava Flow, AZ. Specifically, we computed performance metrics in two time regimes – for use by flight controllers during remote real-time flight operations and for use during debrief after an operations shift. To support flight operations, we also provided web-based dashboard displays of performance metrics computed from robot telemetry data. These displays were updated automatically with the latest computed value for performance measures. To support debrief meetings, we took snapshots of metric values at the end of each shift and performed additional computations on these values to produce a debrief report spanning each shift. The debrief report was generated in a web page using the eXtensible Markup Language (XML) stylesheets and Javascript.

3.2 Metrics for Real-time Flight Operations

To investigate the use of performance metrics during real-time operations, we supported two robot ground control positions [1]: the *Flight Director*, who is responsible for managing and coordinating the flight control and the *Robot Operations Coordinator*, who is responsible for the health and status of the K10 robot. In both cases, metrics are used in real-time for time and resource management.

The Flight Director primarily uses performance metrics to manage the use of time. During operations, the Flight Director is concerned with whether the robot is operating normally. This includes both whether the assigned tasks complete successfully and whether these tasks can be accomplished in the allocated time. The questions asked by the Flight Director include: How much time before an ongoing task is completed? How much time it will take to complete the unfinished tasks in the current plan? and How much of the time allocated to a plan has been expended so far? This becomes particularly important when an anomaly occurs and the Flight Director must make choices about whether to abandon some tasks.

To support the Flight Director in assessing and managing the timing of task performance, we provided a set of real-time metrics that measure how long tasks take and that compare these measures to expected performance. Specifically we computed and displayed the following five timers:

1) Plan Timer: The Plan Timer counts down from the expected time it will take to complete a plan while the plan is being executed. The timer is updated each time an update about plan execution status is received until the plan is done. A plan is considered done when all tasks are either successful or aborted. This timer resets when a new plan is uplinked. The expected time is computed by summing the expected time for each planned task. Expected times for sampling tasks are based on typical performance during the ORT and field test. Expected times for traverse tasks are computed for a linear path between waypoints at default robot speed. The algorithm for the Plan Timer is shown below.

$$
\begin{aligned}
&PlanTimer(t) = E(dt_{plan}) - \left(t_{current} - t_{start}\right)\\
&where\\
&E(dt_{plan}) = \exp ected\ time\ to\ complete\ plan\\
&t_{start} = time\ when\ plan\ begins\ to\ execute\\
&t_{current} = time\ of\ \ last\ update\ in\ plan\ status
\end{aligned}
$$

2) Plan Wait Timer: The Plan Wait Timer increments when a plan is NOT active (i.e., no task is active, paused or pending). This includes time spent handling robot anomalies as well as idle time waiting for a new plan. This timer resets when a plan completes. Thus it measures all the “wait” time between plans. The algorithm for the Plan Timer is shown below.

$$
\begin{aligned}
&PlanWaitTimer(t) = PlanWaitTimer(t) + dt_{elapsed}\\
&where\\
&x = elapsed\ time\ between\ adjacent\ timetags\\
&\qquad when\ plan\ not\ active
\end{aligned}
$$

3) Lidar Panorama Timer: The Lidar Panorama Timer counts down from the expected time it will take to complete a Lidar panorama while the lidar is active. The timer is updated each time an update about the Lidar subsystem status is received until the panorama is complete. This timer resets when a new panorama begins. The expected time for the Lidar panorama task is based on typical performance during the ORT and field test. The algorithm for the Lidar Panorama Timer is shown below.

$$
\begin{aligned}
&LidarTimer(t) = E(dt_{panorama}) - \left(t_{current} - t_{start}\right)\\
&where\\
&E(dt_{panorama}) = \exp ected\ time\ to\ complete\ panorama\\
&t_{start} = time\ when\ Lidar\ panorama\ subsystem\ goes\ active\\
&t_{current} = time\ of\ \ last\ update\ of\ \ Lidar\ subsystem\ status
\end{aligned}
$$

4) PanCam Timer: The Panoramic Camera (PanCam) Timer counts up while a PanCam panorama is being taken. For the 2009 field test at Black Point Lava Flow, there are five different types of panorama: (1) Medium Width Image, Low Resolution, (2) Medium Width Image, Medium Resolution, (3) Wide Image, Low Resolution, (4) Wide Image, Medium Resolution, and (5) Narrow Image, High Resolution. Each of these types of panoramas take a different amount of time. Since the field test at Black Point Lava Flow was the first field test where we have used variable width and resolution PanCam imaging, we employ a “count up” timer to collect data on typical task times at each resolution. The algorithm for the PanCam Timer is shown below.

$$
\begin{aligned}
&PanCamTimer(t) = PanCamTimer(t) + dt_{elapsed}\\
&where\\
&x = elapsed\ time\ between\ adjacent\ timetags\\
&\qquad when\ taking\ panorama
\end{aligned}
$$

5) MicroImager Timer: The MicroImager Timer counts up while a MicroImage is being taken. Microimaging takes between 10 and 20 seconds. Because this task is so short, counting down from some expected time was not deemed useful. The algorithm for the MicroImage Timer is shown below.

$$
\begin{aligned}
&Micro\mathrm{Im}ageTimer(t) = Micro\mathrm{Im}ageTimer(t) + dt_{elapsed}\\
&where\\
&x = elapsed\ time\ between\ adjacent\ timetags\\
&\qquad when\ microimaging
\end{aligned}
$$

The Robot Operations Coordinator uses “low-level” performance metrics to manage robot resources such as remaining battery power. Such measures are needed to anticipate when robot maintenance is needed. Some batteries on the K10 robot provide feedback about remaining capacity and some do not. For batteries that are instrumented, we provide displays of the average battery capacity for each battery controller. Each controller manages eight batteries. For batteries that are not instrumented (such as the Lidar battery) we track battery usage (i.e., total runtime of the instrument using the battery) to give the Robot Operations Coordinator an idea of how much capacity remains. Utilization is tracked throughout the shift over multiple sampling intervals. When a Lidar battery is swapped out, we reset Lidar Runt Time. The algorithm used for Lidar Run Time is shown below.

$$
\begin{aligned}
&LidarRunTime(t) = t_{end} - t_{start}\\
&where\\
&t_{start} = time\ when\ Lidar\ panorama\ subsystem\ goes\ active\\
&t_{end} = time\ when\ Lidar\ panorama\ subsystem\ goes\ inactive
\end{aligned}
$$

Finally we compute data communication quality metrics for use by all flight controllers. During the field test at Moses Lake, we observed two days with significantly degraded communication. The metric for communication quality during this field test was a count of the number of times the ground lost communication with the remote robot (called a *data dropout,* or *Loss of Signal[LOS]*) during a support period (for this test we compute a count over the shift). Larger dropout counts indicated more data were unavailable for computation, potentially impacting the accuracy of metric values. The duration of dropouts varied significantly at Moses Lake, however, prompting the definition of a second metric to compute the percentage of a support period (i.e., shift) that was spent without communication. This permits estimating the amount of data not available to flight operations and not included in metrics computed during real-time operations. The algorithms for both these metrics are shown below.

$$
\begin{aligned}
&DataDropout(tp) = \begin{cases} t & if\ \ tp_i - tp_{i-1} > 10\ \mathrm{sec}\\ f & if\ \ tp_i - tp_{i-1} <= 10\ \mathrm{sec}\end{cases}\\
&where\\
&tp_i = time\ of\ \ pose\ message\ i
\end{aligned}
$$

$$
\begin{aligned}
&\%ShiftInLOS(t_i) = \left[TimeInLOS(t_i)/TimeInShift(t_i)\right]*100.0\\
&where\\
&TimeInLOS(t_i) = Elapsed\ time\ in\ LOS\ at\ t_i\\
&TimeInShift(t_i) = Elapsed\ time\ in\ shift\ at\ t_i
\end{aligned}
$$

3.3 Metrics for Shift Debrief

Operationally, the ground control team holds a debrief meeting immediately after each shift (contiguous period of operations). We compute performance metrics over the course of each shift for building a debrief report. The debrief report has three sections: (1) a robot performance summary, (2) an anomaly summary, and (3) an event log. The performance summary provides metrics about robot productivity, task breakdown, and data collected during the shift. The anomaly summary provides metrics about unplanned interventions in robot operations, problems experienced by the robot, and loss of communication. The event log interleaves log notes by the flight team with events detected automatically in the robot telemetry stream.

Three metrics are computed to assess robot productivity – time spent performing planned tasks (called productive time), time spent doing activities other than planned tasks (called overhead time), and the ratio of these two measures (called Work Efficiency Index; [5]). The algorithm used for each of these metrics is shown below.

$$WEI(t_i) = PT(t_i)/OT(t_i)$$

$$where$$

$$PT(t_i) = time\ executing\ planned\ tasks$$

$$OT(t_i) = time\ outside\ planned\ tasks$$

To assess task performance we compute the breakdown across the shift of robot drive time, time taking Lidar, time taking each of the five types of PanCam, and time microimaging. We also measure the number of samples taken by each type of instrument over the shift. Finally, we compute the total distance traveled by the robot during the shift. The algorithms used for these robot performance metrics are shown below.

$$RunTime(t) = \sum_{i=1} t_i - t_{i-1} \quad if\ t_i - t_{i-1} < 10 \text{ sec}$$

$$where$$

$$t_i = time\ of\ pose\ message$$

$$DriveTime(t) = \sum_{i=1} t_i - t_{i-1}$$

$$where$$

$$t_{i-1} = last\ time\ locomotor\ or\ navigator\ went\ active$$

$$t_i = time\ both\ locomotor\ and\ navigator\ go\ inactive$$

$$DistanceTraveled(x,y) = \sum_{i=1} \sqrt{\left[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2\right]}$$

$$if\ d(x,y) > 0.025m$$

$$where$$

$$(x_i, y_i) = current\ rover\ pose\ estimate$$

$$(x_{i-1}, y_{i-1}) = previous\ rover\ pose\ estimate$$

To assess anomalies in robot operations, we detect unplanned interventions in robot operations and use them to compute the Mean Time To Intervene (MTTI) and the Mean Time Between Interventions (MTBI). We show how many of each type of problem intervention occurred during the test, and the percent of the shift spent on each type of intervention. We also summarize communication quality by measuring the total number of LOS periods and the percentage of the shift spent in LOS. The algorithms for MTTI and MTBI metrics are shown below. The algorithms for communication quality were described previously.

$$MTTI(t_i) = TI(t_i)/CI(t_i)$$

$$where$$

$$TI(t_i) = time\ spent\ \text{int}ervening\ upto\ t_i$$

$$CI(t_i) = number\ of\ \text{int}erventions\ by\ t_i$$

$$MTBI(t_i) = TBI(t_i)/CBI(t_i)$$

$$where$$

$$TBI(t_i) = total\ time\ between\ \text{int}erventions\ upto\ t_i$$

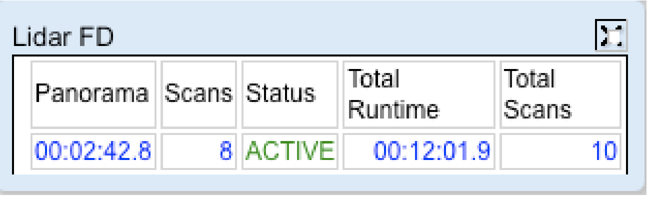
$$CBI(t_i) = number\ of\ periods\ between\ \text{int}erventions\ by\ t_i$$

The event log combines information derived from the robot telemetry stream with log entries made by users. Events detected automatically include (1) Acquisition of Signal (AOS) and Loss of Signal (LOS), (2) instrument sample start and end, (3) problem start and end, and (4) plan uplink, start, and end. Problems reported include (1) emergency stop, (2) critical failure in robot subsystem (locomotor, navigation, or plan executor), (3) joint failure, and (4) navigation position error (usually due to a stuck wheel). These are problems where intervention by a person is likely required to fix the problem. User log entries are made from our real-time display. They are timestamped at the time of log entry. User log entries are distinguished from detected log entries using italics text in the display.

4. RESULTS

The performance monitoring software was used by K10 ground control during the robotic recon portion of the 2009 D-RATS field test at Black Point Lava Flow, AZ. In this section we summarize the results from using these metrics during field test operations.

Instrument timers (Figure 4) were used in a number of ways during real-time operations. First, they were used to make better estimates of task duration for data acquisition tasks. Accurate estimates of the time to acquire the five types of PanCam panoramic images and the three types of Lidar scans were not available prior to the field test. We averaged sample collection times during operations early in the test to determine more accurate estimates. These estimates were used by the science team when building plans. They also were used by the Flight Director when monitoring robot progress.



Lidar FD

Panorama	Scans	Status	Total Runtime	Total Scans
00:02:42.8	8	ACTIVE	00:12:01.9	10

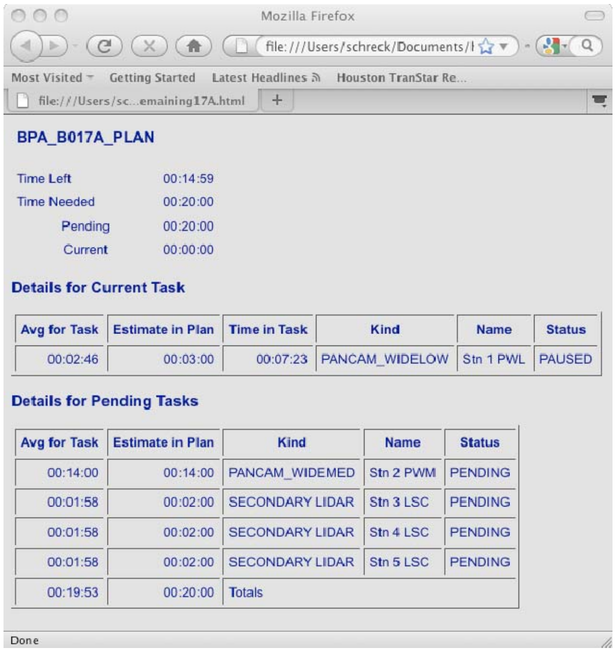
Figure 4. Example of Lidar Task Timer Display

Second, the elapsed data acquisition time from an instrument timer was compared to the expected acquisition time for the sample type to determine the time remaining in an acquisition task. The medium and high resolution PanCam samples took between 10 and 17 minutes to perform and the Flight Director frequently used these timers to determine time to task completion,

Third, the instrument timers were used to detect data acquisition problems. These problems were evidenced by a larger than expected elapsed time with no indication that a sample had been taken (i.e., the sample count did not increase). The Lidar instrument had difficulty in completing a panorama on multiple occasions due to the extreme thermal conditions and the Lidar timer was useful in detecting these problems quickly.

Finally, the Robot Operations Coordinator used the total Lidar runtime to estimate when to swap out the Lidar battery, since direct sensing of battery level was not available. During the field test we added the ability for the user to reset this timer after a battery swap.

Plan timers were used by the Flight Director to identify the currently active task and when it was marked complete or aborted, and to determine what type of data was being acquired. The plan timers were helpful in determining when a plan had gone beyond the allocated time, but did not provide the Flight Director with sufficient timing information to detect when plan execution was getting behind early enough to take action before the allocated time was expended. When plan execution did get behind, the plan timer also did not aid the Flight Director in determining whether some portion of the remaining tasks might be completed in the time left. We developed new metrics during the field test to address these needs. Specifically we computed the time needed to complete the ongoing task and all pending tasks using estimates of task time previously measured. Using this information, the Flight Director could assess if adequate time remained and, if not, could inspect estimated task times to aid in selecting which remaining tasks to perform (a form of contingency re-planning). For the field test, this information was provided as a snapshot display from a checkpoint of metrics and thus did not update as plan execution continued. For future tests, we believe a version of this display that updates with progress on the plan would be preferable. Figure 5 shows an example of the display used during the field test for these additional plan timing metrics.



BPA_B017A_PLAN

Time Left	00:14:59
Time Needed	00:20:00
Pending	00:20:00
Current	00:00:00

Details for Current Task

Avg for Task	Estimate in Plan	Time in Task	Kind	Name	Status
00:02:46	00:03:00	00:07:23	PANCAM_WIDELOW	Stn 1 PWL	PAUSED

Details for Pending Tasks

Avg for Task	Estimate in Plan	Kind	Name	Status
00:14:00	00:14:00	PANCAM_WIDEMED	Stn 2 PWM	PENDING
00:01:58	00:02:00	SECONDARY LIDAR	Stn 3 LSC	PENDING
00:01:58	00:02:00	SECONDARY LIDAR	Stn 4 LSC	PENDING
00:01:58	00:02:00	SECONDARY LIDAR	Stn 5 LSC	PENDING
00:19:53	00:20:00	Totals		

Figure 5. Display of Time Remaining in Plan 17A on June 20

High environmental temperatures increased the risk of robot subsystems overheating during operations. The metrics for battery management were used by the Robot Operations Coordinator primarily to monitor battery temperature for potential overheating. To support this task, we computed the maximum temperature observed for each group of eight batteries and compared it to the current temperature. The Flight Director felt the thermal information currently provided should be supplemented with summary metrics of thermal performance,

Communication quality varied significantly over the course of the field test. The average percentage of time spent in LOS for the entire field test was 13.4%, and varied from 1% to 38% of the day in LOS (Figure 6). Our metric for communication quality defined LOS as a dropout of all robot data. It was not uncommon, however, to experience dropped data messages without losing all robot data. As a result, the LOS metric is a conservative estimate of the quality of communication that can underestimate the impact of communication anomalies on data availability and quality of metrics. The Flight Director supplemented the LOS metrics with the message dropout warnings available for the individual robot subsystems to detect degraded communication due to dropped messages.

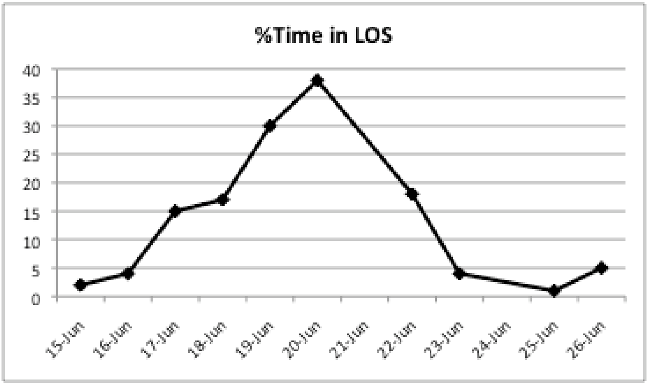
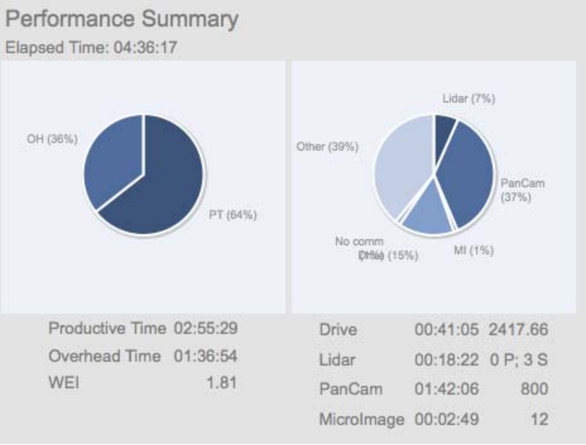


Figure 6. Daily Percentage of Time in LOS for Field Test

We produced a summary report at the end of each day using the metrics computed in real-time. These reports are intended to clarify how the robot spent its time, what data were collected, and what problems were encountered. These reports also should aid in identifying trends in expected robot performance over the field test. The summary measures in the debrief report identify how well the robot performed its tasks (see Figure 7 for an example of robot productivity on June 25) and what problems occur that affect that performance (see Figure 8 for an example of robot anomalies on June 25). An event log provides the details about what circumstances contributed to these summary measures. For example, one metric in the anomaly summary is the mean time spent by personnel intervening unexpectedly in robot operations. The log can be inspected to see exactly when samples were taken or when interventions were made. By inspecting adjacent events, the user may gain additionally insight into the operation.

These reports were used as supplements to the console notes taken by the flight controllers. Both the Flight Director and Robot Operations Coordinator felt these reports would be more useful if flight operations personnel could edit them. The types of editing mentioned includes removing some data points from metric computations and adding comments after operations when reviewing and analyzing performance. At the end of the test these reports also represent a mission performance summary of robot productivity and robot reliability.

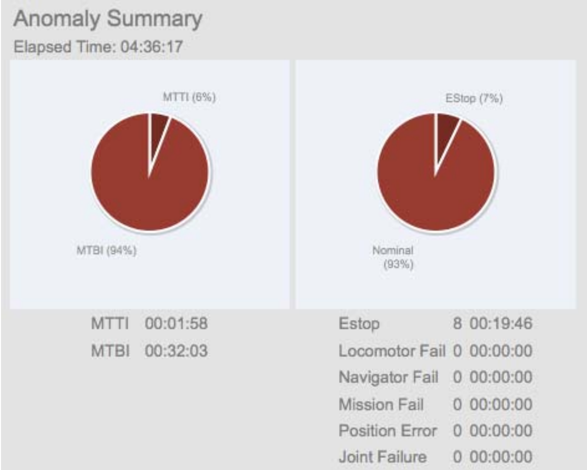

Performance Summary

Elapsed Time: 04:36:17

Productive Time	02:55:29
Overhead Time	01:36:54
WEI	1.81

Drive	00:41:05	2417.66
Lidar	00:18:22	0 P; 3 S
PanCam	01:42:06	800
MicroImage	00:02:49	12

Figure 7. K10 Robot Productivity on June 25


Anomaly Summary

Elapsed Time: 04:36:17

MTTI	00:01:58
MTBI	00:32:03

Estop	8	00:19:46
Locomotor Fail	0	00:00:00
Navigator Fail	0	00:00:00
Mission Fail	0	00:00:00
Position Error	0	00:00:00
Joint Failure	0	00:00:00

Figure 8. K10 Robot Anomalies on June 25

The average K10 robot productivity for the entire field test at Black Point was 37% productive time, 63% overhead time, and an average WEI of 0.73. Figure 9 shows these productivity metrics computed in real-time for each day of the field test.

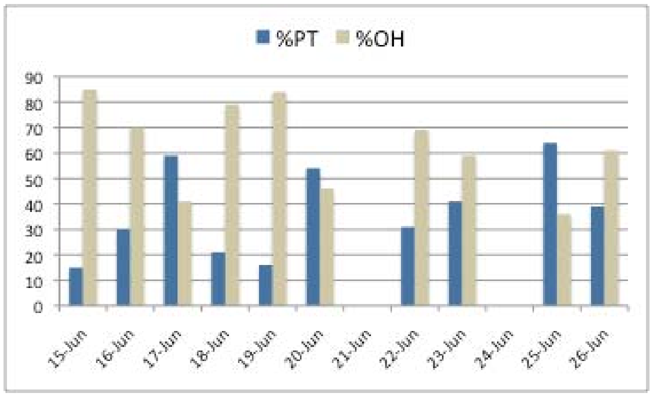


Figure 9. Daily K10 Robot Productivity Metrics for Field Test

We further decomposed robot time into the percentage of time spent on each type of task. Figure 10 shows this breakout from June 17-26. We eliminated data on June 15-16 because of an error in the drive time computation used in real-time for those days.

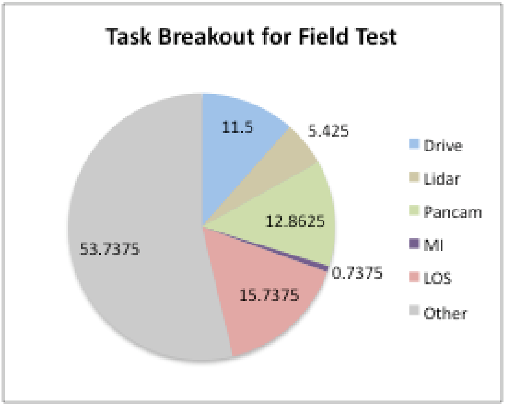


Figure 10. K10 Robot Task Breakout for the Field Test

The average MTTI computed in real-time for the entire field test at Black Point was 5.6 minutes, ranging from a minimum of 1.6

minutes to a maximum of 17.9 minutes. The average MTBI was 24 minutes, ranging from a minimum of 5.5 minutes to a maximum of an hour. Figure 11 shows these reliability metrics computed in real-time for each day of the field test.

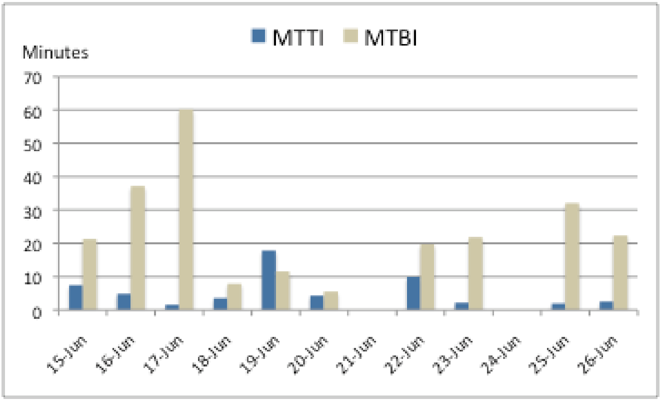


Figure 11. Daily K10 Robot Reliability for the Field Test

5. CONCLUSIONS

We have described an approach for in-line computation of robot performance metrics to aid human-robot interaction during remote operation of robots in space. Results of evaluating our approach during recent field tests with the K10 robot indicate real-time computation of robot performance can aid both robot operations and debrief after operations.

Plan and task timing metrics were used frequently during operations. These metrics were most meaningful when used in the context of expectations. Instrument timers were combined with knowledge of expected data acquisition times to monitor progress on the task and to detect when these task were not collecting data as expected. Similarly plan timers were combined with estimates of plan duration to monitor progress on the plan and to detect when the robot was getting behind. New metrics were identified and computed during the field test that compare the time left to complete the plan with estimates of the time needed to complete the remaining tasks. This information was used to determine whether adequate time remained to complete a plan and, if not, which of the remaining tasks should be performed. In all these cases, the expected task timing also was computed by the performance software. Comparing actual timing to expected timing is useful when inspecting performance after the mission as well. During the field test we computed the ratio of actual time to complete a plan with the estimated time to complete a plan. Using this metric we determined that 11 of 20 plans taken to completion were performed within the allocated time.

Operational use of WEI and Percentage of Time on Task to measure robot productivity indicates that the Percentage of Time on Task is more meaningful in real-time. WEI is difficult to interpret for real-time use. When overhead time is very small, WEI can be very large (or can be undefined if overhead is zero). The meaning of such large numbers is not clear. By normalizing productive time to total time in operations (i.e., shift time), the Percentage of Time on Task is guaranteed to vary between 0 and 100, ensuring greater consistency across operations and shifts.

Communication quality varied significantly over the course of the field test, ranging from 1- 38% of daily operations spent out of communication. Additionally we observed frequent dropped messages that affected data availability and the resulting quality of metrics. As a result of this loss of data, we observed small inaccuracies in statistics on task durations and data sample counts due to communication anomalies. Additional metrics are needed that characterize how dropped messages affect data availability. The prevalence of dropped messages also indicates that metrics computed using this lossy data are subject to error due to missed data and could benefit from algorithms that consider the quality of the data messages used to compute them.

6. FUTURE WORK

Early detection of plan threats give ground control more flexibility in re-planning because more time and resource remain than if the threat is detected late. Plan threats include getting behind when performing a plan, using more resource (such as battery power) than planned, or losing robot capability that affects plan completion. We plan to investigate metrics for early detection of plan threats. For example, can we detect a robot getting behind as a divergence between the time remaining in the plan and the time needed to complete the plan?

We also plan to investigate approaches for detecting when dropped messages impact quality of metrics. This includes performance algorithms that consider the quality of the data messages used to compute them.

We believe the performance data we have computed over multiple NASA field tests can be useful for future field tests. It can be used to characterize typical robot performance for terrain types or specialized operations and thereby improve our interpretation of future robot performance (i.e., was this typical robot performance for this type of terrain?). It also can be used to establish realistic expectations when designing the activities for future field tests.

7. ACKNOWLEDGMENTS

This research was funded by the NASA Small Business Innovative Research program, Human-Robotic Systems project and ESMD Analogs program. We would like to thank Rob Landis, Timothy Kennedy, and Frank Jurgens for their feedback about flight control metrics. Thanks also to David Lees and Mark Allen for their assistance in integrating the performance monitoring software with the K10 ground data system, to Estrellina Pacis for her assistance in defining metrics for operations assessment, and to Matt Deans for information about the robotic recon planning process.

8. REFERENCES

[1] Fong, T., M. Bualat, M. Deans, M. Allan, X. Bouyssounouse, M. Broxton, L. Edwards, R, Elphic, L. Fluckiger, J. Frank, L. Keely, L. Kobayashi, P. Lee, S. Y. Lee, D. Lees, E. Pacis, E. Park, L. Pedersen, D, Schreckenghost, T. Smith, V. To, and H. Utz. *Field Testing of Utility Robots for Lunar Surface Operations*. *AIAA Space 2008*. San Diego, CA.

[2] Jacob Crandall, Michael Goodrich, Dan Olsen, Jr., and Curtis W. Nielsen, “Validating Human–Robot Interaction Schemes in Multitasking Environments”, *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, Vol. 35, No. 4, July 2005

[3] Jacob Crandall and M. L. Cummings. Developing Performance Metrics for the Supervisory Control of Multiple Robots. *Human-Robot Interaction 2007*. March 2007.

[4] J. Arnold, “Towards a Framework for Architecting Heterogeneous Teams of Humans and Robots for Space Exploration”, M.S. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology. 2006.

[5] Mike Gernhardt, Work Efficiency Indices. Presentation at Johnson Space Center. November 15, 2005

[6] Edward Tunstel, “Performance Metrics for Operational Mars Exploration Rover,” *Journal of Field Robotics*, Volume 24 Issue 8-9, 651 – 670, September 2007.

[7] Tolga Kurtoglu, Ole J. Mengshoel, and Scott Poll, A Framework for Systematic Benchmarking of Monitoring and Diagnostic Systems,” *2008 International Conference on Prognostics and Health Management Proceedings*, October 6-9, 2008.

[8] Abhinav Saxena, Jose Celaya, Edward Balaban, Kai Goebel, Bhaskar Saha, Sankalita Saha, and Mark Schwabacher, “Metrics for Evaluating Performance of Prognostic Techniques,” *2008 International Conference on Prognostics and Health Management Proceedings*, October 6-9, 2008.

[9] Debra Schreckenghost, Terrence Fong, Tod Milam, Estrellina Pacis, and Hans Utz. *Real-time Assessment of Robot Performance During Remote Exploration Operations*. IEEE Aerospace Conference. Big Sky, MT. March 2009.

A Biologically Inspired Sensory Driven Method for Tracking Wind-Borne Odors

Brian K. Taylor
Case Western Reserve University
10900 Euclid Ave
Glennan 418
(216) 368 - 6044
btaylor@case.edu

Brandon L. Rutter
Case Western Reserve University
10900 Euclid Ave
Glennan 418
(216) 368 - 5216
rutter@case.edu

Roger D. Quinn
Case Western Reserve University
10900 Euclid Ave
Glennan 418
(216) 368 - 3222
rdq@case.edu

ABSTRACT

The ability of a vehicle to track a wind-borne odor to its source has numerous applications such as search and rescue missions and security operations. The Sensory Coupled Action Switching Modules (SCASM) control concept, based initially on known neural pathways in stick insects, provides a way to control a system using only sensory feedback. To date, SCASM has only been implemented in legged locomotion systems. However, results from implementing SCASM on legs of differing morphology suggest that it can be applied to control paradigms beyond legged locomotion. Here, we apply SCASM to the problem of odor tracking. We first develop a degenerate-case SCASM controller for tracking a 2D continuous odor plume to its source. In simulation, we then test this controller in a variety of probabilistic plume conditions. Basic performance metrics are developed to determine how well the controller can "Succeed" at finding the source, and "Linger" near the source. Results indicate that high success rates (≥ ~80%) coupled with high lingering ratios (≥ ~15) lead to good source localization. Overall, the study begins to demonstrate the feasibility of applying SCASM to control paradigms beyond legged locomotion.

Keywords

Biologically Inspired Robotics, Biologically Inspired Control, Odor Tracking, Chemical Plume Tracking, Pattern Generation, Sensory Coupled Action Switching Modules, Performance Metrics, Simulation

1. INTRODUCTION

1.1 The SCASM Control Concept

Sensory Coupled Action Switching Modules (SCASM) is a control concept that coordinates a system to generate an emergent behavior (e.g., forward stepping) using sensory feedback. As an example, consider a legged engineering system. A SCASM controller would coordinate the joints of the leg using sensory feedback such as joint angles and leg load to achieve an emergent behavior (e.g. forward or side stepping). SCASM was initially based on known local neural circuits and sensory pathways in stick insects. Ekeberg et al. [1] implemented a dynamic simulation of the stick insect middle leg that only used the known local neural circuits and sensory feedback pathways to generate stepping motion. Their goal was to determine whether or not these influences (e.g., increasing leg load causing the thoraco-coxa joint to go from protraction to retraction) were sufficient to explain the stepping motion of the leg. Their simulation successfully reproduced the stepping motion of the stick insect middle leg and provided possible predictions about how the joints of the front and hind legs might be coordinated to generate stepping. Rutter et al. [2,3] followed up on this work by developing a robotic stick insect leg that uses the same neural circuits and sensory feedback information that is available to the animal. Lewinger et al. [4,5] implemented SCASM on a biologically inspired hexapod, where the SCASM controller coordinates the joints within a particular leg, and a behavioral controller coordinates the interleg motion. Also, Rutter successfully reorganized the stick insect robot controller developed in [2] and [3] to generate stepping motion in a robotic cockroach leg [6]. It should be noted that cockroach and stick insect legs are anatomically quite different from one another. An important feature of SCASM applied to these robotic models is that it has an elegant implementation and requires less computational power than traditional engineered control systems. At the same time, they are able to achieve substantial behavioral robustness to disturbances such as changes in substrate height or transient limb obstruction without the need for specific scenario-handling software often used in traditionally engineered legged systems.

A high level way to represent a SCASM controller is an *event space diagram* (first used in [6] to describe the cockroach and stick insect leg controllers). A generic event space diagram is shown in Figure 1.

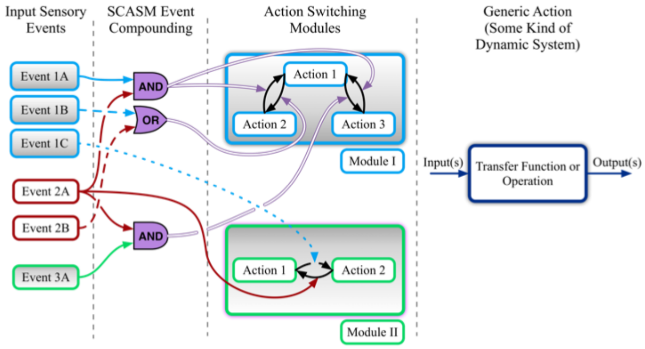


Figure 1. Generic Event Space Diagram.

In Figure 1, we start with Input Sensory Events. These are events in the controlled system or its environment that can be detected via sensory feedback signals (e.g., leg load, joint angle, odor concentration). Some of these events may then be compounded together in some fashion (e.g., logic gates or weighted sums) to form higher level and/or more abstracted sensory feedback events. This is biologically plausible since neurons are capable of forming various kinds of logic gates and applying weights to input signals. Not all Input Sensory Events necessarily undergo compounding. After compounding, we have Action Switching Modules. Each module controls some piece of the system (e.g., in a legged system, a module would control a joint such as a knee). Within each module, there are actions (e.g., in a joint, possible actions might be flexion and extension). The individual and/or compounded sensory feedback signals elicit transitions between different actions within each module. For a given module, all actions are not necessarily reachable from one given action (e.g., in Module I, Action 3 is only reachable from Action 1, not Action 2). To provide an example information flow, suppose that we have Sensory Events 1A and 2A. These signals would be "AND"ed together, and their combination would elicit transitions from Actions 2 or 3 to Action 1 in Module I. This transition could trigger Sensory Event 1C, which would cause a transition from Action 1 to Action 2 in Module II. This could trigger Sensory Event 3A, which could cause a transition from Action 1 back to Action 3 in Module I. In this way, we can see that we have Action Switching Modules that are Coupled through Sensory feedback. The color-coding between some of the modules and sensory inputs is to indicate that the actions defined within a particular module can elicit direct sensory feedback. For example, in a legged system, switching from flexion to extension in a joint would cause a change in the angle being sensed at that joint. In contrast, we can also use sensory feedback from the environment which is not inherent to the system itself. For example, in an odor tracking vehicle, odor presence will change the vehicle's behavior. However, odor is a property of the environment, not the vehicle. Each action can be thought of as defining a dynamic system. For the legged robotic systems in [2,3,5], the dynamic system is affected by changing muscle model parameters. The muscle models at a joint determine motor commands, and the physical properties of the robot complete the system.

We note here that a SCASM controller in some circumstances can be represented as a finite state machine. However, it is possible to construct a SCASM controller where action switching is not discrete, and a finite state machine representation cannot be created. Also, even if a SCASM controller is represented as a finite state machine, the connections in this kind of representation are not always useful from the standpoint of determining or modifying the underlying mechanisms that generate the system's behavior.

1.2 Applying SCASM to Odor Tracking

With an understanding of SCASM control for the joints of a leg, our goal is to apply it as a generalized concept to a different problem: tracking an odor upwind to its source. The ability to track fluid-borne chemicals can allow vehicles to track and locate things such as: explosives, chemical/biological threats, people and animals in search and rescue scenarios, and illegal substances [7-10]. [7] and [8] attempted to mimic the behavior of the tobacco hornworm moth, while [9] and [10] use a combination of biological and engineering approaches to track chemical plumes in air and water, respectively. The metrics used in these works focus on recording whether or not the agent successfully came within some acceptable distance of the odor source [8-10], and recording the raw time it took the agent to reach the source [8,9].

A SCASM based odor tracker would present a new method for chemical plume tracking, and may aid researchers in better understanding the odor tracking capabilities of biological systems. Using SCASM-controlled simulated and robotic models is already beginning to increase understanding of the legged locomotion capabilities of biological systems [1,6]. More importantly, beyond developing a new odor tracking strategy and aiding biological work, a SCASM based odor tracker would demonstrate that SCASM, although originally based in legged insect locomotion, is a more broadly applicable concept. Until now, it has only been implemented in legged systems. However, based on the success of implementing SCASM on legged systems of differing morphology and a multilegged system, it appears that the SCASM concept can be applied to a variety of other problems. We envision SCASM being applied to control paradigms where it is desired to have the system go though a coordinated and potentially repeating set of actions in concert with the environment. Furthermore, recall that SCASM appears to require less computational power than traditional engineering solutions for legged locomotion. This suggests that SCASM may have the potential to be a more computationally efficient method for controlling current and future systems.

In this paper, we develop a SCASM controller to operate in a simplified 2D environment. In a MATLAB simulation, we then test this controller on a kinematic vehicle that uses standard vehicle coordinates under a variety of: plume conditions (e.g., environmental odor content), vehicle assumptions (e.g., is the vehicle's velocity affected by the wind velocity), and controller parameters (e.g., constant vs. variable vehicle speed). Basic performance metrics that gauge whether the vehicle can find the odor source and how well the vehicle lingers near the odor source are developed to gain an understanding of how well the controller is able to localize the odor source under different environmental and controller conditions. Although they are simplified, the environments in this study give us a basic idea of what kinds of conditions are conducive to good performance. Overall, the study begins to demonstrate the feasibility of using SCASM control in non-legged systems. We note that the focus of this study is to gain insight into how varying the controller and environmental parameters affects the vehicle's performance. We do not attempt to optimize the controller's parameters.

2. Methods

2.1 Using a Simplified 2D Environment

In general, real-world odor tracking is a 3D problem [7-10]. Also, a fluid-borne odor is usually discontinuous due to a turbulent fluid velocity field [7-13]. However, based upon the goals of this work, we use a simplified 2D odor environment, which allows for easier evaluation of the controller's behavior and performance (more is said about our long term goal of 3D work in Section 2.2 and Section 4). The environment is infinite in the X and Y directions, contains a "triangular" plume, and employs uniform steady (i.e., constant in time) freestream wind velocity. While this is not a fluid mechanics based environment, it gives us a well understood model that eases our ability to test the controller's basic behavior.

2.2 Developing an Event Space Diagram

With an environment in place, we must now develop an event space diagram to represent an odor tracking SCASM controller. For odor tracking, the sensory feedback signals that the controller receives are egomotion (i.e., vehicle motion relative to the environment) and odor presence (i.e., does the vehicle sense odor at a particular point in space and time). Figure 2 illustrates the initial event space diagram for our system.

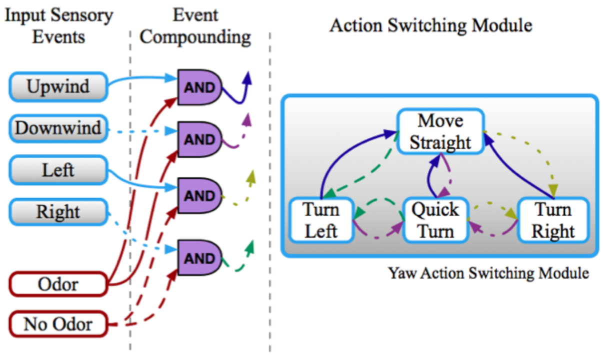


Figure 2. Current event space diagram for the 2D controller.

As shown in Figure 2, we currently have one action switching module which controls yaw motion. This module contains 4 actions. Because we only have one action switching module, there is not any sensory feedback coupling between modules, so this is a degenerate case of a SCASM control network. Since our study is 2D, egomotion is separated into the following cases: moving up/downwind, and moving left or right relative to the wind. The information flow in the controller is as follows. If the vehicle is moving upwind AND senses odor, then the controller transitions to the *Move Straight* action, which moves the vehicle in a straight line along its own x-axis. If the vehicle is moving downwind AND senses odor, then the vehicle is in the plume but facing in the wrong direction (we need to move upwind to find the odor source). In this case, the controller transitions to the *Quickturn* action, which is a high rate clockwise turn that reorients the vehicle such that it faces upwind. If the vehicle is moving to the right relative to the wind AND does not sense odor, we might envision that the vehicle is on the right side of the plume and moving away from it. In this case, the controller transitions to the *Turn Left* action, which is a moderate rate counterclockwise turn. Likewise, if the vehicle is moving to the left relative to the wind AND does not sense odor, we might envision that it is on the left side of the plume and moving away from it. In this case, the controller transitions to the *Turn Right* action, which is a moderate rate clockwise turn. These actions seek to keep the vehicle in the plume and facing upwind. In all of the turning states, the vehicle still has a forward velocity. Speed and turn rates in this controller are constant (i.e., in any particular action, the vehicle moves in a straight line or a circle). We note that this controller is not optimized and that its performance is dependent on the vehicle's starting location and pose within the world. However, despite its imperfections, it gives us a starting point for developing a SCASM based odor tracker.

We also note that since we are in 2D, our event space diagram only has one action switching module for controlling yaw motion. A 3D controller would have pitch and/or roll modules in addition to yaw, and sensory events regarding the vehicle's full orientation. This would create a large amount of sensory feedback coupling (e.g., roll feedback influencing the yaw actions). In addition, compared to previous SCASM controllers, the action switching module in Figure 2 has a large number of actions (action switching modules to date have had only 2 actions). This is due to the fact that actions in the yaw module are very simple (i.e., constant vehicle velocity and turn rates), so more actions are necessary to generate an emergent behavior. More complex and robust actions containing internal feedback would allow us to cut down on the total number of necessary actions. However, the simple actions in Figure 2 provide a starting point from an implementation standpoint.

2.3 Plume Conditions

The controller was tested in three kinds of "triangular" plumes: *Deterministic*, *Uniform Probability* and *Distance Dependent Probability*. In the *Deterministic* environment, if the vehicle is within the plume (i.e., within the triangular region), then it successfully detects the presence of odor. This is the plume condition that the event space diagram in Figure 2 is designed for. In the *Uniform Probability* environment, if the vehicle is within the plume, then it successfully detects the presence of odor with some preset fixed probability (e.g., if the vehicle is in the plume, then it successfully detects odor 75% of the time). This paradigm is used because a wind-borne odor is sensed in a discontinuous manner. With this condition, the vehicle will not detect odor 100% of the time even if it stays in the plume. For the *Distance Dependent Probability* environment, if the vehicle is in the plume, then it successfully detects odor with a probability that decays as distance from the odor source increases. This is done because odor is more difficult to detect farther away from its source. For this study, the decay of probability away from the source is modeled by a sigmoid, which is shown in Eq. 1

$$P = \frac{A}{1+e^{ar}} \qquad \text{Eq. 1}$$

Here, P is the probability of detecting an odor, a is what we refer to as the "*sigmoid parameter*", A is a constant and r is the Euclidian distance between the vehicle and the odor source. A is set to 2 so that when the vehicle is at the odor source (i.e., r = 0), the probability of sensing odor is 1. This equation is plotted for various values of a in Figure. 3.

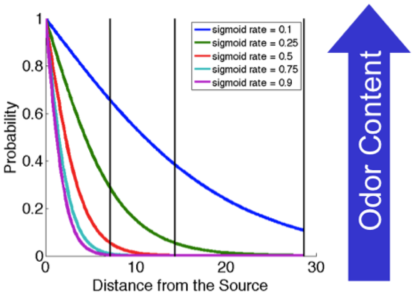


Figure 3. Plot of Eq. 1 for various values of sigmoid parameter *a*. Odor content increases as *a* decreases. The vertical lines indicate the starting distances that were used in this study.

Figure 3 shows that, for a fixed distance from the odor source, as a increases, the probability of detecting an odor drops. We therefore say that increasing a decreases the environment's odor content. In this study, the following values of a were used (ordered from highest to lowest odor content): a {0.10, 0.25, 0.50, 0.75, 0.90}. A sigmoid was used because it can be constructed such that it is unity at the odor source and decays monotonically to zero.

2.4 Initial Simplifications and Trials

For the initial part of the study, the simulation used the following simplifications and assumptions:

1) The vehicle is a kinematic entity with no mass or inertia (i.e., a coordinate system whose motion and orientation is completely determined by speed and rotation rate commands). This level of abstraction can be used to understand ground or flight vehicle behavior.
2) The vehicle can perfectly measure the true wind velocity and its ground velocity.
3) The wind velocity cannot affect the vehicle's velocity.
4) The vehicle's commanded speed is constant.
5) The wind velocity is constant.

In Section 3.1, we show one result from these initial assumptions to illustrate how the controller manifests its behavior. Specifically, we examine the controller's performance in a *Uniform Probability* environment and relate it to the event space diagram shown in Figure. 3. In Sections 2.5 and 2.6, we remove some of the above simplifications to gain insight into the vehicle's performance under different operating conditions.

2.5 Constant Speeds Influenced by Wind

In this section, we remove simplification 3 described in Section 2.4 by adding the wind velocity to the vehicle's velocity at each time step so that the vehicle can be "pushed" by the wind. For this paradigm, we test the vehicle at 3 starting distances from the source. The baseline starting distance was determined by dividing the maximum vehicle velocity to be tested by the maximum turn rate to be tested (10 length units/second and 40°/second, respectively), giving a distance of 14.3230 length units/second. Turn rate testing is still under investigation, and is not presented in this paper. Scaling this distance by nondimensional factors ($R_{0\text{-Initial}}$) of 0.5, 1.0, and 2.0, the starting distances used were 7.1620, 14.3230 and 24.6478 length units. For each starting distance, the vehicle's speed is varied through 5 different values: {0.1, 0.5, 1.0, 5.0, 10.0} length units/second. For each vehicle speed, the controller is tested in each of the 5 *Distance Dependent Probability* environments. Figure 3 illustrates the starting distances that are used in relation to the initial probability of sensing odor. For each *Distance Dependent Probability* environment, data was analyzed for 8 R_{Critical} values. This gives us a 4-dimensional parameter space ($R_{0\text{-Initial}}$: vehicle velocity: sigmoid parameter: R_{Critical}). R_{Critical} is a performance metric reference value that is defined in Section 2.7.

2.6 Variable Speed Influenced by Wind

In this section, we remove simplifications 3 and 4 described in Section 2.4 by varying the vehicle's commanded speed throughout the course of a run and allowing the wind velocity to influence the vehicle's velocity (the wind velocity is still constant). This was done based on the data collected from the protocol described in Section 2.5. The idea is that the vehicle speeds up when it is far from the odor source, and slows down when it is close to the odor source. This would make the vehicle "rush" to get close to the odor source, and then slow down to search a local area for the odor source. Because the vehicle does not explicitly have access to its position, distance from the odor source is approximated by counting the number of times an odor is sensed over a fixed time span. The farther away the vehicle is from the source, the less it should sense odor. With this in mind, the vehicle's velocity is varied in the following way

$$v = \left(v_{base}\right)\frac{M}{Number\ \ of\ \ times\ \ odor\ \ is\ \ sensed} \qquad \text{Eq. 2}$$

where v_{base} is a preset baseline vehicle velocity (set to 0.1 length units/second), and M is the number of time steps over which the vehicle counts whether or not it has sensed odor (set to 20 for this study). If the vehicle senses no odor over M timesteps, then the velocity is set to 1 length unit/second to prevent division by zero.

2.7 Performance Metrics

For the data collected in Sections 2.4 – 2.6, Success and Lingering metrics were employed to begin quantifying "good performance". The success metric asks "Did the vehicle pass within a particular distance (R_{Critical}) of the odor source?" This kind of metric has been used in both biological studies and engineered odor-tracking systems to determine whether the agent (animal or vehicle) has "reached" the odor [8-11]. We define a success percentage as

$$Success\% = \frac{Number\ \ of\ \ successful\ \ trials}{N} \qquad \text{Eq. 3}$$

where N is the number of trials. The lingering metric seeks to determine how well the vehicle can localize the odor source. We use a lingering metric because it gives a description of the vehicle's near source behavior.

To compute the lingering metric, we first define a lingering distance as

$$R_{Linger} = qR_{Critical} \qquad \text{Eq. 4}$$

where R_{Linger} is the distance from the source that defines lingering and q is a multiplying factor. For our study, q was arbitrarily set to 3, indicating that the lingering vicinity was within three times the distance at which the vehicle was considered to have "reached" the point source. With this in mind, we can compute a lingering ratio in the following way:

$$Linger\ \ Ratio = \frac{\max(Actual\ \ Time\ \ in\ \ Linger\ \ Diameter)}{Straight\ \ Line\ \ Time\ \ in\ \ Linger\ \ Diameter} \text{Eq. 5}$$

This ratio compares the maximum time spent near the odor source during a trial to the time it would take the vehicle to cross the lingering diameter in a straight line. Higher values imply that the vehicle spends a great amount of time near the source and lower values imply that the vehicle only passes by the source for short periods of time. For N trials, we define a lingering percentage as

$$Linger\% = \frac{Num\_Trials\ \ above\ \ a\ \ specified\ \ linger\ \ ratio}{N} \text{ Eq. 6}$$

For source localization, we would expect relatively high success percentages coupled with relatively high lingering percentages (i.e., the vehicle finds the odor and stays near it). For this study, the specified lingering ratio was set to 15. This choice was based on preliminary observations of the controller's performance. The success and lingering metrics are evaluated for multiple values of R_{Critical} to understand how the quantitative performance of the vehicle changes when the reference value of the performance metrics is changed. R_{Critical} is varied through the following values: R_{Critical} = {0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00} length units. With these metrics, it is possible for the vehicle to "Succeed" but not remain near the source for a long period of time. It is also possible for the vehicle to "Linger" near the source without technically "Succeeding". These metrics gauge how well the vehicle is able to find the source and remain near it.

We note that in practice, the specific values of $R_{Critical}$ and q would depend on the desired performance of the vehicle for a particular scenario (e.g., high-resolution localization may demand values of $R_{Critical}$ and q that are smaller than what is presented in this study). The metric values used here were chosen to gain insight into the controller's performance over a range of baseline values.

2.8 General Trial Information

For each parameter combination ($R_{0\text{-Initial}}$: vehicle velocity: sigmoid parameter: $R_{Critical}$), 20 trials were performed. At each time step, the vehicle sensed odor with a probability P as defined in Eq. 1. Also, for each trial, the initial starting location and orientation were randomized subject to the constraint of the specified starting distance. The vehicle was only allowed to start within the odor plume since it did not have a plume finding behavior such as casting (moving back and forth primarily across the wind with occasional downwind drifting) [7,11,12]. The same 20 random number generator seeds were used for the trials at each parameter combination. The turn rates used were as follows: Turn Right = 20°/second; Turn Left = -20°/second; Quickturn = 40°/second (turn rates are given relative to the vehicle's coordinate system).

3. Results

3.1 Vehicle Behavior and its Relationship to the Event Space Diagram

In Figure 4 we analyze an odor track from the *Uniform Probability* environment of P = 90% in reference to the event space diagram of Figure 2.

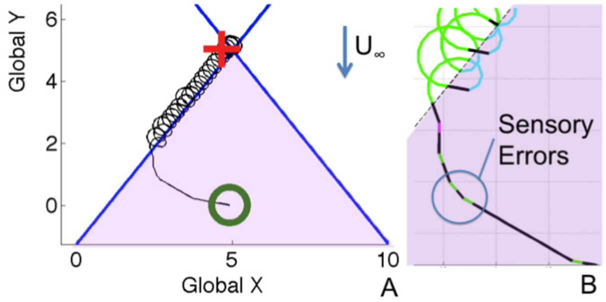


Figure 4. A) Sample odor track from the P = 90% *Uniform Probability* environment. The vehicle starts at the open circle and ends at the cross. B) Zoomed in portion of the odor track. Black lines represent the *Move Straight* action, green lines represent the *Turn Right* action, magenta lines represent the *Turn Left* action and cyan lines represent the *Quickturn* action. In this and all remaining figures, U_∞ is the wind velocity.

In the first leg of the odor track (i.e., the long black line in Figure 4B), the vehicle senses odor AND has a component of upwind velocity, so the controller transitions to the *Move Straight* action. The vehicle remains in this action until it leaves the odor plume, at which point the vehicle senses a lack of odor AND that it is moving to the left relative to the wind. This causes a transition to the *Turn Right* action. The vehicle stays in this action until it re-enters the odor plume. However, now the vehicle senses that it is in the plume AND moving downwind, which triggers a transition to the *Quickturn* action. Once the vehicle has regained a component of upwind velocity in the plume, the controller transitions to the *Move Straight* action. This cycle of *Move Straight*, *Turn Right* and *Quickturn* repeats itself, which gradually moves the vehicle up the edge of the plume towards the source in a looping manner. Under these conditions, an unintended feature of the controller is that the vehicle is able to orbit the odor source. This capability was not explicitly built into the controller. It is a consequence of the sensory feedback connections in the event space diagram. We note that in Figure 4, the odor track does not form perfect lines or circles. This is due to the fact that the controller is in a probabilistic environment, and can therefore encounter sensory errors, resulting in intermittent transitions to other actions. For example, the main action of the first part of the odor track is *Move Straight*, but there are intermittent transitions to the *Turn Right* action (Figure 4B).

3.2 Constant Speed Influenced by Wind

Figure 5 shows the success and lingering percentages for the following conditions: $R_{0\text{-Initial}}$ = 0.5, Vehicle Speed = {0.5, 1.0, 10} length units/second

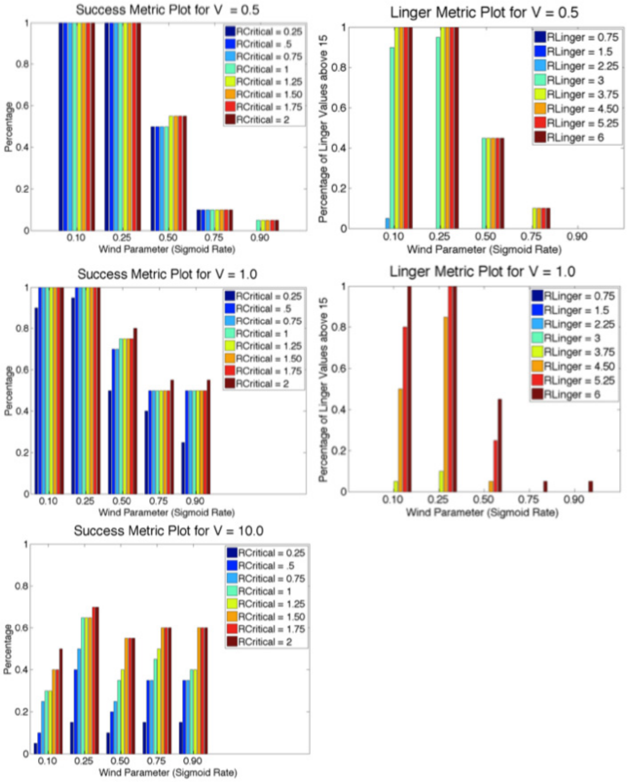


Figure 5. Success (Left) and lingering (Right) metrics for $R_{0\text{-Initial}}$ = 0.5. For each plot, for a given sigmoid parameter, each bar corresponds to a different $R_{Critical}$ or R_{Linger} value.

For the success metric, as speed increases, the success percentage increases across sigmoid parameters. Also, with an increase in speed, the overall magnitude of success increases from 0.5 to 1 length units/second, and then decreases for speeds greater than 1 length unit/second (the data for V = 0.1 length units/second had maximum success rates of 85% and 40% in the a = {0.10, 0.25} environments respectively, and maximum linger rates of 15% and 30% in the a = {0.10, 0.25} environments respectively). For the lingering percentage, the vehicle's ability to linger is higher at a vehicle speed of 0.5 length units/second. Also, as the vehicle speed increases, the lingering percentage increases in the higher R_{Linger} values and decreases in the lower R_{Linger} values. Upon examining the odor tracks and comparing them to the plots in Figure 5, it appears that higher success rates coupled with higher linger rates for lower values of R_{Linger} provide a useful description of source localization. The vehicle was still able to localize the source at vehicle speeds of 5 and 10 length units/second. However, the localization is not as good when the vehicle moves quickly. The lingering metric is not shown for V = {5, 10} length units/second because based on the lingering percentage definition given by Eq. 6, none of the trials for these vehicle speeds lingered. Overall, it appears that starting this close to the source, the vehicle has the greatest capability to "Succeed" and "Linger" in sigmoid parameter environments of 0.1 and 0.25. In cases where the vehicle failed to "Succeed" and/or "Linger", the vehicle "spiraled" away from the source. This was due to the fact that when the odor was sparse enough within the plume, the vehicle would enter the "Turn Right" or Turn Left" actions, causing it to move in a circle. Since the vehicle had a component of wind velocity added to its own velocity at each time step, the vehicle moved in a circle while being "blown" away from the source (Figure 6-A1). Depending on the trial environment, this kind of behavior could occur even when the vehicle localized the source (Figure 6-B1). Figure 6 provides sample odor tracks that illustrate success vs. failure, good vs. poor lingering capability, and overall behavior.

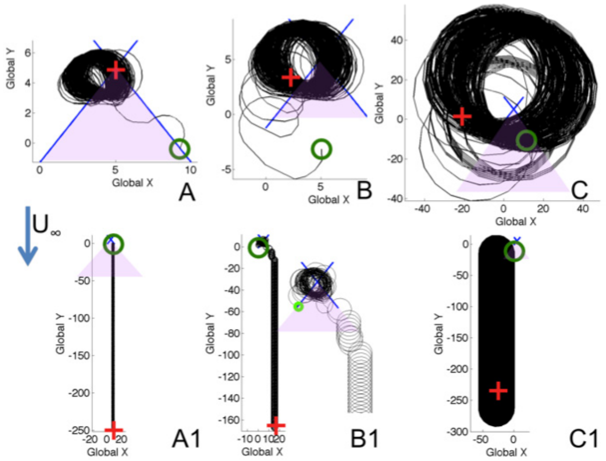


Figure 6. The top row (a = 0.1) illustrates trials where the vehicle "Succeeded" and "Lingered". The bottom row illustrates trials with a success and/or lingering failure (a = 0.5 for A1, a = 0.25 for B1 and C1). The vehicle velocities for the odor tracks that are shown are 0.5, 1, and 10 length units/second (left, middle and right columns, respectively). The vehicle starts at the open circles and ends at the crosses. Note that while C appears to linger, its lingering ratio is under 15.

Figure 7 shows the success and "Linger" metrics for a more distant starting position given by: $R_{0\text{-}Initial}$ = 2, Vehicle Speed = {0.5, 1.0, 10} length units/second

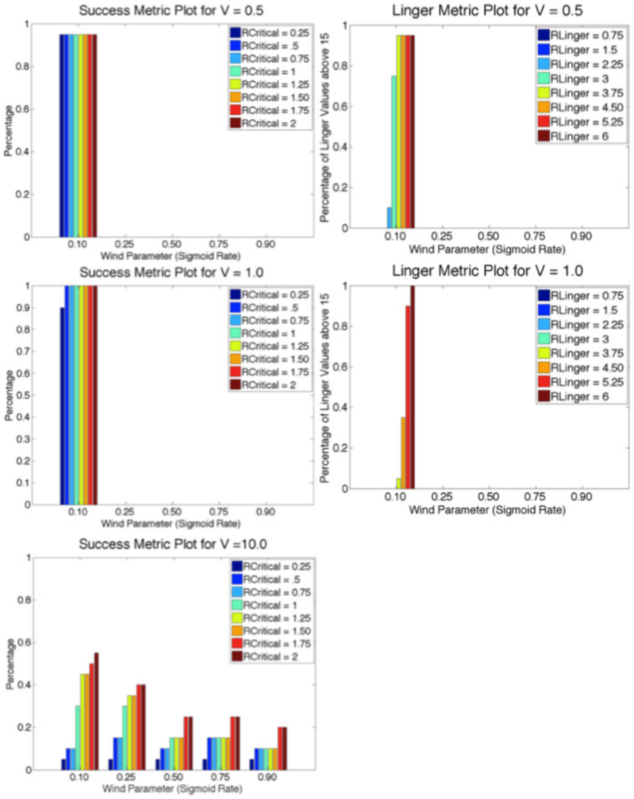


Figure 7. Success (Left) and Lingering (Right) metrics for $R_{0\text{-}Initial}$ = 2. For each plot, for a given sigmoid parameter, each bar corresponds to a different $R_{Critical}$ or R_{Linger} value.

Here, again, as vehicle speed increases, the success percentage across sigmoid parameters increases. Also, the success percentage increases from 0.5 to 1 length units/second, and then decreases for vehicle speeds greater than 1 length unit/second. From the lingering percentage, "tight" localization only occurs for a sigmoid parameter of a = 0.1. The vehicle still exhibits the behavior of tighter localization with smaller velocities (e.g., the lingering metric is higher for lower values of R_{Linger} at a vehicle speed of 0.5 length units/second vs. a vehicle speed of 1 length unit/second). Success only appears to occur at the higher vehicle velocity values (plots for a vehicle velocity of 0.1 length units/second are not shown because the vehicle did not succeed or linger at this velocity for this initial distance). Lingering plots for vehicle speeds of 5 and 10 length units/second are not shown because the vehicle failed to linger at either of these speeds based on the lingering percentage definition given by Eq 6. Figure 8 provides sample odor tracks. The failure mechanisms are the same as those described for the trials where $R_{0\text{-}Initial}$ = 0.5.

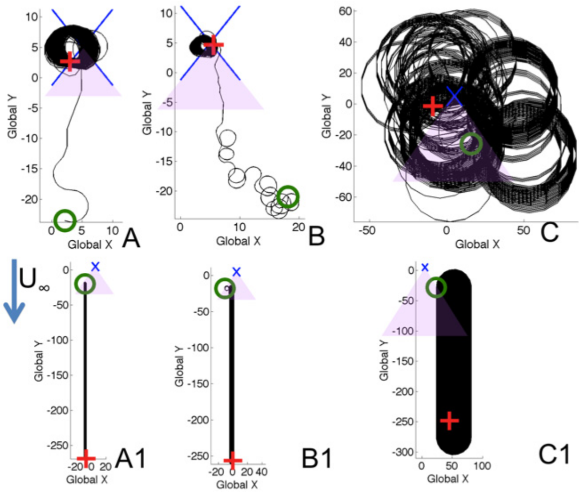


Figure 8. The top row (a = 0.1) shows trials where the vehicle "Succeeded" and "Lingered". The bottom row (a = 0.25) illustrates trials with a success and/or lingering failure. The vehicle velocities for the odor tracks that are shown are 0.5, 1, and 10 length units/second (left, middle and right columns, respectively). The vehicle starts at the open circles and ends at the crosses. Note that while C appears to linger, its lingering ratio is under 15.

3.3 Variable Speed Influenced by Wind

The results in Section 3.2 have two implications. First, when the vehicle is far from the source, moving faster increases the chance of success across sigmoid parameters. Second, it may be better to have the vehicle move slower when it is close to the source since this can increase lingering capability (i.e., high values of the lingering metric for lower values of R_{Linger}). These hypotheses make physical sense. If the vehicle were far away from the source, it would make sense to move quickly to get into the neighborhood of the source as fast as possible, and then slow down to search a smaller area for the source. The hypothesis that varying velocity can lead to performance improvements was tested by varying the vehicle's commanded velocity during a given run according to Eq. 2. Results for starting distances of $R_{0\text{-Initial}} = 0.5$ and $R_{0\text{-Initial}} = 2.0$ are shown in Figure 9.

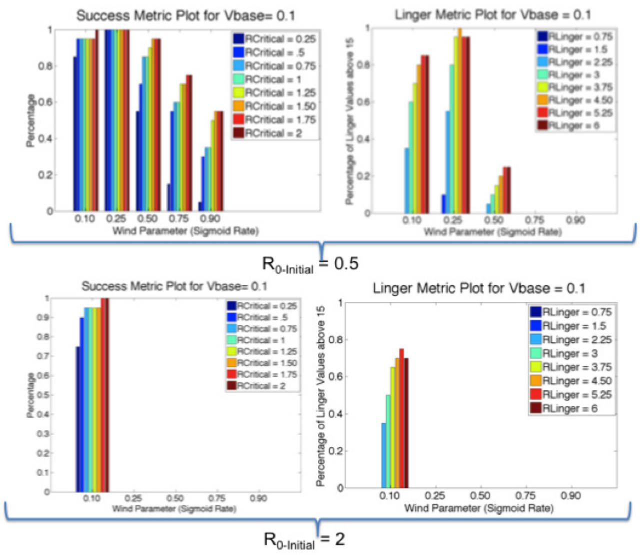


Figure 9. Illustration of the Success (Left) and Lingering (Right) metrics. For each plot, for a given sigmoid parameter, each bar corresponds to a different $R_{Critical}$ or R_{Linger} value. The top 2 plots are for $R_{0\text{-Initial}}$ starting values of 0.5. The bottom 2 plots are for $R_{0\text{-Initial}}$ starting values of 2.

From Figure 9, when the vehicle starts far away from the source, it only succeeds in a sigmoid parameter environment of 0.1. However, when it starts closer to the odor source, the vehicle is able to achieve success in all sigmoid parameters, with source localization occurring in sigmoid parameters of 0.1 and 0.25. This is consistent with the results from the wind-influenced constant speed trials in Section 3.2.

4. Conclusions and Discussion

We have developed a preliminary SCASM network for tracking odors in a simplified 2D environment. We tested the controller in an environment where the odor decays as the vehicle moves away from the source and where the vehicle's velocity can be influenced by the wind. We used the results from these tests to begin understanding how the environment affects the controller's performance by employing basic performance metrics. Overall, this study illustrates the development of a basic SCASM-based odor tracker and demonstrates the possibility of extending SCASM for use in control paradigms beyond legged locomotion.

It appears that simultaneously having relatively high success rates (in the range of approximately 80% to 100%) and relatively high lingering rates (in the range of approximately 75% to 100%) results in tighter localization of the odor source (metric values taken from Figures 5, 7 and 9). As stated in Section 2.7, the lingering ratio of 15 was based on our initial observations of the controller's performance. Based on the metrics developed here, the current controller's preferred odor content environments appear to be a = 0.1 and a = 0.25 when the vehicle is started close to the source (both for constant and variable vehicle speed). When the vehicle starts farther away from the source, its preferred environment seems to be a = 0.1. This trend suggests that as the starting distance increases, the robustness of the controller across sigmoid parameters decreases, which makes sense based on Figure 3. At the farthest starting distance, the only environment that has a non-negligible probability of odor detection is a = 0.10, whereas at the closest starting distance, sigmoid parameters of 0.10 and 0.25 have non-negligible probabilities of odor detection. At higher speeds, the vehicle is capable of succeeding in sigmoid parameter environments with odor contents less than those of a = 0.1 or a = 0.25. However, due to the vehicle's larger turning radius at these speeds, the controller does not seem to be able to linger as well, which can either decrease or destroy the vehicle's localization ability. Changing the vehicle's velocity from a constant to a variable value that depends on how often odor is sensed did affect the nature of the resulting odor tracks (i.e., for one action, an odor track can have different turn radii). Also, with a variable vehicle speed, for $R_{0\text{-Initial}} = 0.5$, the vehicle seemed to exhibit better lingering performance in the lower values of R_{Linger}, suggesting that for this initial condition, better localization may be possible with a variable vehicle speed. However, the success and linger metrics indicate that even with this change, the preferred odor content environments remained the same, so there was not an improvement in robustness across sigmoid parameters.

Qualitatively, when the controller was able to "Succeed" and "Linger", the odor tracks appeared to be indicative of the source location (e.g., Figures 6A and 8A show the vehicle orbiting the odor source with a relatively small turning radius as compared to Figure 6C). Even when the controller was not able to succeed, or linger that well, there were cases where it was still able to produce potentially useful behaviors, such as outlining one of the plume boundaries (data not shown). In fact, in some of the cases where the vehicle succeeded, the path that the vehicle took was along one of the plume's edges. This occurred a great deal for a vehicle speed of 0.1 length units/second and $R_{0\text{-Initial}} = 0.5$ (data not shown). The effect of the wind speed was to continuously push the vehicle away from the source. In successful runs, this could hinder the lingering capability of the vehicle (i.e., the vehicle would go through a cycle of finding the source and then being pushed away). In runs where the vehicle failed to locate the odor source (and even in some of the "successful" runs), the wind speed caused the vehicle to spiral away from the source because the vehicle entered a turning state due to a lack of odor and the wind velocity was continuously added to the vehicle's velocity (Figures 6-B1 and 8-B1). As the vehicle speed increased, the wind speed's influence became less and less pronounced, as one would expect.

We note that the linger ratios had a large amount of variability, including a few distant outliers. A preliminary analysis revealed that the linger metrics may not be normally distributed. We plan to further investigate and quantify this variability in future studies.

A number of steps can be taken to extend and improve upon this work. First and foremost, we would like to repeat this study, but looking at the effects of changing other controller and environmental parameters, such as the vehicle's turn rate, wind speed, and the type of wind field used (i.e., unsteady in time and/or variable in space). Also, we plan to continue developing new performance metrics and refining the current ones. For example, the kind of statistical analysis that is applied to the odor tracks of *Drosophila Melanogaster* in [9] would provide a quantitative analysis of the odor tracks in addition to the success and lingering metrics. To quantify the vehicle's ability to indicate the plume boundary and odor source, it may be possible to use a clustering algorithm on the action information that the vehicle records during the course of a run. With a given run's actions and positions, a clustering algorithm may provide estimations of the odor plume structure and source, which could then be compared to the real boundary and odor source. In addition, we would like to increase the controller's robustness by improving its ability to negotiate low odor content environments (i.e., $a \geq 0.25$). This could be accomplished by incorporating a plume finding behavior such as insect casting into the event space diagram. We would also like to increase the realism of the simulation by incorporating dynamics and aerodynamics into the vehicle, and by eventually developing a 3D controller. Our long-term goal is the implementation of a SCASM odor tracking controller in 2D and 3D hardware platforms (such as those described in [7] and [13]).

5. ACKNOWLEDGMENTS

We thank Nicole Doorly for her contributions in reviewing the data. This research was performed under an appointment to the U.S. Department of Homeland Security (DHS) Scholarship and Fellowship Program, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by Oak Ridge Associated Universities (ORAU) under DOE contract number DE-AC05-06OR23100. The opinions expressed here do not necessarily reflect the policies and views of DHS, DOE, or ORAU/ORISE.

6. REFERENCES

[1] Ekeberg, Ö., Blümel, M., Büschges, A. "Dynamic simulation of insect walking," Arthropod structure & development, vol. 33, pp. 287 (14 pages), 2004

[2] Rutter, B.L., Lewinger, W.A., Blümel, M., Büschges, A., Quinn, R.D. (2007) Simple Muscle Models Regularize Motion in a Robotic Leg with Neurally-Based Step Generation. Proceedings of ICRA 2007, Rome

[3] Rutter, B.L., Lewinger, W., Taylor, B.K, Wilson, M., Blümel, M., Ekeberg, Ö., Büschges, A., Ritzmann, R.E., Quinn, R.D. (2006) "Neurally-Based Robot Control for Neuromechanical Modeling of Insect Stepping," Soc. Neuroci. Abstr. CD ROM 32: 449.13.

[4] Lewinger, W. A., Rutter, B. L., Blümel, M., Büschges, A., and Quinn, R. D. "Sensory Coupled Action Switching Modules (SCASM) generate robust, adaptive stepping in legged robots," in CLAWAR 2006: 9th International Conference on Climbing and Walking Robots.

[5] Lewinger, W.A., Rutter, B.L., Quinn, R.D., "Irregular Terrain Navigation and Leg Coordination Improve Walking Behavior for Small Legged Robots", Adaptive Motion in Animals and Machines (AMAM) 2008, Cleveland USA

[6] Rutter, B.L., Bender J.A., Taylor, B.K, Ritzmann, R.E., Quinn, R.D. (2008) "Experiments in Locomotion with Neuromechanically Based Robotic Insect Models" Soc. Neuroci. Abstr. 198.7.

[7] Rutkowski, A.J., "A Biologically-Inspired Sensory Fusion Approach to Tracking A Wind-Borne Odor in Three Dimensions", Ph.D Dissertation, Department of Mechanical and Aerospace Engineering, Case Western Reserve University, January 2008.

[8] Edwards, S., et al. *Moth-Inspired Plume Tracking Strategies In Three-Dimensions*. in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. 2005.

[9] Porter, M.J. and J.R. Vasquez. *Bio-Inspired Navigation of Chemical Plumes*. in *Information Fusion, 2006 9th International Conference on*. 2006.

[10] Farrell, J.A., P. Shuo, and L. Wei, *Chemical plume tracing via an autonomous underwater vehicle.* Oceanic Engineering, IEEE Journal of, 2005. **30**(2): p. 428-442.

[11] Belanger, J.H., Arbas, E.A., "Behavioral Strategies Underlying Pheromone Modulated Flight in Moths: Lessons Learned from Simulation Studies", Journal of Comparative Physiology A (2008), vol 183: pgs 345 – 360.

[12] Budick, S.A., Dickinson, M.H., "Free-flight responses of *Drosophila Melanogaster* to Attractive Odors", The Journal of Experimental Biology (2006), vol 209: pgs 3001 – 3017

[13] Bailey, J.K., Willis, M.A., Quinn, R.D., "A Multi-Sensory Robot for Testing Biologically-Inspired Plume Tracking Srategies", International Conference on Advanced Intelligent Mechatronics 2005, Monterey California, USA

A Confidence Measure for Segment Based Maps

Rolf Lakaemper
Temple University
Philadelphia, PA, USA
lakamper@temple.edu

ABSTRACT

Map confidence, or map quality based on regional consistency is an important measure to evaluate the quality of robot maps. It is classically handled analyzing occupancy grids, which is an unnatural choice if the map is not represented by data points, but by line segments. We define a map-confidence measure that is tailored for segment based maps, without leaving the compact data representation by segments. The presented confidence measure is not based on comparison to ground truth data, but evaluates the map (ground truth free) based on map consistency.

1. INTRODUCTION AND APPROACH

The interest in robot mapping based on higher geometric structures like linear elements is currently growing. Obvious advantages in runtime, memory efficiency and simpler mid-level analysis capability make such mapping approaches powerful competitors to the classic, point based techniques. Computing the confidence in scans, and evaluating a global map based on the confidence of aligned single scans in the context of the global map is an important task in robot mapping. This paper gives an example, how map confidence can be computed purely based on line segments. The presented approach evaluates the confidence in each single segment; it can be used to delete inconsistent segment data ('map cleaning'), as well as to score the quality of a given segment map based on segment consistency.

The core algorithm was originally designed as a processing module of a segment based robot mapping system (description of this system is part of a future publication). Its purpose there is to clean intermediate mapping results, consisting of a low number of aligned, segment based local maps, from inconsistent or noisy segments. In a straightforward manner, such a module can be extended to a global (or regional) confidence measure: the more consistent segments (in a certain region), the better.

An important design paradigm of the presented research is not to leave the very efficient and compact data representation by segments. Such an approach leads to multiple advantages compared to point/grid based methods:

- The segment based approach captures structural information. This information goes significantly further than the information of object presence, contained in raw point data. Figure 3, (b,c,d) shows examples: the red segments are detected as noise. In point density based confidence approaches like occupancy grids [2], the original data points would not only be labeled as correct (since the point density in this area is high), but would have enhanced the confidence in this region.
- The segment based approach is fast. In indoor environments or urban outdoor environments, a typical scan consists of $n < 20$ segments of sufficient length, while the number of data points is typically one to two orders of magnitude (factor $10 - 100$) higher. This becomes especially important when point relations between different scans have to be evaluated, which usually implies algorithms with runtime between $O(nlogn)$ and $O(n^2)$.
- The segment based approach is memory efficient. Compared to occupancy grids, the memory consumption is significantly lower.
- The segment based approach is precise. Segment endpoints don't have to be adjusted to a resolution parameter, hence there are no quantization errors. This is in contrast to grid based approaches.

The basic idea of the approach is to cluster segments, based on an inter segment-distance measure. The quality of clusters defines the confidence in the participating segments, which in turn defines the confidence in the entire map. The main steps are i) the definition of a perceptually consistent segment-distance measure, ii) the adaption of a classic clustering technique (hierarchical clustering) to gain a parameter free clustering system, and iii) a new measure for intra cluster consistency, which directly leads to the final goal, the confidence measure.

2. RELATED WORK

To the authors's best knowledge, there are no publications available about generic map evaluation, the reason being that map evaluation is highly task specific. Task specific map evaluation is usually performed in the broader environment of robot competitions, such as RoboCup [1] or the US Department of Energy Grand Challenge [13]. Test arenas, developed by the National Institute of Science and Technology (NIST) exist [4], as an effort to create robot maps in standard environments. These arenas were used in various events, e.g. the RoboCup Rescue competition and the Response Robot Evaluation Exercise [11].

An occupancy grid based evaluation tool, the Jacobs Map Evaluation Toolkit [3], was utilized in the RobocupRescue competition 2008. Aside from functionalities like ground truth map creation, it consists in its core of a metric comparing the (grid/pixel based) maps. In short, correspondences between foreground points of the evaluated map and a ground truth map are established. The correspondence quality is computed using the spatial distance of the corresponding points.

In contrast, the presented evaluation method does not perform a comparison to a ground truth map, but aims to analyze the consistency of a single map. Working on a higher data structure, line segments, it tries to capture regional structural properties. These are evaluated based on their ambiguity of representation: a single cluster represents a single feature, a high intra cluster distance can be interpreted as ambiguity, or low confidence.

A more general introduction and overview of benchmarking and evaluation in robotics is given in [10].

3. INTER SEGMENT DISTANCE

This section introduces a distance measure between pairs of line segments s_1, s_2. The basic idea of the distance measure is to merge two line segments to an 'average' segment $\bar{s}$. The distance is the merging cost, which consists of three parts:

- the angular distance between s_i and $\bar{s}$, $i = 1, 2$
- the spatial distance between s_i and $\bar{s}$, $i = 1, 2$
- the spatial distance between s_1 and s_2.

The first two parts penalize the amount of 'non collinearity' of the segments, the third part penalizes spatial distance. Although used as a distance measure between two segments, the design is based on comparison to a 'virtual' average segment. This is motivated by certain experiments, suggesting that human perception assigns or connects line segments to larger structures under certain circumstances. For example, two collinear, overlapping line segments are perceived as one line, i.e. both segments represent the same element and should therefore have a distance of zero (which is the case for our distance measure). Please note that such a distance measure is no metric. It already disobeys the most 'intuitive' axiom of the metric axioms, the *identity of indiscernibles* ($d(a, b) = 0 \leftrightarrow a = b$), since two non identical collinear segments s_1, s_2 with $s_1 \cup s_2 \neq \emptyset$ have a distance $d(s_1, s_2) = 0$. This fact becomes important for the choice of the clustering algorithm, see section 4

The definition of the measure is out of scope of this paper and will be part of a future publication. Figure 1 gives examples of segment configurations and resulting distances.

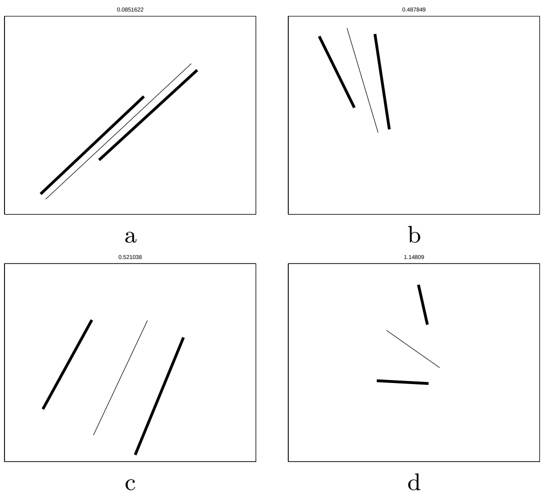


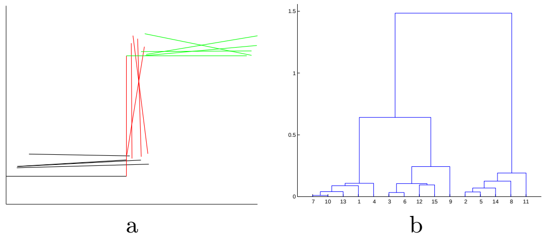
Figure 1: Segment configurations with increasing distance. a) 0.09 b) 0.49 c) 0.52 d) 1.14. The thin line is the merged segment. The increase in a)-c) results from larger intra segment distance, while d) results from angular distance.

4. CLUSTERING

For the clustering, agglomerative hierarchical clustering in 'single' mode is utilized. This method seeks to build a bottom up hierarchy of clusters, starting with each segment being a single cluster, ending in a single cluster containing all segments. Pairs of clusters are merged as one moves up the hierarchy. The merge is determined in a greedy manner: the two clusters with minimal distance are merged to a single one. Hierarchical clustering allows for different strategies to determine the distance of the newly emerged cluster to the remaining elements. In our case, we use the 'single mode' strategy: the distance between two clusters is the minimum distance between their elements. There is a geometric motivation for the use of this mode: in the example of collinear, slightly overlapping segments single mode clusters these segments to a single group — intuitively, single mode clustering acts like a connected components algorithm, the necessary topology being defined through the distance measure (small distance = neighbors).

Hierarchical clustering has two main properties which suggest its use in the segment merging context: first, it is, in its first stage, parameter free, i.e. no pre-defined number of clusters has to be determined. Parameters might be introduced later in a follow up stage, which selects the level of clustering (agglomerative hierarchical clustering always ends in a single cluster). Second: it is simply based on mutual distances between the data points (here: line segments), yet without the need to embed them in a metric space. This means, hierarchical clustering can deal with any distance measure (especially non-metrics, as in the given case).

We want to illustrate the segment clustering by a simple example, see Figure 2. The data set of this example consists of 15 segments, which can intuitively be combined to 3 clusters. Figure 2,b), shows the resulting *dendrogram.* Each horizontal bar shows the *linkage* L_i between two clusters, L_i is assigned the minimal distance between elements of the left and right subtree of the linkage; in the dendrogram this cost is displayed by the height of the bar. In this simple case, the dendrogram clearly suggests the three clusters. The critical step in hierarchical clustering is to define the step to end the clustering process. We do so if a potential merge de-



a b

Figure 2: Clustering example. Left: Segments. Right: Dendrogram. See text for details.

creases the intra cluster consistency significantly. To determine clusters, we assign a *consistency value* $c(L_i)$ to each linkage L_i. $c(L_i)$ compares the linkage distance L_i with all linkage distances L_i^l, L_i^r of the left and right subtree:

$$L_i = \frac{L_i - \mathrm{mean}(L_i^l \cup L_i^r)}{\mathrm{mean}(L)} \tag{1}$$

with L being the set of all linkages. Data elements s_1, s_2 (segments) belong to one cluster if all linkages connecting s_1 and s_2 do not exceed a certain threshold T_c. Normalizing by $\mathrm{mean}(L)$ makes the approach scale independent. Our consistency measure has a clear geometric motivation, and performed well in different examples (see results in section 7). Determining clusters from dendrograms can be performed in different ways. More details about hierarchical clustering can be found e.g. in [5].

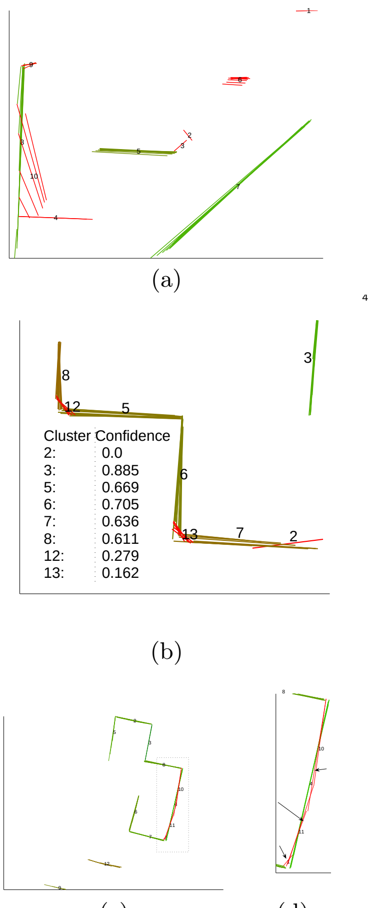
5. CLUSTER QUALITY: THE CONFIDENCE MEASURE

The main step in our evaluation is to determine the consistency of each cluster. Please observe that we already computed an intra cluster consistency value $c(L_i)$ to determine the clusters. $c(L_i)$ has certain drawbacks handling outliers, it is not necessarily consistent with the perceptual consistency. We therefore introduce a new intra-cluster-consistency measure $\mathcal{C}$ which is stronger perceptually motivated and adjusts better to the specific problem. $\mathcal{C}$ is used to re-evaluate each cluster, it is, however, too expensive to be utilized in the clustering process itself. It is therefore only applied after the clustering process is finalized.

In $\mathcal{C}$, collinear structures are favored, while clusters containing wide-spread segment sets are penalized. Similar to the segment distance measure, each segment in the cluster is compared to an average cluster segment, the cluster representative. In analogy to classic intra cluster consistency measures, the angular and spatial distance to this representative is taken into account to determine the cluster consistency. Intuitively, all angular distances of segments in one cluster to the average cluster representative are computed, as well as the transitional distances. For angular and translational distances, two separate confidence measures $\mathcal{C}_a, \mathcal{C}_t \in [0..1]$ (angular/translational respectively) are computed (see details below). The final confidence $\mathcal{C}$ is computed as

$$\mathcal{C} = \min(\mathcal{C}_a, \mathcal{C}_t). \tag{2}$$

A high confidence (1) is therefore only assigned if both, angular and translational confidence are high. Additionally, clusters must contain a certain minimal number of segments (in the current system: three segments), otherwise they are assigned a confidence of $\mathcal{C} = 0$. Figure 3 shows examples for clusters and their consistency value $\mathcal{C}$. Please note that especially Figure 3 shows the superiority of a segment based evaluation to point based occupancy grids. Figure 4 is a comparative example showing the performance of the two confidence measures $c(L_i)$ and $\mathcal{C}(C_i)$: the tendency of both measures is approximately equal (this is why we can use the computationally cheaper $c(L_i)$ during the clustering), yet $\mathcal{C}$ yields more perceptually consistent results.



(a)

(b)

(c) (d)

Figure 3: Evaluating clusters using the confidence measure $\mathcal{C}$. The red/green-ness is determined by confidence (the greener, the more confident). a) regions with non matching angles, widespread structures and areas of insufficient density are marked as non confident. b) segment based confidence detects structural inconsistency: the 45 degree corner scans are detected as inconsistent. d) a magnified view of the marked part of c): the correctly detected inconsistent segments have a huge overlap with consistent segments: detection of such areas is not possible with occupancy grids, but only with methods detecting underlying structural information.

5.1 Angular Confidence $\mathcal{C}_a$

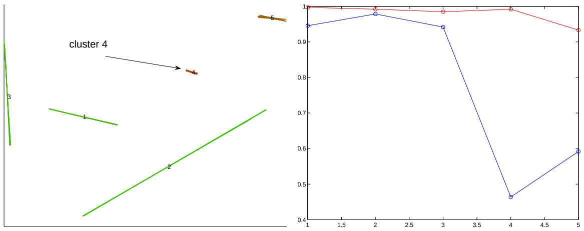


Figure 4: Left: A simple example of a map with high confidence in all 5 regions (clusters). Right: (blue) cluster quality using the distance matrix based method $c(L_i)$, evaluating cluster 4 (arrow in left figure) to be of lower quality; (red)$\mathcal{C}$, which is in accord with the perceived high consistency of all clusters.

For all segments s_i of a single cluster L, we compute their angles $a_i \in [-\pi/2..\pi/2]$ with the x-axis. We define the cluster's average angular direction θ using the weighted circular mean wcm

$$\theta = wcm(2a_i, |s_i|)/2 \tag{3}$$

where $|s_i|$ denotes the length of segment s_i, used as the corresponding angle's weight (multiplication/division by 2 ensures correct handling of segment directions). The angular distance d_a is computed as the weighted (by length) sum of distances to θ:

$$d_a = \frac{\sum_i \min((|a_i - \theta|) \mod \pi,\ \pi - ((|a_i - \theta|) \mod \pi))\ l_i}{\sum_i l_i}. \tag{4}$$

Finally, the angular confidence is computed as

$$\mathcal{C}_a = \exp \frac{d_a^2}{2\sigma_a^2} \tag{5}$$

with a parameter $\sigma_a = 0.05$ which was experimentally determined and fixed.

5.2 Translational Confidence $\mathcal{C}_t$

For the translational confidence, we compute the maximal distance t_i of each segment to a cluster-representative line S. S is defined by θ and a point $P = \sum(p_i)/\#L$, the average center point (p_i: center point of s_i, $\#L$: number of segments in cluster L). The translational distance is defined by

$$d_t = \frac{\sum t_i}{\#L} \tag{6}$$

Finally, the angular confidence is computed as

$$\mathcal{C}_d = \exp \frac{d_t^2}{2\sigma_t^2} \tag{7}$$

with a parameter $\sigma_a = 0.1$ which was experimentally determined and fixed. Observe that σ_t is scale dependant. The current value is determined for robot maps with scale unit of one meter.

6. MAP EVALUATION

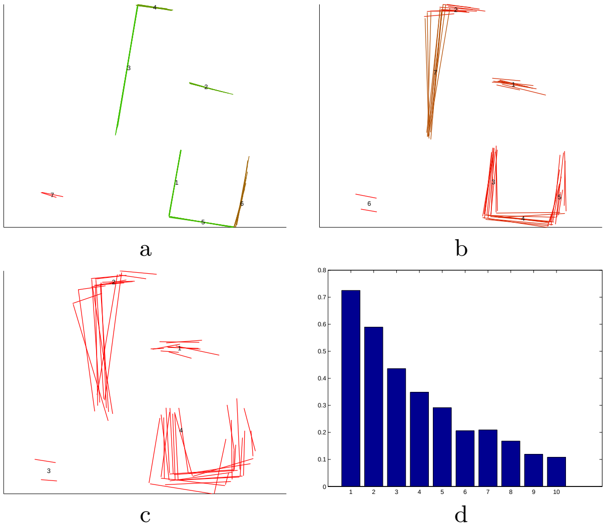
It is a small step from regional evaluation of single clusters to global map evaluation. Given all clusters C_i along with their confidence measure $\mathcal{C}(C_i)$, we define the global confidence $\mathcal{M}$ of a map by

$$\mathcal{M} = \frac{\sum_i \#C_i\ \mathcal{C}(C_i)}{\sum_i \#C_i} \tag{8}$$

with $\#C_i$ denoting the cardinality of C_i. $\mathcal{M}$ computes the average consistency of all segments, defining the confidence of a segment by the consistency of the cluster it participates in.

7. RESULTS

7.1 Random Distortion


a

b

c

d

Figure 5: Global map evaluation using $\mathcal{M}$. A map with high confidence was randomly increasingly distorted in 10 steps (Figure shows step 1(a),5(b) and 10(c)). (d) shows the global confidence diagram for the resulting maps, x-axis: step 1-10, y-axis: confidence $\mathcal{M}$.

In this experiment, a map with high confidence was randomly increasingly distorted in 10 steps. Figure 5 shows steps 1, 5 and 10 and the confidence measures $\mathcal{M}$ for each of the 10 distortion levels. Expectedly, the results show decreasing confidence.

7.2 Map Comparison

In this experiment, we compare results of two mapping algorithms. The first algorithm [6] is a point based alignment (not segment based) algorithm. However, it results in corrected poses of single scans. We used an algorithm explained in [8] to extract segments from these single scans, and superimposed them, using the corrected poses (Figure 6,a). The second map (Figure 6,b) was computed by a new, segment based algorithm, which will be topic in a future publication. Both output maps consist of the same single scans' segments, yet aligned using different poses. It can clearly be seen that the first map is less consistent. Our evaluation algorithm does not only capture the overall difference in quality (quality of first map: $\mathcal{M} = 0.2769$, quality

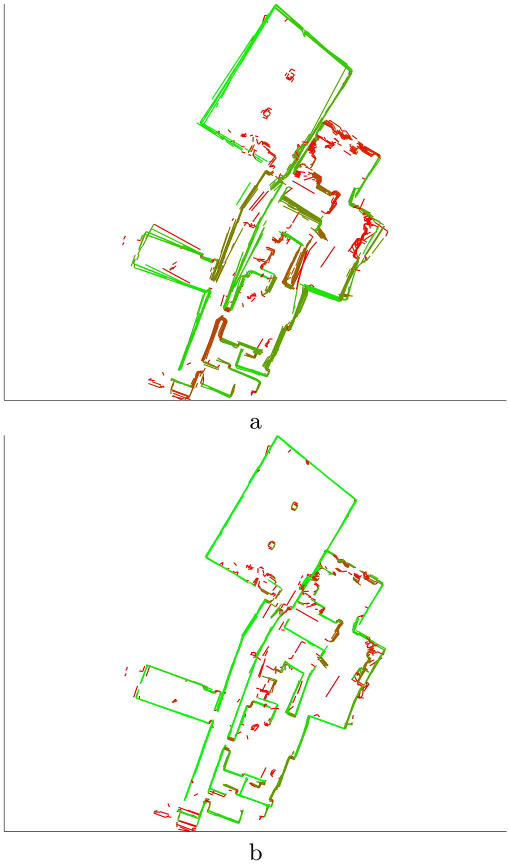


Figure 6: Evaluation of two maps of the data set 'Freiburg082'. a) Quality of map $\mathcal{M} = 0.2769$. b) $\mathcal{M} = 0.4355$. Colors: level of green (vs red) shows confidence: The greener, the more confident, the more red, the worse. The higher regional confidence in (b) leads to the better total confidence value.

of second map: $\mathcal{M} = 0.4355$), but also identifies the confidence of single clusters (regions). It is interesting to show segments above a certain confidence level only (Figure 7). In this data set, this leads to structural de-noising of the map: usually, large and static (in contrast to smaller and/or moving) objects in the environment yield high confidence representation. Therefore, the main structure of the environment is highlighted (of course, using higher quality clusters only, the map quality based on our evaluation measure increases). Additional merging of the clusters to single segments yields a clear map in a very compact representation (here: 80 segments). We performed a second comparative experiment on a different data set (data set NIST), using the mapping algorithms FFS [9] and FFS with Virtual Scans [7]. The latter one is an extension of the first, and leads to (visually inspected) improved results. Numerical evaluation of the results using the presented measure is consistent with the visual impression, see Figure 8. The maps only differ slightly in certain regions. However, the overall visual impression of (b) is slightly better than the one of (a), which is also expressed in the evaluation. The experiments leading to the respective maps are documented in [7].

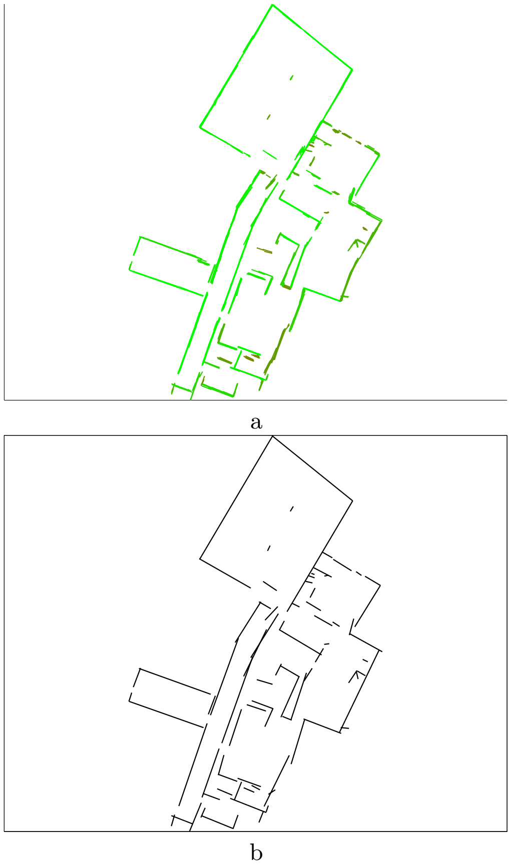


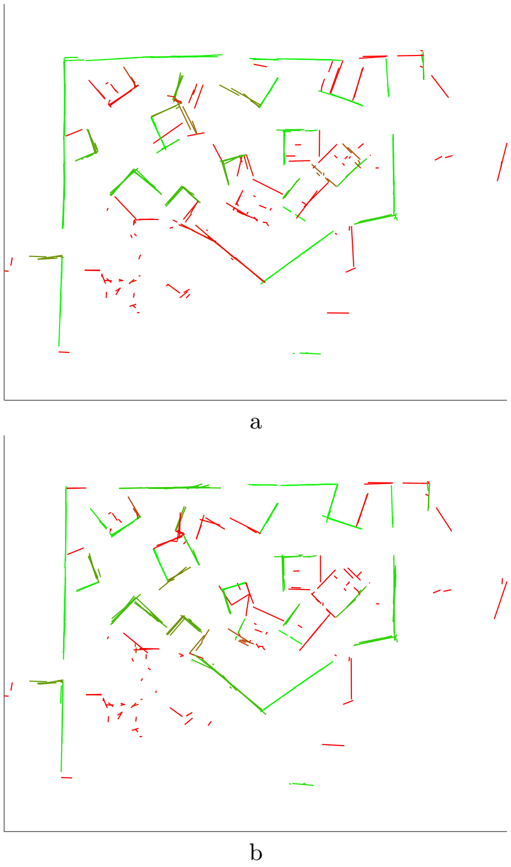
Figure 7: Using segments of high confidence only yields structural de-noising. a) segments of Figure 6,(b), belonging to clusters L_i with a confidence $c(L_i) > 0.3$ (80 clusters, overall confidence $\mathcal{M} = 0.6554$). b) clusters of (a) represented by single representative merged segments (80 segments).

8. RUNTIME

The presented algorithm has an order of magnitude of $O(n^2)$, n = total number of segments, which results from computation of the pairwise segment distance matrix. The MATLAB implementation of the algorithm needed 1 second for the experiment using data set NIST (332 segments), and 5 seconds for the experiment using data set Freiburg082 (1975 segments), both on a 1.8GHz laptop PC.

9. CONCLUSION AND OUTLOOK

The presented confidence measure evaluates maps in consistency with visual perception. In its core, it uses a classical clustering algorithm, hierarchical clustering, which is adapted to the current problem utilizing a segment distance measure and a segment based cluster confidence measure. Since segment based representation captures structural features better than its lower representation counterpart, point based maps, erroneously mapped/aligned features can be detected even if they overlap with correct features. This leads to detection of structural consistency, which is the main property evaluated by the presented approach. With a re-definition of segment distance and cluster confidence, the



a

b

Figure 8: Mapping of data set NIST using algorithms FFS (a) and FFS with Virtual Scans (b). The evaluation leads to values of $\mathcal{M} = 0.3386$ (a) and $\mathcal{M} = 0.3876$ (b), reflecting the slight visual improvement of (b) over (a).

approach is extendable to 3D, which makes it interesting for 3D mapping algorithms based on planar elements, e.g. [12].

10. ACKNOWLEDGEMENTS

Thanks to Alexander Kleiner, University of Freiburg, for the data set 'Freiburg082'.

11. REFERENCES

[1] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo. Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 24(11-12):943–967, 2007.

[2] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[3] I. Varsadan, A. Birk, and M. Pfingsthorn. Determining Map Quality through an Image Similarity Metric. In *Proceedings of the RoboCup Symposium*, July 2008.

[4] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa. Test arenas and performance metrics for urban search and rescue robots. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3396–3403 vol.3, Oct. 2003.

[5] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, September 1967.

[6] A. Kleiner and C. Dornhege. Real-time localization and elevation mapping within urban search and rescue scenarios: Field reports. *J. Field Robot.*, 24(8-9):723–745, 2007.

[7] R. Lakaemper. Improving sparse laser scan alignment with virtual scans. In *International Conference on Intelligent Robots and Systems (IROS08)*, Nice, France, September 2008. IEEE.

[8] R. Lakaemper. Simultaneous multi-line-segment merging for robot mapping using mean shift clustering. In *International Conference on Intelligent Robots and Systems (IROS09)*, St Louis, MO, USA, September 2009. IEEE.

[9] R. Lakaemper, N. Adluru, L. Jan Latecki, and R. Madhavan. Multi robot mapping using force field simulation: Research articles. *J. Field Robot.*, 24(8-9):747–762, 2007.

[10] R. Madhavan, R. Lakaemper, and T. Kalmar-Nagy. Benchmarking and standardization of intelligent robotic systems. In *14th International Conference on Advanced Robotics (ICAR 2009)*, Munich, Germany, June 2009.

[11] NIST. NIST Response Robot Evaluation Exercise. Search and Rescue: Texas Engineering Extension Service (TEEX), November 2008.

[12] K. Pathak, N. Vaskevicius, J. Poppinga, M. Pfingsthorn, S. Schwertfeger, and A. Birk. Fast 3d mapping by matching planes extracted from range sensor point-clouds. In *International Conference on Intelligent Robots and Systems (IROS09)*, St Louis, MO, USA, September 2009. IEEE.

[13] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692, 2006.

Evaluation of RoboCup Maps

Benjamin Balaguer, Stefano Carpin
School of Engineering
5200 North Lake Road
University of California, Merced, USA
+1(209)228-4152
{bbalaguer,scarpin}@ucmerced.edu

Stephen Balakirsky
NIST
100 Bureau Drive
Gaithersburg, MD, USA
+1(301) 975-4791
stephen@nist.gov

Arnoud Visser
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam, NL
+31 (20) 525-7532
A.Visser@uva.nl

ABSTRACT

This paper describes the steps taken to create the scoring criteria aimed at measuring the quality of maps produced by teams participating in the RoboCup Rescue Virtual Robots competition. Since metrics have already been developed by a few research groups, we start by highlighting the most popular solutions to this problem, emphasizing their strengths and weaknesses. Having put the difficulty of creating map benchmarks into perspective, we present our map benchmark suite, appropriate for Urban Search and Rescue missions, along with examples taken from former competitions.

Categories and Subject Descriptors

F.2.3 [**Theory of computation**]: Analysis of Algorithms and Problem Features—*Tradeoffs among complexity measures*

General Terms

Performance, Measurement

Keywords

Map evaluation, RoboCup, Performance Metrics

1. INTRODUCTION

One of the competitions in RoboCup is the Virtual Robot Rescue league, where participants are called upon to deploy teams of robots capable of locating victims and hazards in unstructured areas. As opposed to other RoboCup competitions [17, 20, 1, 11], the Virtual Robot Rescue league asks robot teams to map unknown environments, with little a priori information. The theme behind the league is Urban Search and Rescue (USAR), where robots are deployed in disaster scenarios (e.g. earthquakes, landslides) and have to work cooperatively as unified teams while taking into account humans, whether they be victims or first responders (e.g. firefighters, rescue teams). As such, the maps that are generated by the robots need to incorporate useful information that first responders can exploit, adding a new dimension to robot mapping. Indeed, robots now have to generate multiple maps, some of which are for their own needs (e.g. navigation) while others are explicitly for first-responders (e.g. victim locations with safest paths to reach them).

The Virtual Robot Rescue League uses the Unified System for Automation and Robot Simulation (USARSim) [7] to simulate disaster scenarios. The simulation is realistic due to a community of users and developers who strive to validate each robot, sensor, or other physical properties [6, 8, 13]. This community involvement created a remarkably accurate simulation capable of modeling multifaceted disaster environments ranging from traffic accidents to earthquakes and explosions, each possibly exploiting the effects of smoke, fires, debris, water, to name a few. In addition to the near-zero participation cost and the ability to create realistic city-sized disasters, the simulation offers ground truth data that would otherwise be difficult to gather. The large amount of robotic platforms and sensors in USARSim translates into a challenging situation for map scoring. Indeed, each team solves the mapping problem differently using a diverse set of robots and sensor configurations, resulting in a massive mismatch between maps, from scaling to rotational differences. This puts us in the unique position of having to come up with a map benchmark robust enough to take into account all of these differences along with the opportunity of having a tremendous amount of data available.

Evidently, and despite the fact that it is still frequently employed, a qualitative approach is fundamentally insufficient for a competition where results have to be both repeatable and reliable. Not wanting to develop a map benchmark from scratch, and optimistically hoping that a solution had already been published, we performed an extensive case study, a part of which is shown in Section 2. Realizing that no current solution was robust enough for the problem at hand (i.e. teams would be able to take advantage of the metrics' weaknesses), we developed a mapping benchmark suite comprised of standards and categorized metrics, which are described in Section 3 and 4, respectively. It is worthwhile to note that the standards were so well received that they have subsequently been implemented as part of the Real Robot Rescue League. We close the paper with concluding remarks and possible future work in Section 5. While this paper focuses on mapping, a companion paper highlights the overall RoboCup 2009 competition [3].

2. CASE STUDY

Map benchmarking is a relatively novel effort, and so is robot benchmarking in general. Therefore, the amount of formerly published scholar work is rather limited (the reader is referred to a forthcoming special issue of the Autonomous Robots journal on *Characterizing mobile robot localization and mapping*). In this section, we quantitatively compare some of the most popular benchmark metrics that have been previously published. We run the metrics with two binary occupancy grid maps, one generated by a robot and the other being ground truth. Each grid cell can only have a value of 1 for occupied space or a value of 0 for free space. Please note that the discussion in this section is entirely based on our binary map representation and that results might be different with a probabilistic occupancy grid map. Even though we have performed a full case study on different environments, we only present a representative example in Fig. 1 and Table 1 due to space constraints.

The first set of four metrics, namely the Map Score [14], Overall Error [5], Normalized Map Score [16], and Occupied Map Score [16], represents an approach requiring pixel-to-pixel comparisons between the ground truth and robot-generated maps. The Map Score metric counts the number of ground truth and robot map pixels that are the same. As such, a perfect score would be obtained when the Map Score metric equals the number of pixels in the map. The Overall Error metric counts the number of ground truth and robot map pixels that are different, where a perfect score would be zero. The Map Score metric measures accuracy whereas the Overall Error metric measures error and that adding both metrics together will equal the total number of pixels. The two aforementioned metrics are utilized over all the pixels, regardless of what they represent (i.e. occupied or free space). Consequently, the two metrics are biased towards maps with large regions of correct free space, as shown in Fig. 1 and Table 1. From the table, Team A and Team E have the best scores and, looking at the figure, the bias is clear: the two maps with the smallest amount of discovered walls receive a higher score. Research groups have attempted to remove this bias by introducing the Normalized Map Score and Occupied Map Score metrics. They work the same way as the Overall Error metric (i.e. looking for pixel mismatches) but are only run on the occupied space of the maps. The Normalized Map Score runs on the occupied space of the ground truth map whereas the Occupied Map Score runs on the occupied space of the robot-generated map. Unfortunately, these metrics only move the bias, which is now dependent on the occupied space. Using the Normalized Score metric, the robot maps that have thick walls do better, as shown by Team C and Team E, since they do a better job in replicating the wall thickness of the ground truth map. In contrast, Team A and Team B do better with the Occupied Map Score metric, due to their thin walls that allow for a greater margin of error when compared to the thicker ground truth walls.

Another interesting pixel-to-pixel approach is presented through the Picture-Distance function [4]. In this metric, the score represents the Manhattan-distance between an occupied pixel in the ground truth map and the closest occupied pixel in the robot-generated map. The process is repeated over all the occupied pixels of 1) the ground truth map and 2) the robot-generated maps. Finally, the result is normalized by dividing it by the total number of pixels considered. The Picture-Distance function is a measure of map error and, as such, the best possible score is zero. A look at Fig. 1 and Table 1 quickly shows that the two teams who have explored the most, Team C and Team D, do better with this metric. From both the method used and the experiment performed, it is clear that the method is also biased, towards exploration (i.e. wall discovery).

Moving away from the bias of pixel-to-pixel comparisons brings us to correlation coefficients, a comparison measures valued between -1 and 1, with -1, 0, and 1 representing perfect inverse correlation, no correlation, and perfect correlation, respectively. The Baron's Cross Correlation coefficient [16] attempts to correlate two images by using the ground truth and robot-generated pixels' mean and standard deviation. Since averages are used, and the pixel's values can only be 0 or 1, the Baron's coefficient rewards robot maps that have a similar number of occupied and free pixels to the ground truth. Consequently, the coefficient is influenced both by wall thickness and exploration, as can be seen in Fig. 1 and Table 1 where Team C and Team E have the highest scores. The Pearson's Correlation coefficient [12] evaluates the occupied space of the map as a spatial function, trying to linearly describe one map from the other. The Pearson's coefficient requires an approximately similar point distribution between the two map. This drawback is evidenced by the results for Team A and Team E, where, even though both maps are very similar they have extremely different Pearson's coefficients. It is worthwhile to note that both correlation coefficients can be unpredictable, as shown by the scores of Team A and Team B.

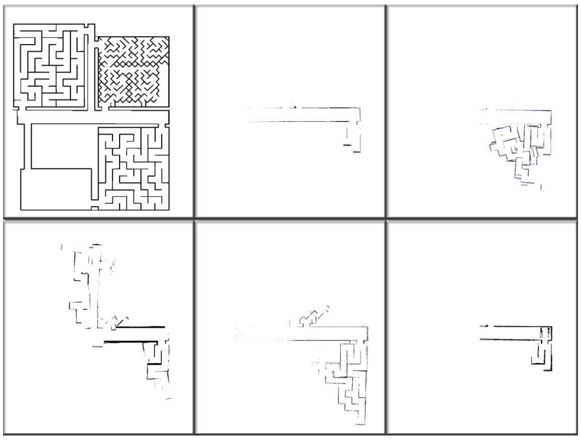


Figure 1: Example set of maps used for the Case Study, the results of which are in Table 1. The first image is the ground truth with the remaining images being, from left to right and up to down, Team A, Team B, Team C, Team D, and Team E, respectively.

3. MAP REPRESENTATION STANDARDS

One of the principal obstacles impeding the development of a consistent map benchmark comes from the lack of standards between the large amount of mapping algorithms that have been developed, through the years, by various research groups. Indeed, each algorithm works differently, from the way they represent maps (e.g. occupancy grids, topological, feature-based, etc...) to the different scales and rotations

Metric	Team A	Team B	Team C	Team D	Team E
Map Score [14]	**586779**	586192	585049	585297	**586815**
Overall Error [5]	**56577**	57164	58307	58059	**56541**
Normalized Map Score [16]	56065	55785	**55227**	55363	**54367**
Occupied Map Score [16]	**512**	**1379**	3080	2696	2174
Baron's Correlation [16]	-0.005	0.017	**0.036**	0.032	**0.098**
Pearson's Correlation [12]	0.298	-0.060	**0.479**	0.295	**0.591**
Picture-Distance [4]	210.09	254.37	**129.89**	**189.44**	221.61

Table 1: Metrics comparison for the maps shown in Fig. 1. The seven rows represent each metric taken from different publications. The bold font shows the two best results for a specific metric.

that they may encompass. Having to rank maps generated by many different robotics groups and, as a consequence, facing the same map representation problems, we have imposed two mapping standards on participants, the GeoTIFF [19] image format and the MIF [9] vector format. We have found, over the years, that participants embrace them, primarily for their ease-of-use, while giving the administrators powerful tools to generate a fair mapping benchmark.

3.1 GeoTIFF Image Format

The GeoTIFF image format embeds geographical information as an integral part of the map. The power of GeoTIFF lies in its ease of use, open standard, and layering capabilitites. Indeed, it is very simple to geo-reference any map, by providing an additional file comprised of six parameters, namely the X and Y positions of the upper-left pixel, the scale of a pixel in the X and Y directions, the rotation, and the skew. These six parameters take into account any potential differences in scale, translation, and rotation between maps. GeoTIFF is an open standard, a fact that translates into a plethora of open tools that work across different platforms and programming languages. Last but not least, it is very easy to embed multiple layers on top of the original map, a powerful way to display varied information on the maps. Evidently, from a map benchmark standpoint, the GeoTIFF image format allows every map, including ground truth, to be overlaid on top of each other, as shown in Fig. 2; making it straightforward to evaluate the maps either quantitatively or qualitatively.

3.2 MIF Vector Format

The MIF vector format is similar to the GeoTIFF format in that it possesses the same qualities of allowing georeferencing, remaining easy to use, being an open standard, and working well with layers. The difference between the two, however, lies in what can be represented. Whereas GeoTIFF represents images, MIF works with geometric primitives (e.g. points, lines, polygons) that can have an arbitrary number of attributes. The MIF vector format can be best exploited to display topological or feature-based maps, where labeled nodes or features can give high-level information or particular landmarks of interest to first responders. Fig. 3[1] shows some examples of what can be achieved with a MIF vector file.

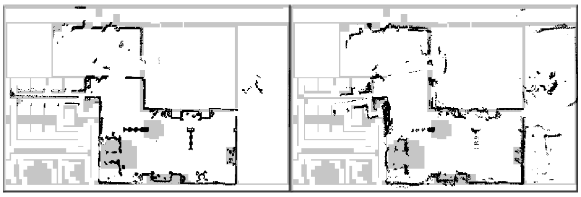
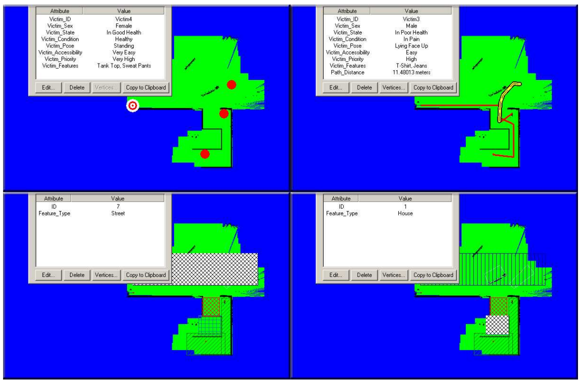


Figure 2: Examples of two robot-generated maps (black) overlaid on top of the ground truth map (gray) for an indoor environment.



Attribute	Value
Victim_ID	Victim4
Victim_Sex	Female
Victim_State	In Good Health
Victim_Condition	Healthy
Victim_Pose	Standing
Victim_Accessibility	Very Easy
Victim_Priority	Very High
Victim_Features	Tank Top, Sweat Pants

Attribute	Value
Victim_ID	Victim3
Victim_Sex	Male
Victim_State	In Poor Health
Victim_Condition	In Pain
Victim_Pose	Lying Face Up
Victim_Accessibility	Easy
Victim_Priority	High
Victim_Features	T-Shirt, Jeans
Path_Distance	11.48013 meters

Attribute	Value
ID	7
Feature_Type	Street

Attribute	Value
ID	1
Feature_Type	House

Figure 3: Four examples of MIF vector files, overlaid on top of the robot-generated map. The upper-left picture shows points representing victims' location labeled with various information about each victim. The upper-right picture shows line segments highlighting the best path to reach each victim, labeled with the victim's information and path's length. The lower pictures display regions of interests, including a street (left) and a house (right).

4. MAP BENCHMARK

It is clear from the Case Study that no published algorithm is adequate on its own or as part of a map benchmarking suite. They each have some sort of bias and cannot solve the problem of error propagation, the toughest challenge when evaluating maps, where similar mapping errors can affect maps differently depending on when the error occurred. For example, an orientation error at the beginning of a mission will result in a map that is wrong through the rest of the mission, whereas the same orientation error at the end of the mission will affect a much smaller portion of

[1] The text in the figures is provided to give the readers an idea of the amount of information that can be included as part of the MIF formats. It does not need to be read.

the map. It is our belief that the maps should be equally deserving, provided that everything else is equal. Additionally, a map is application-specific and, in our case, both the USAR scenario and the first responders have to be considered as part of the map benchmark. As such, we devised a categorized benchmark comprised of Metric Quality, Skeleton Quality, Attribution, Grouping, Utility, and Creativity. Each category possesses a weight, the combination of which can be used to steer the competition towards one or more research agendas.

4.1 Metric Quality

The Metric Quality tries to solve the same problem that was studied in the Case Study: the comparison of the robot-generated occupancy grid map to ground truth, from an accuracy standpoint. In order to bypass the aforementioned problem of error propagation, we further divide the Metric Quality into Global and Local Quality. The Global Quality is a measure of the number and severity of mapping errors whereas the Local Quality is a measure of accuracy between these mapping errors. Using Fig. 4 as an example, one can see that both robot-generated maps are similar in terms of Global Quality, each having a small error with the lower hallway. The right map, however, is worst in terms of Local Quality, since it is missing some walls in the center of the map.

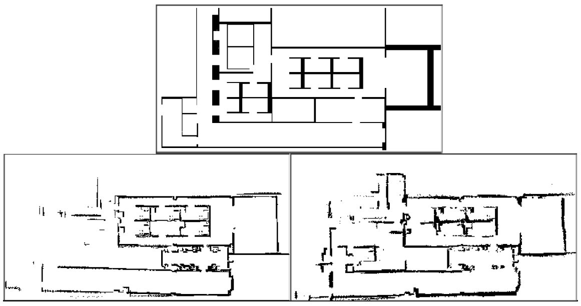


Figure 4: Example for the Metric Quality evaluation, where the upper map is ground truth and the lower-left and lower-right maps are different robot-generated maps.

4.2 Skeleton Quality

The Skeleton Quality evaluates a topological map rather than an occupancy grid map, which can be more useful to first responders. A first responder should be able to follow a skeleton map to reach a chosen point. In this case, the quality is determined from the number of false positives and false negatives. A false positive occurs when a node cannot be accessed whereas a false negative takes place when a clear topological location is available but has not been included in the skeleton map. Fig. 5 shows examples of skeleton maps with similar qualities. The first map has a lot of false positives in the lower and right sections of the map, where topological locations have been identified in unexplored space. The second map contains both false positives, where a topological node is inside a wall, and false negatives, along the left side of the hallway.

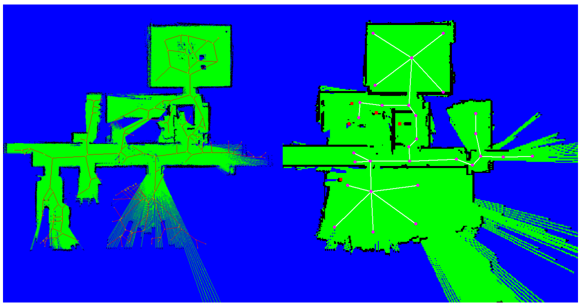
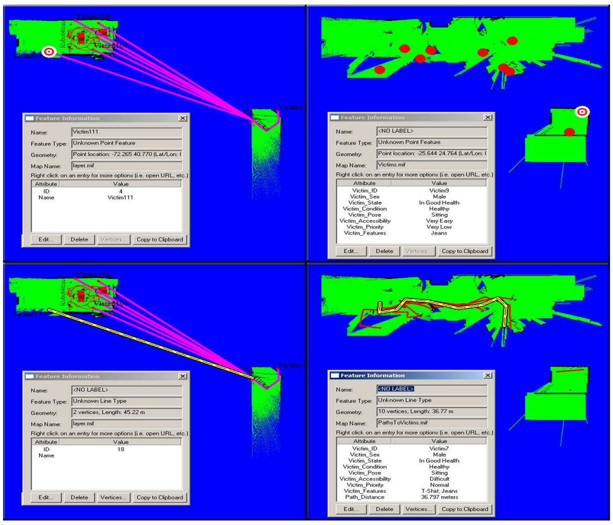


Figure 5: Example for the Skeleton Quality evaluation, with two different robot-generated maps.

4.3 Attribution

The Attribution section of our mapping benchmark aims to reward teams that can successfully deliver a feature-based map with valuable information for first responders. The type of information that can be embedded into the map is fairly open, even though most teams deliver feature-based maps indicating victim locations and information, best paths to reach victims, robot paths, and important landmarks. The Attribution is scored based on the amount and accuracy of the data. As an example, Fig. 6 shows two maps, each providing victim locations and best paths to reach them. Both maps provide accurate victim locations but the left one offers a lot more information about the victim, ranging from the sex, the condition, the priority given to get rescued, the ease of accessibility, etc... Similarly, both maps provide paths to reach the victims but the paths of the left map are inaccurate, going through a section of unexplored space. Based on this example, the right map would get a better score.



Name: Victim111
Feature Type: Unknown Point Feature
Geometry: Point location: -72.265 40.770 (Lat/Lon: (
Map Name: layer.mif
Right click on an entry for more options (i.e. open URL, etc.)

Attribute	Value
ID	4
Name	Victim111

Name: <NO LABEL>
Feature Type: Unknown Point Feature
Geometry: Point location: -25.644 24.764 (Lat/Lon: (
Map Name: Victims.mif
Right click on an entry for more options (i.e. open URL, etc.)

Attribute	Value
Victim_ID	Victim9
Victim_Sex	Male
Victim_State	In Good Health
Victim_Condition	Healthy
Victim_Pose	Sitting
Victim_Accessibility	Very Easy
Victim_Priority	Very Low
Victim_Features	Jeans

Name: <NO LABEL>
Feature Type: Unknown Line Type
Geometry: 2 vertices, Length: 45.22 m
Map Name: layer.mif
Right click on an entry for more options (i.e. open URL, etc.)

Attribute	Value
ID	18
Name	

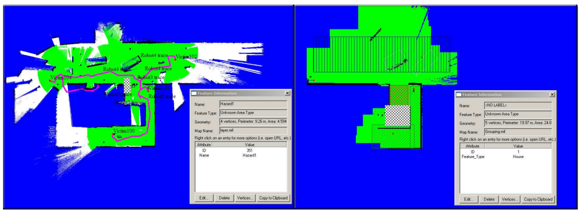
Name: <NO LABEL>
Feature Type: Unknown Line Type
Geometry: 10 vertices, Length: 36.77 m
Map Name: PathsToVictims.mif
Right click on an entry for more options (i.e. open URL, etc.)

Attribute	Value
Victim_ID	Victim7
Victim_Sex	Male
Victim_State	In Good Health
Victim_Condition	Healthy
Victim_Pose	Sitting
Victim_Accessibility	Difficult
Victim_Priority	Normal
Victim_Features	T-Shirt, Jeans
Path_Distance	36.797 meters

Figure 6: Example for the Attribution metric for two different-robot generated maps. The left and right columns each represent a different robot-generated map. The first row shows the victims' attribution while the second row shows the victim paths' attribution.

4.4 Grouping

The Grouping metric is very similar to the Attribution in that it is, essentially, a feature-based map aimed at helping first-responders better navigate the environment. It differs in that instead of being point-based, it groups and labels regions of space. Grouping stems from the fact that a section of occupied pixels represents particular landmarks that can be labeled. Fig. 7 offers a contrasting example, where the left map is comprised of a single group labeled ”Hazard” and the right map contains many different groups labeled as ”House”, ”Street”, ”Vehicle”, among others. Once again, the metric is scored based on the amount and accuracy of the information provided and, in this example, the right map would receive a better score than the first one.



Name: Hazard1
Feature Type: Unknown Area Type
Geometry: 4 vertices, Perimeter: 9.26 m, Area: 4.594
Map Name: layer.mif
Right click on an entry for more options (i.e. open URL, etc.)

Attribute	Value
ID	351
Name	Hazard1

Name: <NO LABEL>
Feature Type: Unknown Area Type
Geometry: 5 vertices, Perimeter: 19.97 m, Area: 24.8
Map Name: Grouping.mif
Right click on an entry for more options (i.e. open URL, etc.)

Attribute	Value
ID	1
Feature_Type	House

Figure 7: Grouping example with two different robot-generated maps.

4.5 Utility

The map Utility takes a look at the overall information provided by the teams. In other words, the map Utility aims at answering the question of how useful are all the layers to a first responder. This metric regroups the other metrics together but looks at a larger scope, where teams have to balance the amount of information they provide with the way it would look on the screen. As more and more information is given, it is harder to display it neatly while still making it easy to understand. The clever use of layers greatly affects the utility of a given map.

4.6 Creativity

For the purpose of the competition, we have added an unorthodox metric that rewards teams for creative new ways of representing valuable information to first responders. Teams are given bonus points for innovative map layers that could help first-responders better do their jobs. In the past, a team came up with the geo-referencing of victims’ pictures, a layer that was quickly adopted by the rest of teams in later competitions. More recently, a team showed the best communication coverage attained while navigating the environment so that first-responders could replicate it should they need to establish a communication network. An example is shown in Fig. 8.

5. CONCLUSIONS

We have presented the necessary steps to come up with a fair map benchmarking suite capable of scoring maps produced by USAR robots working in close cooperation with first responders. We strongly believe in committing to easily-adoptable, yet powerful, open standards such as GeoTIFF that take little additional work from programmers while providing great benefits. Similarly, we value open-source development by requiring teams to provide public access to their software and encouraging participants to share and reuse code and ideas. In that sense, the competition can be viewed more as an open workshop where teams are equally looking to learn as they are to win. From a benchmarking standpoint, the open-source phenomenon brings an interesting component, where algorithmic progress can easily be measured from year to year due to the fact that the software is both available and archived. We hope that the community would follow in our footsteps and make algorithms and data sets public, so that benchmarks can be accepted and evaluated by an entire community rather than a relatively small research group. Two projects going in that direction are OpenSLAM [15] and Radish [18]. OpenSLAM provides open-source SLAM algorithms and Radish offers laser range finder data sets. While we praise both initiatives, they are not as extensively used as they should and are missing benchmark tools that would be used to evaluate the quality of the SLAM algorithms (from a localization or a mapping standpoint) for specific applications.

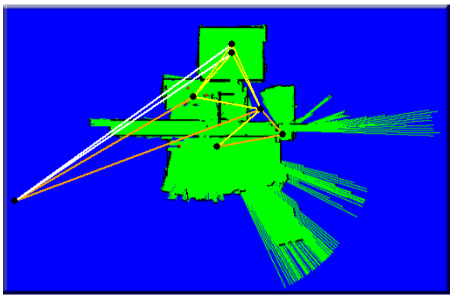


Figure 8: Example of a successful Creativity metric, displaying a communication network. Each transmitter is shown as a point with the lines showing the connections between each link. The point in the left represents the base station.

Throughout the years, we have devoted our map benchmarking endeavors to planar occupancy grid maps comprised of certainty values (i.e. either 0 for free space or 1 for occupied space). While this restriction has been reasonable over the last few years, mainly due to the popularity of occupancy grid maps, a surge of newly fashionable robotic platforms ranging from underwater robots to unmanned air vehicles coupled with highly three-dimensional terrain is slowly making two-dimensional occupancy grid maps inadequate. Indeed, teams have already started to explore three-dimensional mapping algorithms [10]. Evidently, the switch from two to three dimensional mapping is not straightforward in terms of map benchmarking and offers an interesting research avenue for future work. Furthermore, it is important to note that three-dimensional mapping does not have a map representation that is well recognized throughout the robotics community and that occupancy grids do not offer an easy transfer from two to three dimensions due to the increase of space and time complexities. We contend that more work needs to be achieved to come up with a community-accepted standard representation for three-dimensional maps.

All things considered, a general ”all-purpose” mapping benchmark is still far from being developed due to the aforementioned problems of map representation, algorithmic differences, lack of open-source data or algorithms, and application dependability. We are convinced that mapping benchmarks need to be tied to the application at hand and, as such, do not see a generalized map benchmark in the near-future. It is rewarding to see, however, that there is an increase in awareness as to the importance of the problem and hope that this paper will help steer map benchmarking towards the right direction.

Acknowledgments

An extended version of this paper appeared in [2].

6. REFERENCES

[1] M. Akbarzadeh, M. Khademi, M. Kahani, S. Haidarian, Y. Mohammadi, M. Mojtahedi, A. Taherinia, H. M, and H. Molla-Ahmadian. Robocup: Introducing the international robotic competition and teams at ferdowsi university of mashhad. Technical report, Ferdowsi University of Mashhad, 2005.

[2] B. Balaguer, S. Balakirsky, S. Carpin, and A. Visser. Evaluating maps produced by urban search and rescue robots: Lessons learned from robocup. *Autonomous Robots*, 2009.

[3] S. Balakirsky, S. Carpin, and A. Visser. Evaluating the robocup 2009 virtual robot rescue competition. In *Proceedings of PerMIS*, 2009.

[4] A. Birk. Learning geometric concepts with an evolutionary algorithm. In *The Fifth Annual Conference on Evolutionary Programming*, 1996.

[5] J. Carlson, R. Murphy, S. Christopher, and J. Casper. Conflict metric as a measure of sensing quality. In *IEEE International Conference on Robotics and Automation*, 2005.

[6] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. Bridging the gap between simulation and reality in urban search and rescue. In *Robocup 2006: Robot Soccer World Cup X*, 2006.

[7] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. USARSim: a robot simulator for research and education. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1400–1405, 2007.

[8] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, and J. Wang. Quantitative assessments of usarsim accuracy. In *Proceedings of PerMIS 2006*, 2006.

[9] M. I. Corporation. The mapinfo interchange file (mif) format specification. Technical report, MapInfo Corporation, 1999.

[10] P. de la Puente, A. Valero, and D. Rodriguez-Losada. 3D mapping: testing algorithms and discovering new ideas with USARSim. In *Proceedings of the IROS workshop Robots, Games, and Research: Success stories in USARSim*, 2009.

[11] P. Dempsey. Engineering football mini robocup. *Engineering and Technology Magazine*, 1(14):24–26, 2008.

[12] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature Extraction, Foundations and Applications*. Series Studies in Fuzziness and Soft Computing. Springer, 2006.

[13] M. Lewis, J. Wang, J. Manojlovich, S. Hughes, and X. Liu. Experiments with attitude: Attitude displays for teleoperation. In *Proceedings of the 2003 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1345–1349, 2003.

[14] M. C. Martin and H. P. Moravec. Robot Evidence Grids. Technical Report CMU-RI-TR-96-06, Robotics Institute - Carnegie Mellon University, March 1996.

[15] OpenSlam. http://www.openslam.org, 2009.

[16] S. O'Sullivan. An empirical evaluation of map building methodologies in mobile robotics using the feature prediction sonar noise filter and metric grip map benchmarking suite. Master's thesis, University of Limerick, 2003.

[17] S. Patterson. In this soccer match players are robotic but that's the goal. *The Wall Street Journal*, page A1, 2006.

[18] Radish – the robotics data set repository. http://radish.sourceforge.net, 2009.

[19] N. Ritter and M. Ruth. Geotiff format specification. Technical report, NASA Jet Propulsion Laboratory, 1995.

[20] E. Sklar, J. Johnson, and H. Lund. Children learning from team robotics: Robocup junior 2000. Technical report, The Open University, Milton Keynes, UK, 2000.

Evaluating Speech Translation Systems: Applying SCORE to TRANSTAC Technologies

Craig Schlenoff
National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
301-975-3456
craig.schlenoff@nist.gov

Brian Weiss
National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
301-975-4373
brian.weiss@nist.gov

Michelle Potts Steves
National Institute of Standards and Technology
100 Bureau Drive, Stop 8940
Gaithersburg, MD 20899
301-975-3537
michelle.steves@nist.gov

Greg Sanders
National Institute of Standards and Technology
100 Bureau Drive, Stop 8940
Gaithersburg, MD 20899
301-975-4451
greg.sanders@nist.gov

Fred Proctor
National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
301-975-3425
frederick.proctor@nist.gov

Ann Virts
National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
301-975-5068
ann.virts@nist.gov

ABSTRACT

The Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) program is a Defense Advanced Research Projects Agency (DARPA) advanced technology research and development program. The goal of the TRANSTAC program is to demonstrate capabilities to rapidly develop and field free-form, two-way translation systems that enable speakers of different languages to communicate with one another in real-world tactical situations without an interpreter.

The National Institute of Standards and Technology (NIST), along with support from MITRE and Appen Pty Ltd., have been funded to serve as the Independent Evaluation Team (IET) for the TRANSTAC Program. The IET is responsible for analyzing the performance of the TRANSTAC systems by designing and executing multiple TRANSTAC evaluations and analyzing the results of the evaluation.

To accomplish this, NIST has applied the SCORE (System, Component, and Operationally Relevant Evaluations) Framework. SCORE is a unified set of criteria and software tools for defining a performance evaluation approach for complex intelligent systems. It provides a comprehensive evaluation blueprint that assesses the technical performance of a system and its components through isolating variables as well as capturing end-user utility of the system in realistic use-case environments.

This document describes the TRANSTAC program and explains how the SCORE framework was applied to assess the technical and utility performance of the TRANSTAC systems.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
ACM 978-1-60558-747-9/09/09

Categories and Subject Descriptors

I.2.7 [**Computing Methodologies**]: Natural Language Processing – *machine translation, speech recognition and synthesis*

General Terms

Algorithms, Measurement, Performance, Experimentation, Human Factors, Languages

Keywords

Performance evaluation, speech-to-speech translation system, SCORE, TRANSTAC

1. INTRODUCTION[1]

Performance evaluation of advanced technologies can often be very challenging. It is the authors' belief that the design of an effective evaluation is as much a research issue as is the

[1] Certain commercial products and software are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the products and software identified are necessarily the best available for the purpose.

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

technology development itself. One must be able to accurately answer questions such as:

- Does the overall system do what it claims to do?
- What are the factors that would cause the overall system to fail?
- Is the system useful to the end-user (whether it be military, law enforcement, first responders, industry, etc.)?
- What are the key situations that the technology would be most useful for?
- How well do the individual components of the system perform and what is their impact on the performance of the overall system?
- How can we isolate specific capabilities of the system and test their performance?

In order to address this, the SCORE Framework (System, Component, and Operationally Relevant Evaluations) Framework was developed. SCORE is a unified set of criteria and software tools for defining a performance evaluation approach for complex intelligent systems. It provides a comprehensive evaluation blueprint that assesses the technical performance of a system and its components through isolating and changing variables as well as capturing end-user utility of the system in realistic use-case environments. [1]

The paper is organized as follows: Section 2 gives an overview of the TRANSTAC effort; Section 3 describes the SCORE framework; Section 4 describes how the SCORE Framework was applied to assess the TRANSTAC systems, Section 5 describes the metrics used in the TRANSTAC program, and Section 6 concludes the paper.

2. OVERVIEW OF THE DARPA TRANSTAC PROGRAM[2]

The Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) program is a Defense Advanced Research Projects Agency (DARPA) advanced technology research and development program. The goal of the TRANSTAC program is to demonstrate capabilities to rapidly develop and field free-form, two-way translation systems that enable speakers of different languages to communicate with one another in real-world tactical situations without an interpreter.

Several prototype systems have been developed under this program for numerous military applications including force protection and medical screening. The technology has been demonstrated on PDA (personal digital assistant) and laptop platforms. NIST was asked to assess the usability of the overall translation system and to individually assess each component of the system (the speech recognition, the machine translation, and the text-to-speech).

All of the TRANSTAC systems work fundamentally the same. Either English speech or an audio file is fed into the system. Automatic Speech Recognition (ASR) processes the speech to recognize what was said and generates a text file of the speech. That text file is then translated to another language using Machine Translation (MT) technology. The resulting text file is then spoken to the foreign language speaker using Text-To-Speech (TTS) technology. This same process then happens in reverse when the foreign language speaker speaks. This is shown in Figure 1.

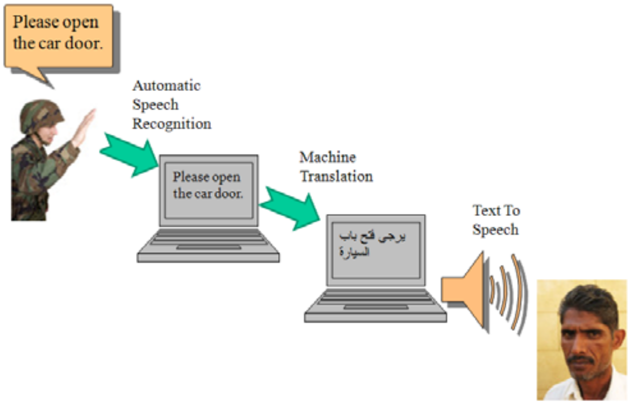

Please open the car door.
Automatic Speech Recognition
Please open the car door.
Machine Translation
يرجى فتح باب
السيارة
Text To Speech

Figure 1: How Speech Translation Works

3. OVERVIEW OF THE SCORE FRAMEWORK

The SCORE Framework [2] [3] has been developed at the National Institute of Standards and Technology (NIST) over the past three years to provide formative evaluations of advanced technologies that are still under development. SCORE is built around the premise that, in order to get a true picture of how a system performs in the field, it must be evaluated at the component level, the system level, the capability level and within operationally-relevant environments.

SCORE is unique in that:

- It is applicable to a wide range of technologies, from manufacturing to defense systems
- Elements of SCORE can be decoupled and customized based upon evaluation goals
- It has the ability to evaluate a technology at various stages of development, from conceptual to fully mature
- It combines the results of targeted evaluations to produce an extensive picture of a systems' capabilities and utility

To date, SCORE has been used to evaluate a wide range of advanced technologies, including Soldier-worn sensor systems, technologies allowing real-time multimedia information sharing among Soldiers in the field, two-way speech translation systems, and autonomous robotic platforms. It has been the foundation for ten technology evaluations involving Soldiers and Marines from

[2] Due to DARPA restrictions, the results of the evaluation cannot be published. Instead, this paper will focus on the evaluation approach as opposed to the results.

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

around the country. SCORE has been used as the basis of two DARPA programs to evaluate advanced technologies.

SCORE defines five evaluation goal types, as shown in Figure 2:

- *Component Level Testing – Technical Performance* – involves decomposing a system into components to isolate those subsystems that are critical to system operation.
- *Capability Level Testing – Technical Performance* – involves decomposing a system into capabilities (where the complete system is made up of multiple capabilities). A capability can be thought of as an individual functionality, such as the ability for a sensor system to send and receive a picture or the ability for a translation system to identify and translate names (discussed below).
- *Capability Level Testing – Utility Assessments* –assesses the utility of an individual capability. The benefit of this evaluation type is that specific capability utility and usability to the end-user can still be addressed even when the system and user-interface are still under development.
- *System Level Testing – Technical Performance* –assesses the system as a whole, but in an ideal environment where test variables can be isolated and controlled. The benefit is that tests can be performed using a combination of test variables and parameters, where relationships can be determined between system behavior and these variables and parameters based upon the technical performance analysis.
- *System Level Testing – Utility Assessments* –assesses a system's utility, where utility is defined as the value the application provides to the system's end-user. In addition, usability is assessed. which includes effectiveness, learnability, flexibility, and user attitude towards the system.

Considering each of these evaluation elements, SCORE takes a tiered approach to measuring the performance of intelligent systems. At the lowest level, SCORE uses elemental tests to isolate specific components and then systematically modifies variables that could affect the performance of those components to determine those variables' impact. Typically, this is performed for each relevant component with the system. At the next level, the overall system is tested in a highly structured environment to understand the performance of individual variables on the system as a whole. Then, individual capabilities of the system are isolated and tested for both their technical performance and their utility using task tests. Lastly, the technology is immersed in a longer scenario that evokes typical situations and surroundings in which the end-user is asked to perform an overall mission or procedure in a highly-relevant environment which stresses the overall system's capabilities. Formal surveys and semi-structured interviews are used to assess the usefulness of the technology to the end-user.

4. APPLYING SCORE TO TRANSTAC

Technical performance of the individual components of the TRANSTAC system was performed using offline tests (represented by the red arrow in Figure 3). Both technical and utility performance of the entire system was performed using lab-based evaluations of a laptop-based system (represented by the gray arrows in Figure 3) and more field-friendly utility systems (represented by the green arrows in Figure 3). Utility evaluations were also performed out in the field with the field-friendly systems (represented by the blue arrow in Figure 3). Lastly, the specific capabilities of the TRANSTAC systems (such as their ability to recognize proper names) were tested both for their technical capability and their utility (represented by the purple arrows in Figure 3). Each of these tests is discussed in detail below.

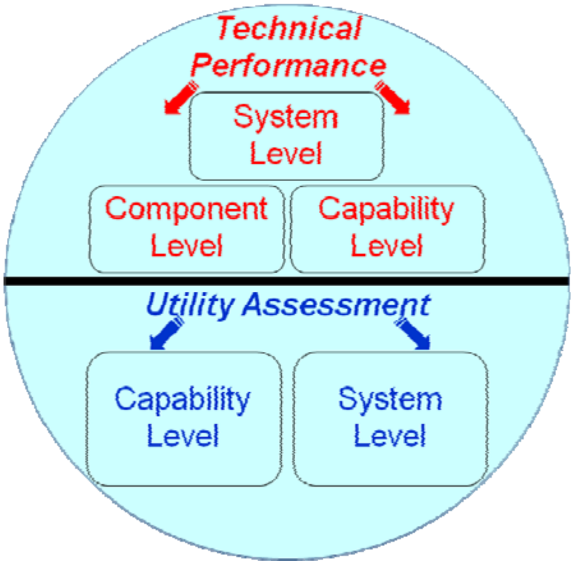


Figure 2: SCORE Architecture

4.1 Offline Evaluations

The offline evaluation was performed to assess the technical performance of the TRANSTAC systems at the component level. There were three primary components that were being tested: the Automated Speech Recognition, the Machine Translation, and the Text-to-Speech. The offline evaluation was performed so that the component evaluation would be conducted on identical inputs for all systems. In advance of the evaluation, research teams were provided with the required log formats for storing the results of the offline processing. They were also provided with sample offline data that could be used to develop logging scripts and produce sample outputs. A verification script was provided to check the output for log format errors.

During the offline evaluation, research teams provided the same versions of their systems that were used for the live evaluation. Research teams were provided with audio files for speech recognition and subsequent translation. Separately, they were provided with transcription files for text translation.

Each system processed approximately 1000 audio files of utterances in each language and stored the results in system logs. In the context of this paper, an utterance is the words spoken by a

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

human from the time s/he starts speaking to the time that the TRANSTAC system begins to translate. An utterance can contain one or many concepts (individual pieces of information), but efforts have been made to have a comparable average number of concepts among all offline utterances from one evaluation to the next.

For each audio file, the system stored the results of ASR, the translation based on ASR output, and time stamps marking the beginning and end of each process (recognition, translation, and TTS, if used). Outputs from the transcription inputs were the same except that results related to speech recognition were left blank. When processing was complete, a verification script was run on the logs to ensure that the output conformed to the required format. Logs were also checked for the correct number of outputs.

In addition to the above, thirty well-formed foreign language text strings were fed into the TTS engine of the TRANSTAC systems. These engines read in the text strings and output audio files which contained the spoken version of the text.

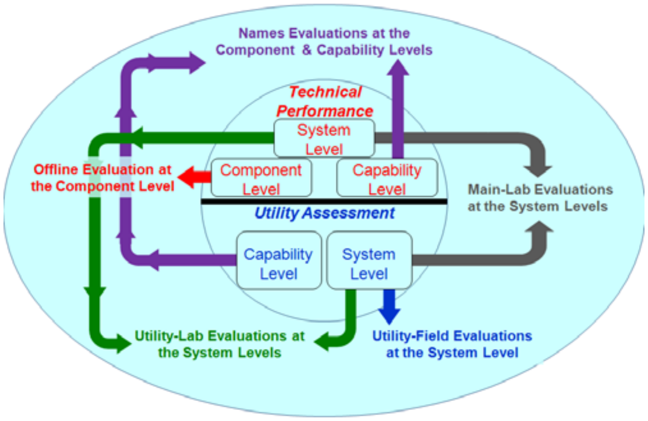


Figure 3: SCORE Applied to TRANSTAC

Analysis on the offline evaluation focused on component level performance of the TRANSTAC systems using automated metrics and human judgments. The following metrics were used to analyze the offline data:

- Human Judgment
  - Low-level concept transfer, performed by bilingual human judges
  - Likert judgment [4] at utterance level, performed by bilingual human judges
  - Likert judgment performed by bilingual human judges, to assess TTS
- Automated Metrics
  - Word Error Rate (WER) to assess ASR and TTS
  - METEOR, BLEU – to assess ASR and MT together

More details about these metrics can be found in Section 5.

4.2 Lab-Based Evaluations

The main difference between the offline evaluation described in Section 4.1and the live lab and field evaluations described in Sections 4.2 and 4.3 is that the live evaluations allow speakers to generate their own utterances of inquiries and responses while the offline evaluations uses scripted, recorded utterances by both speakers to provide an apples-to-apples comparison.

Figure 4 shows an example of one of the teams' TRANSTAC system. The main processing unit is a standard laptop, in which a head-mounted microphone (top right) and a speaker (bottom right) are plugged in. A hand-held control (bottom left) is as plugged into the laptop which allows the Soldier/Marine to let the system know when each speaker is about to talk. Each 'START' button corresponds to a different speaker and the button on the bottom allows the Soldier/Marine to replay the last audio that was output from the TRANSTAC system.

Lab-based evaluations were used to assess the technical capability and utility of the TRANSTAC systems at the systems level. Approximately twenty scenarios are used to assess the performance of the TRANSTAC systems in a lab setting. These scenarios have either been structured scenarios or spontaneous scenarios. Structured scenarios provide a set of questions to the English speaker that they needed to find answers to. The foreign language speaker was given the answers to those questions in paragraph format. A dialogue occurred between the two speakers and the number of answers that the English speaker was able to obtain was noted.

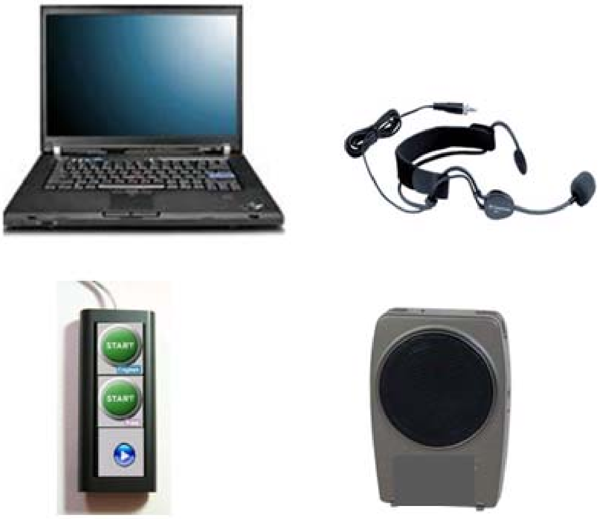


Figure 4: Example of Laptop-Based TRANSTAC System

For spontaneous scenarios, a brief paragraph was provided to the English and foreign language speaker to give them the proper background to carry on a meaningful conversation. The background could state that they were performing a census survey and were going house to house gathering information about peoples' living conditions. The direction that the Soldier/Marine takes the conversation was up to them, as long as it is within the bounds of the scenario description. There are advantages and disadvantages to both types of scenario, which is outside the scope of this paper. However, in both cases, the goal was to measure the number of meaningful interactions that the Soldier/Marine and the foreign language speaker has in a finite amount of time.

In addition, after the interaction, questionnaires were provided to the English and foreign language speakers to gauge their perception of the TRANSTAC systems.

All scenarios were performed in an indoor environment, usually in a conference room of a hotel. The Soldier/Marine and the foreign language speakers were stationary, with the TRANSTAC system on the table between them. All lab scenario runs were performed in this environment, with each scenario occurring within a ten minutes period. Noise masking technology was deployed to stop the speakers from hearing each other. They could only respond to what came out the TRANSTAC system. The goal of this type of evaluation is to place the systems in what many would consider an ideal environment (no background noise, minimal movement, etc.) to get an upper bound on how well they could perform.

Because there were two physical systems (a laptop version and a more field-friendly) we used the same lab-based evaluation procedures for both systems.

For the lab-based evaluations, the following metrics were used to analyze the data:

- A count of high-level concepts found out by the Soldier/Marine in response to the questions he asked.
- Analysis of the questionnaire performed by Soldiers/Marines and foreign language speakers after each scenario in which they participated.

More details about these metrics can be found in Section 5.

4.3 Field-Based Evaluations

The field-based evaluations were used to assess the utility of the TRANSTAC systems at the system level. The field scenarios were performed outdoors with Soldiers/Marines wearing combat gear (body armor, helmet, gloves, etc.). They carried a "utility version" of the TRANSTAC systems while performing the scenarios. Following the scenarios, the Soldiers/Marines filled out questionnaires and participated in interview sessions with the evaluation team.

The field environments were not intended to be completely representative of what the Soldiers/Marines would experience overseas. To replicate this type of environment would be a very difficult undertaking and it would not tell us much more than a more simplistic environment would. The reason for performing field evaluations was to subject the systems to the type of environmental variable that they would realistically be exposed to, such as wind, background noise, and the motion caused by the Soldier/Marine carrying the systems around with them. It also allowed the user to see how easy the system was to use while carrying around other gear such as bullet-proof vests and weapons.

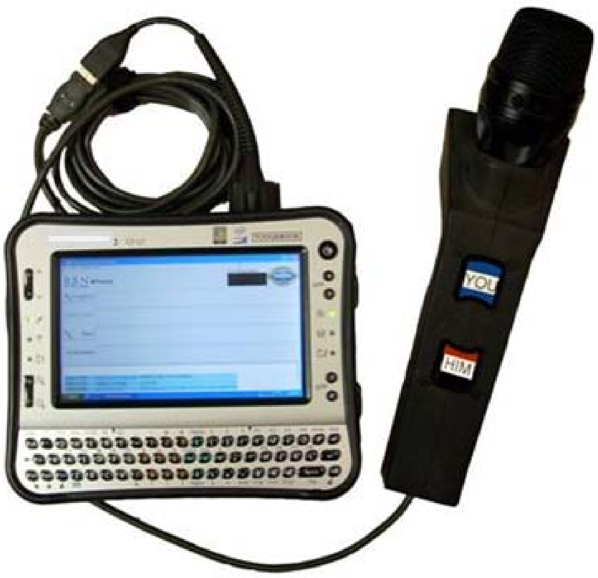


Figure 5: Example TRANSTAC Utility System

An example of a utility version of the TRANSTAC system is shown in Figure 5. The "YOU" button on the microphone was meant to be push when the Soldier/Marnie was speaking (since they are the controller of the systems) and the "HIM" button was meant to be pushed when the foreign language speaker was speaking. A sample field environment that was used for testing is shown in Figure 6.

For the field-based evaluations, the following metrics were used to analyze the data:

- Analysis of the questionnaire performed by Soldiers/Marines and foreign language speakers after each scenario in which they participated.
- Semi-structured interviews with the Marine/Soldiers and foreign language speakers.

More details about these metrics can be found in Section 5.

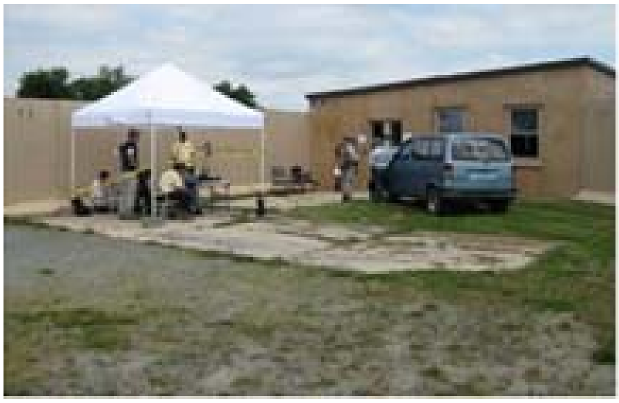


Figure 6: Sample Field Environment

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

4.4 Proper Names Evaluations

The proper names evaluation was an example of a capability evaluation used to assess specific functionalities of the TRANSTAC system. The goal of the capability evaluation was to isolate specific functionalities of a system and test its performance with scenarios that are tailored to stress that functionality. The evaluation team focused on the ability for the TRANSTAC system to identify and convey proper names in a dialogue. In this context, proper names were people names, street names, and city names that were being conveyed from the foreign language speaker to the English speaker. Three unique, names-laden scenarios were created as scripted dialogues and recorded by unique speakers. Each scenario was very rich in proper names; they typically contained approximately 50 to 55 proper names within the 30 to 40 foreign language utterances. This recorded data was used to create the offline names evaluation.

The offline names evaluation was run similar to that of the other offline evaluations. Specific recorded utterances were selected and fed directly into the TRANSTAC systems. However, the metrics from this test focus on how the systems specifically handle the translations of the proper names, as discussed below.

The live names evaluation was run in a different manner than that of the live lab evaluation. The speakers were provided with the scripted names scenarios and instructed to read them verbatim into the TRANSTAC system. After hearing TRANSTAC translation of the English utterance, the foreign language speaker responded with their corresponding scripted utterance which again was spoken into the TRANSTAC system. That foreign language utterance was then translated into English. If the English speaker was able to understand the name that was translated/conveyed by the TRANSTAC system, they noted that and moved on to the next utterance. If the English speaker was unable to ascertain a name from the TRANSTAC output, then they were able to rephrase their original English utterance in any manner they saw fit. Likewise, the foreign language speaker, upon hearing the TRANSTAC output once the English speaker rephrased their utterance, could rephrase theirs accordingly to convey the desired name. The output of this evaluation produced both technical performance and utility assessment data. This took the form of measuring the number of names successfully transferred per unit time and collecting survey responses from the end-users regarding their specific names interactions. There were three names scenarios that were performed during the evaluation.

To evaluate the live and offline names evaluation, each TRANSTAC output was analyzed to see how well the proper name was translated from the foreign language to English. This was performed by a panel of human judges. A score was provided to each output which classified each name translation as either:

- Right name, right pronunciation
- Right name, wrong pronunciation
- Name translated as word (these were the cases where a proper name can also have a separate meaning… Black could be a person’s last name or a color)
- Wrong name translation
- Name not recognized

5. METRICS APPLIED TO TRANSTAC

In order to get a comprehensive picture of the performance of the TRANSTAC system, a large number of performance metrics were used when evaluating the systems. Many of these metrics are described below. The TRANSTAC community is in agreement that the two aspects that best characterize the performance of the systems are: (1) the semantic adequacy of the translations, leading to justified user confidence in the system’s translations, and (2) the ability of Marines/Soldiers and foreign language speakers to successfully carry out a task-oriented dialogue in a narrowly focused domain of known operational need under conditions that reasonably simulate use in the field. The metrics that were use to assess these capabilities are:

1. **High-Level Concept Transfer:** Semantic adequacy of the translations was assessed by bilingual judges telling us *whether* the meaning of each utterance came across. The high-level concept metric is the number of utterances that are judged to have succeeded. Thus, failed utterances are not directly scored (other than taking up time). The high-level concept metric is an efficiency metric which shows the number of successful utterances per unit of time, as well as accuracy. This metric is roughly quantitative.

2. **Likert Judgment:** A judgment of the semantic adequacy of the translations was performed by having a panel of bilingual judges rate the semantic adequacy of the translations, an utterance at a time. We asked our panel of five bilingual judges to assign a Likert-type score to each utterance, choosing from a seven-point scale.

+3	Completely_adequate
+2	
+1	Tending_adequate
0	
–1	Tending_inadequate
–2	
–3	Inadequate.

   The judges were provided with a substantial set of exemplars showing utterances which were deemed to correspond to the four values (completely adequate, tending adequate, tending inadequate, inadequate) and were asked to choose the in-between values only if on the fence between two of those values.

3. **Low-Level Concept Transfer:** A directly quantitative measure of the transfer of the low-level elements of meaning in each utterance. In this context, a low-level concept is a specific content word (or words) in an utterance. For example, the phrase “The house is down the street from the mosque.” is one high-level concept, but is made up of three low-level concepts (house, down the street, mosque).

   We had an analyst who is a native speaker of each source language identify the low-level elements of meaning (low-level concepts) in representative sets of input utterances from

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

the offline datasets and then asked a panel of five bilingual judges to tell us which low-level concepts were successfully transferred into the target-language output (where failures are deletions, substitutions, or insertions of concepts). Progress from one evaluation to the next may be presented as an odds ratio. Odds of successful concept transfer is a more quantitative measure of translation adequacy than the Likert-type judgments of semantic adequacy — the Likert-type judgments give the bilingual judges the opportunity to take into account the relative importance of the various concepts while the low-level concept transfer does not. [4]

4. **Automated Metrics:** A suite of automated metrics, intended to enable the research team to better understand what aspects of performance account for the end-to-end success of their systems. We hope to identify automated metrics that can be run quickly and easily yet will correlate strongly with judgments of semantic adequacy provided by bilingual judges. For speech recognition, we calculated Word-Error-Rate (WER) — using SCTK version 2.2.2 and automated procedures for normalizing the hypothesis and reference texts. For machine translation, we calculated BLEU [5] using four reference translations. We also measured MT performance by calculating a metric called METEOR defined by Alon Lavie of CMU. For both English and Dari, METEOR was run in the mode where it scores only exact matches (no stemming or synonymy). [6]
5. **TTS Evaluation:** To assess the performance of a TTS component, human judges listened to the audio outputs of the TTS evaluation and compared them to the text string of what was fed into the TTS engine. They then gave a Likert score from 1-5 (five being the best) to indicate how understandable the audio file was in comparison to what was fed into it. In addition, these human judges transcribed what they heard in the audio file in the foreign language and then these transcriptions were compared to the input text files using Word Error Rate.
6. **Surveys/Semi-Structured Interviews:** After each live scenario, the Soldiers/Marines and the foreign language speakers filled out a detailed survey asking them about their experiences with the TRANSTAC systems. The surveys explored how easy the system was to use, how well they perceived it worked, and errors that the users encountered when interacting with the system. In addition, after the field scenarios, semi-structured interviews were performed with all of the participants in which questions such as “What did you like?, What didn’t you like? and What would you change?” were explored.

6. METRICS COMPARISON

Although I cannot discuss detailed results in this paper due to DARPA restrictions, I can discuss, at a meta-level the level of consistency that was found by applying these metrics to the teams’ TRANSTAC output. For the purpose of this comparison, I will show the rank ordering of the teams’ performance by applying the follow metrics described in Section 5: high-level concept transfer, low-level concept transfer, Likert judgment, BLEU, and METEOR. This is shown in Table 1.

Table 1: Metrics Comparison

LANGUAGE DIRECTION	Metric	Team 1	Team 2	Team 3
Dari to English	High Level Concept Transfer	1	2	3
Dari to English	Low-level Concept Transfer	1	2	2
Dari to English	Likert Judgment	1	2	2
Dari to English	BLEU	1	2	2
Dari to English	METEOR	1	2	2
English to Dari	High Level Concept Transfer	1	2	3
English to Dari	Low-level Concept Transfer	1	2	2
English to Dari	Likert Judgment	1	1	1
English to Dari	BLEU	1	2	2
English to Dari	METEOR	1	1	1

As shown in Table 1, the numbers under Team 1, Team 2, and Team 3 show their relative score compared to each other teams when applying the metrics in the second column. For example, Team 1 had the highest relative score applying the high-level concept transfer metric looking at the translation from Dari to English. Team 2 had the second highest score and Team 3 had the third highest score. When two teams have the same number in the same row, it means that the scores were not statistically significant enough to be able to say that one score was better than the other. For example, Team 2 and Team 3 have very comparable scores when applying the low-level concept transfer metric in the Dari to English direction; hence they are both listed as the second ranked team.

The table shows that there is significant comparability in the overall results when applying different metrics. In the Dari to English direction, Team 1 consistently was ranked #1 in all of the metrics applied and Team #2 was consistently ranked #2. The only difference was that there was a statistical difference between Teams 2 and Team 3 when applying the low-level concept transfer metric, where there was not a statistical difference when applying the other metrics.

When looking at the English to Dari direction in Table 1, Team 1 came out with the highest relative rank in all five metrics again. However, Teams 2 and Team 3’s scores varied depending which metrics was applied. Looking at Team #3, it was ranked third when applying the high-level concept transfer metric but was tied for first when applying the Likert judgment and METEOR metrics. In situations like this, one usually defaults to the metrics which involves humans, which is sometimes referred to as ground truth or the gold standard. The first three metrics (high-level concept transfer, low-level concept transfer, and Likert) all involved human judges. Unfortunately, this still doesn’t provide much insight as Team 3 is ranked #3, #2, and #1, respectively. As

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

such the only conclusion we can draw from this is that Team #1 appears to be superior overall, while Team #2 and Team #3 are roughly tied for second.

7. CONCLUSION

In this paper, we have discussed the SCORE Framework and shown how it was applied to the DARPA TRANSTAC program. Using SCORE, we were able to evaluate the performance of speech translation systems by looking at the performance of:

- the systems at the component level using offline evaluations,
- the performance of the overall system in ideal environments using lab evaluations,
- the performance of the system in operationally-relevant environments using field test, and
- the specific capabilities of the systems to evaluate proper names.

By putting together the results of all of these evaluations, we are able to gain a much more comprehensive evaluation of an overall system performance.

SCORE has proven to be an invaluable evaluation design tool for the NIST Evaluation Team and was the backbone of eleven DARPA evaluations: six for the DARPA ASSIST program (not discussed in this paper) and five for TRANSTAC program. It is expected to play a critical role in the remaining ASSIST and TRANSTAC evaluations.

The SCORE framework is applicable to domains beyond emerging military technologies and those solely dealing with intelligent systems. Personnel at NIST are applying the SCORE framework to the virtual manufacturing automation competition (VMAC) [7] and the virtual RoboRescue competition [8] (within the domain of urban search and rescue). Their intent is to develop elemental tests and vignette scenarios to test complex system capabilities and their component functions. The framework has proven to be highly adaptable and capable of meeting most any evaluation requirement.

ACKNOWLEDGMENTS

The authors would like to acknowledge the DARPA TRANSTAC program manager, Dr. Mari Maeda, and the members of the NIST IET for their continued support.

REFERENCES

[1] B. Weiss, C. Schlenoff, G. Sanders, M. Steves, S. Condon, J. Phillips, and D. Parvaz, "Performance Evaluation of Speech Translation Systems," in *Proceedings of the LREC 2008 Conference* Morocco: 2008.

[2] C. Schlenoff, M. Steves, B. Weiss, M. Shneier, and A. Virts, "Applying SCORE to Field-Based Performance Evaluations of Soldier Worn Sensor Technologies," *Journal of Field Robotics*, vol. 24, no. 8/9, pp. 671-698, Sept.2007.

[3] B. Weiss and C. Schlenoff, "Evolution of the SCORE Framework to Enhance Field-Based Performance Evaluations of Emerging Technologies," in *Proceedings of the 2008 Performance Metrics for Intelligent Systems (PerMIS) Conference* Gaithersburg, MD: 2008.

[4] G. Sanders, S. Bronsart, S. Condon, and C. Schlenoff, "Odds of Successful Transfer of Low-Level Concepts: A Key Metric for Bidirectional Speech-to-Speech Machine Translation in DARPA's TRANSTAC Program," in *Proceedings of the LREC 2008 Conference* Morocco: 2008.

[5] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* Philadelphia, PA: 2002, pp. 311-318.

[6] S. Condon, J. Phillips, C. Doran, J. Aberdeen, D. Parvaz, B. Oshika, G. Sanders, and C. Schlenoff, "Applying Automated Metrics to Speech Translation Dialogs," in *Proceedings of the LREC 2008 Conference* Morocco: 2008.

[7] S. Balakirsky and R. Madhavan, "Advancing Manufacturing Research Through Competitions," in *Proceedings of the SPIE Defense Security and Sensing* Orlando, FL: 2009.

[8] S. Balakirsky, C. Scrapper, and S. Carpin, "The Evolution of Performance Metrics in the RoboCup Rescue Virtual Robot Competition," in *Proceedings of the Performance Metrics for Intelligent Systems Workshop* Gaithersburg, MD: 2007.

The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

Development and Internal Evaluation of Speech-to-Speech Translation Technology at BBN

David Stallard
BBN Technologies
10 Moulton St.
Cambridge, MA
(617) 873-2825
stallard@bbn.com

Rohit Prasad
BBN Technologies
10 Moulton St.
Cambridge, MA
617-873-4785
rprasad@bbn.com

Prem Natarajan
BBN Technologies
10 Moulton St.
Cambridge, MA
617-873-5472
prem@bbn.com

ABSTRACT

Speech-to-speech translation (S2S) technology holds out the promise of allowing spoken communication across language barriers. Recently, there has been a great deal of progress in S2S technology, much of it under the sponsorship of DARPA's TransTac program. In this paper, we present BBN's S2S system, "TransTalk", whose development has been funded under the TransTac program. We also present various performance metrics, and the result of detailed performance analyses, using the TransTalk system.

Categories and Subject Descriptors

I.2.7 **[Artificial Intelligence]**: Natural Language Processing – *machine translation, speech recognition and synthesis*.

General Terms

Measurement, Performance, Experimentation, Human Factors.

Keywords

Speech-to-speech translation, evaluation, performance analysis.

1. INTRODUCTION

Speech-to-speech translation (S2S) technology holds out the promise of allowing spoken communication across language barriers. Using a suitably configured S2S system, two people who do not share a common language can communicate with one another by speaking through the system. The system translates the speech of one party into synthesized speech in the language of the other party, and vice versa. S2S systems combine multiple technologies, including Automatic Speech Recognition (ASR), Machine Translation (MT), and Text-to-Speech synthesis (TTS).

Recently, there has been significant progress in S2S technology. Much of this progress has taken place under the auspices of DARPA IPTO's TransTac program (Dr. Mari Maeda, Program Manager). "TransTac" stands for "Translation systems for Tactical use". The TransTac program sponsors S2S work between English and several foreign languages of interest to the US government, including Iraqi Arabic, Malay, Farsi, Dari, and Pashto. Systems have been developed for Iraqi Arabic, Malay, Farsi, and Dari. As of this writing, work is also underway for Pashto, one of the main languages of Afghanistan. These systems support conversations in the "force protection" domain, which is broadly construed to include not only checkpoints, searches, and military operations, but also rapport building, civil affairs, and basic medical. In addition to direct research funding, the Transtac program also funds data collection and periodic competitive evaluations, using live military users. Participating research sites are BBN, IBM, and SRI, with additional research contributions from USC and CMU.

A key enabler of recent progress in S2S has been the recent and ongoing progress in Statistical Machine Translation (SMT). SMT uses statistical, corpus-driven approaches, as opposed to hand-written rules, and automated performance evaluation, as opposed to manual performance evaluation. The resulting labor savings has greatly accelerated the progress of MT. Moreover, the statistical paradigm generally provides better performance than hand-written rules, which are often brittle and in conflict with one another. In this way, SMT may be seen as following in the footsteps of ASR, which also underwent dramatic improvement following the adoption of statistical paradigms and automated evaluation.

It should be noted, however, that S2S systems in their present form are not "universal translators", ala Star Trek. On the contrary, they must be configured for particular language pairs by training ASR and translation models using speech and language data (recordings, transcriptions, and translations) in the relevant languages. While existing data can sometimes be found on the Internet or elsewhere, it is more frequently the case that the data has to be collected. Moreover, for optimal performance, the data collected should match the intended domains of conversation of the system. That is, if the intended domain is force protection, data should be collected for that domain. While data outside that domain can be helpful for general modeling of the given language, it often lacks the key concepts and constructions that are important in the specific application domain. In practice, the domain-relevant data is frequently conducted in simulated translingual dialogs between people playing the roles of soldier and civilian.

In the remainder of this paper, we give an overview of BBN's TransTac S2S system, "TransTalk" [1], and its component technologies. We follow this with a description of the automated metrics used in improving our system. Finally, we give results of some performance analyses of our system.

2. SYSTEM OVERVIEW

A block diagram of BBN's TransTac system, "TransTalk" [1], is system is shown in Figure 1. The BBN TransTalk system uses

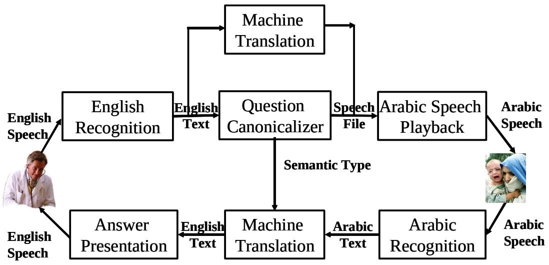


Figure 1: Block Diagram

BBN's Byblos speech recognizer [2], BBN's SMT engine, and Cepstral's text-to-speech synthesizer. Various input modalities are supported, including both handheld and headset microphones. The primary physical interface is the "SuperMic", a handheld unit developed by BBN, which encompasses a directional microphone, speakers, and two push-and-hold "listen" buttons, one for receiving the SME's English speech, and the other for receiving the FLE's foreign speech. Figure 2 shows the BBN TransTalk system running on an Ultra Mobile PC in the SuperMic configuration.

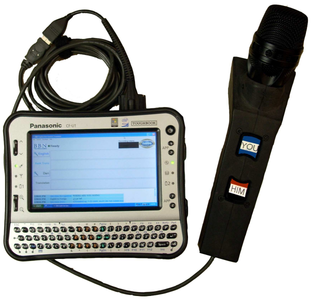


Figure 2: BBN TransTalk System with the BBN SuperMic

English speech received through the physical interface device is sent to the English speech recognizer, which outputs the string of words in the recognizer's vocabulary which it judges to be the most likely given the sounds received and given the recognizer's Language Model (LM). This English text is then sent to both the SMT component and to a separate Question Canonicalizer component. The Question Canonicalizer tries to match the utterance to one of a set of utterances for which it has stored translations. If none is found, the output of the SMT component is used instead. Arabic speech corresponding to the translation is then played back. This may be a pre-recorded wave file, or more generally, the result of text-to-speech synthesis.

The foreign language speaker's reply ("Arabic" in Figure 1) is sent to the foreign language recognizer, which outputs text in the foreign language. This text is then sent to a second speech synthesizer, which speaks it out for the English speaker to here, and/or displaying it on a screen.

A key requirement of TransTac systems is that they be "displayless", aka "eyes-free"; i.e. that the user does not have to look at or interact with a screen display in order to use the system. Because of the requirement of displayless operation, our system uses audio confirmation. In particular, it uses implicit confirmation, in which the system speaks the confirmation utterance for the SME to hear. If the SME judges this correct, he takes no action, and the translation is played out for the FLE to hear. If the SME is not satisfied with what he hears, he simply presses the English "listen" button to barge in, interrupting system output and allowing him to and re-speak.

In the following, we briefly discuss the component technologies involved. The BBN Byblos speech recognizer [2] models speech as the output of context-dependent phonetic Hidden Markov Models (HMMs). The outputs of the HMM states are mixtures of multi-dimensional diagonal Gaussians. Different forms of parameter tying are used in Byblos, including State Tied Mixture (STM) triphone and State Clustered Tied Mixture (SCTM) quinphone models. The mixture weights in both these cases are shared based on the decision tree clustering

Speech recognition is performed using our patented two pass search strategy [3]. The forward pass is a fast-match beam search using an STM acoustic model and an approximate bigram language model. The output of the forward pass consists of the most likely word-ends per frame along with their partial forward likelihood scores. The backward pass operates on the set of choices from the forward pass to restrict the search space, and uses the more detailed SCTM quinphone model and a trigram language model to produce the best hypothesis.

BBN's Statistical Machine Translation (SMT) engine is a phrase-based translation system based [4]. Given, an input foreign language sentence 'f', we estimate the most likely translation into the target sentence 'e' as:

$$\hat{e} = \arg \max_{e} \; P(e \mid f)$$

Word alignments between source-target sentence pairs are generated using GIZA++ based on IBM's Models 1 to 4 [5]. In order to improve the quality of the alignments, word alignments in the forward and backward direction are merged as in [6]. Phrase pairs are automatically extracted from the word alignments by merging neighboring alignment groups using a set of rules. The decoder uses a log-linear model of different features to choose between competing translation hypotheses. The parameters of the model are estimated using statistics of the phrase pairs extracted from the word alignments. The interpolation weights are optimized by minimizing the translation errors on a held out development set.

3. EVALUATION METRICS

In our work, we make frequent use of automated metrics to evaluate experimental configurations of our system, so as to determine whether or not to adopt a given new technique in modeling, data normalization, etc.. The advantage of using automated metrics is that it enables one to efficiently experiment with many different configurations, and to choose the best one according to the metrics. A fairly large "validation set" consisting

Conference'04, Month 1–2, 2004, City, State, Country.

of 11K utterances, is used to test the system performance. For more occasional subjective evaluations, we use the familiar 1 -5 Likert scale.

For ASR development, we of course use the standard Word Error Rate (WER) metric, which is based on edit distance. WER measures the number of insertions, deletions, and substitutions that would be required to transform the correct, or “reference”, transcription into the machine-generated one. For MT development, we use a combination of several automated metrics, including the widely-used BLEU [7] and METEOR [8]. BLEU measures n-gram precision, and also includes a brevity penalty. It tends to reward more fluent translations, but has no notion of word equivalence. METEOR by contrast measures primarily unigram recall, but uses WordNet [9], which recognizes synonymy and stemming equivalences between different words and different forms of the same word. METEOR tends to reward translation adequacy.

We also make use of various metrics developed at BBN. One is the Translation Error Rate (TER) [10]. TER, like WER, is a measure based on edit distance. It differs from WER, however, in that it allows shifts of arbitrary sub-strings, so as to uncover matches that would otherwise be treated as substitution errors. Shifts are counted as an equally weighted component of error rate, along with insertions, deletions, and substitutions. Performing a shift can reduce the overall edit-distance error rate; however, it avoids the multiple substitution and insertion errors that would otherwise be incurred.

A second metric developed at BBN is Semantic Translation Error Rate (STER) [11]. STER is based on TER, but adds the synonymy and stemming capabilities used by METEOR, in order to compute matches for alignment. STER also differs from TER in that it forbids the alignment, even as a substitution, of a concept and a non-concept word. (Non-concept words are function words like “the”, “is”, “of”, etc.) As a result, STER can produce alignments which are linguistically more intuitive than TER. An example is shown in Figure 3.

```
Best Ref: the house is smoking
Orig Hyp: smoke is came from the home
REF :  the  HOUSE is **** ****   SMOKING
TER : [the] SMOKE is CAME FROM @ HOME
REF :   **** **** the house is   smoking
STER:@@ CAME FROM the home [is] [smoke  ]
```

Figure 3: STER Alignment Example

4. STER and Human Judgment

In this section, we evaluate STER in terms of its correlation with human judgments. The corpus used in all the experiments is the offline evaluation set used in the March 2006 TransTac evaluations. This set consists of 1440 spoken Iraqi Arabic utterances spanning four different domains: general survey, intelligence, medical, and municipal services. Each utterance was transcribed in Arabic, and given four reference translations into English.

All SMT experiments provide two sets of scores. One set of scores evaluates translation performance on the reference transcriptions of the utterances (T2T). Another set of scores evaluates translation performance on speech recognition output as the source (S2T).

In the first experiment, we compare the correlation of the various metrics with human judgment. A judge who was a native speaker of Arabic and fluent in English assigned 1-5 Likert scores to each translation output as a rating of their quality. In Table 1, we show the Pearson correlation coefficient, R, of Likert scores for every utterance against TER, METEOR, and STER scores respectively.

From Table 1, we see that the STER metric is better correlated to human judgment than TER. Since TER and STER metrics have different edit costs for edits involving stop words, we performed another experiment to ensure the improved correlation results from the quality of the alignment and not due to edit costs.

Metric	**R(T2T)**	**R(S2T)**
TER	0.4450	0.5536
STER	0.4827	0.6077
METEOR	**0.5342**	**0.6295**

Table 1: Comparison of Pearson correlation coefficient computed w.r.t Likert scores across different metrics.

In Table 2, we provide correlation scores for STER and TER when the stop words have been removed from the hypothesis and reference. Since the edit costs for both the metrics are identical, the improvement in correlation reflects the improvement in the quality of the word alignments.

Metric	**R(T2T)**	**R(S2T)**
TER	0.4550	0.5661
STER	**0.4662**	**0.5875**

Table 2. Comparison of TER and STER after removing stop words

The TER scores in Table 1 and Table 2 show that the removal of stop words results in a slight improvement in the correlation coefficient. However, removing stop words reduces correlation with human judgment for STER. Given STER aligns stop words independently of concept words, the reduction in correlation shows that human judgment is sensitive to non-concept words too. We believe this is due to the fact that stop words positively correlate with human judgment when it can be aligned with other stop words (as in STER) but negatively correlates with human judgment when it can align with both stop words and concept words (as in TER).

Based on the results in Table 1, we can conclude that METEOR correlates best with human judgment. In Table 3, we compare METEOR to SMET, a metric derived from STER alignments as described above. The SMET instead of using the METOR alignments uses the STER alignment for computing the METEOR-equivalent score. As shown in Table 6, SMET has similar score as METEOR and is equally well correlated with human judgment. These results highlight the utility of SMET as a metric which is well correlated to human judgments and at the

Metric	**T2T**		**S2T**	
	R	Score	R	Score
METEOR	**0.5342**	0.6540	**0.6295**	0.5430
SMET	0.5331	**0.6556**	0.6270	**0.5462**

Table 3: Comparison of METEOR and SMET metrics

same time provides useful alignments for human driven analysis of the system output.

An interesting possible consequence of the results in Table 3 is that scores equivalent to METEOR can be computed with a simpler algorithm than METEOR itself uses. We performed a detailed comparison of the individual unigram alignments for concept words in STER and METEOR, and found that only 0.5% of them were different. Notably, SMET does not require METEOR's multiple stages of unigram matching for different word equivalency measures (WordNet, Porter stemming, etc). The STER alignment can also be viewed as enforcing the constraints that METEOR enforces on unigram alignments, namely making them one-to-one, and minimizing alignment crossings.

5. PERFORMANCE ANALYSES

In this section, we present two detailed performance analyses for our system. In the first, we seek to determine which types of machine translation error are the most harmful to performance. In the second, we evaluate the effectiveness of the so-called "back-translation" strategy for user confirmation.

5.1 Quantifying Damage Caused by Errors

The MT component of an S2S system can make many different types of errors, both major (dropping concepts, using the wrong word sense, etc) and minor (plural vs. singular ending, etc). The question arises as to which of these types of errors are the most damaging to the overall quality of translation. Knowing this information can help direct research efforts towards those areas that are most likely to improve translation performance.

To carry out this assessment, the MT component of our system was first evaluated subjectively on a test set consisting of 419 Iraqi and 429 English utterances. A bilingual judge rated each MT output on a 1 – 5 Likert scale, where a score of 5 denoted perfect translation, 4 adequate translation, 3 semi-adequate, and so on. The results of this evaluation are shown in Table 4. The large difference in performance between E2I and I2E is due to the higher perplexity of the Iraqi set (586 vs. 54).

Translation	Type	Likert
E2I	T2T	4.28
	S2T	4.05
I2E	T2T	3.85
	S2T	3.35

Table 4: Likert Scores

As part of the subjective evaluation, the bilingual judge categorized and labeled the specific translation errors made by the MT. The set of error categories was created based on an initial review of the MT output. There were approximately 15 categories, which included major errors, such as dropping a concept or using the wrong sense of a word, and minor errors, such as using the singular form of a word instead of the plural. A principle goal of this effort was to quantify the relative importance of each error category in terms of the "damage" it did to the overall translation performance, so as to better direct our efforts towards improving the system.

As a subjective measure of translation quality, we use the familiar 1 – 5 Likert scale to rank both forward (i.e. English-to-foreign) translations, and back-translations. Note we do not assume that users will actually assign a Likert scores while using the system; but instead view the scores as numerical proxies for user reactions. We assign the following interpretations to the different elements of the Likert scale.

5: Essentially a perfect translation,

4: An adequate though somewhat disfluent translation which conveys the meaning of the utterance.

3: A partial translation which is missing one or more concepts, or is severely disfluent.

2: A translation which is missing most of the concepts.

1: A translation with no apparent relation to the input.

To quantify our notion of "damage", we define the "Likert Error" (LER) for a translation as 5 minus its Likert score. We then define the "Total Likert Error" (TLE) of a set of translations as the sum of the LER's of the translations. Table 5 gives TLE statistics for the utterances in I2E and E2I that contain errors. As can be seen, the average TLE per error and per utterance with error is higher for E2I, but I2E has many more utterances with an error. This is consistent with the lower average Likert score for I2E above.

	#Utts	#Errs	Errs/ Utt	Tot TLE	TLE/ Err	TLE/ Utt
E2I	184	228	1.24	305	1.34	1.7
I2E	273	383	1.40	484	1.26	1.3

Table 5: Total Likert Error Stats

To determine the damage done by each category of error, we make the simplifying assumption that the damage done by an individual error is at least approximately separable from and additive to the damage done by others. The relative importance of an error category C is then the fraction of the TLE that can be ascribed to its instances, or:

$$TLE(C) = Count(C)*LER(C),$$

where $LER(C)$ is the average damage done by instances of C, and quantifies the "seriousness" of the error.

Estimating LER(C) is not wholly straightforward, because many sentences have both multiple errors. For example, the same sentence might have both a "Word Sense" and an "Incorrect Pronoun" error. So we cannot determine the LER simply by averaging over instances of C. The key question is how to apportion the blame between these errors.

One might imagine various heuristic or hill-climbing approaches to this problem. Our approach instead views each annotated utterance as an equation, in which the annotator has asserted that the sum of the error labels equals the given Likert error value. The variables of this equation are the error labels, whose unknown values are the LER weights of the categories. The complete set of annotated utterances can then be viewed as a set of simultaneous equations over the LER's. That is, we seek $\boldsymbol{x}$ such that $\boldsymbol{Ax}=\boldsymbol{k}$, where $\boldsymbol{A}$ is a matrix of coefficients for each equation, $\boldsymbol{x}$ is the vector of unknown LER weights, and $\boldsymbol{k}$ is the vector of annotator-assigned LER values.

Due to the variability inherent in subjective analysis, one cannot in general expect this system of equations to be consistent. For example, a "Missing Concept" error might legitimately result in a higher Likert error in one sentence than in another, depending upon the missing concept. We must instead settle for an approximation $\boldsymbol{Ax=k+e}$, where $\boldsymbol{e}$ is the difference between the predicted and actual LER values, and seek the $\boldsymbol{x}$ that minimizes $|\boldsymbol{e}|$. Fortunately, this is just a least-squares problem, to which an exact solution can be found by solving the equation:

$$\boldsymbol{A^TAx = A^Tk}$$

Once we have estimated the LER value for a category, we multiply it by the frequency of the category to estimate the category's TLE. Table 6 gives the solved-for LER weights and the estimated TLE's for each language direction. Note that the categories "Word Sense", "Wrong Concept", "Missing Concept", and "Pronoun Error" account for the lion's share of the TLE. ("Wrong Concept" is a word or phrase translation that is wrong in all contexts, regardless of word sense). All have high frequencies and, except for "Pronoun Error", also high weights. "Pronoun Error" has a smaller weight (approximately 1.0), reflecting its lesser importance. The error "Wrong Polarity", (e.g. "I am *not* sick" instead of "I am sick") is given a high weight as it should, but because its frequency is low, it contributes only a small share to the TLE.

Iraqi-to-English			
	%Count	LER	%TLE
Word Sense	16.2	1.73	21.3
Wrong Concept	13.3	1.96	19.9
Missing Concept	13.1	1.73	17.2
Pronoun Error	21.4	0.94	15.3
Function Words	9.7	0.87	6.4
Word Order	8.6	0.83	5.5
Wrong Polarity	2.6	1.80	3.6
Other	15.1	--	10.8
English-to-Iraqi			
	%Count	LER	%TLE
Word Sense	17.1	1.88	23.5
Wrong Concept	14.5	2.00	21.4
Missing Concept	10.1	1.94	14.3
Pronoun Error	25.9	1.01	19.1
Function Words	10.5	1.07	8.2
Word Order	8.8	0.81	5.2
Wrong Polarity	0.4	2.0	0.6
Other	12.7	--	7.7

Table 6: Estimated Likert Error values

Interestingly, the weights for some minor errors, such as "Word Order", are driven below 1.0, even though 1 was the lowest Likert error the annotator could give a sentence that contained an error, since fractional scores were not allowed. Thus, the algorithm mitigates somewhat the rather severe quantization of the scoring system, which forces all imperfect but still adequate translations to have the same score. Of course, the advantage of the integer scale is that it is easier for annotators to use than real numbers. A useful future compromise would be to allow half-point scores.

5.2 Evaluating Back-Translation

A key issue in S2S systems is helping the speaker decide whether or not the system translated him correctly. If the user decides the system misunderstood or mistranslated what he said, he can take some form of remedial action in order to keep the dialog on track. Lacking such capability, translingual dialogs may swiftly founder due to mutual incomprehension. Many voice-only dialog and S2S systems use a "confirmation" utterance to convey the system's understanding of what the user said. The user is then allowed to "barge in" and re-speak his utterance if he decides the system was incorrect.

There are various approaches to generating the confirmation utterance. One is to simply read back the ASR output, on the theory that errors in concept words guarantee a mistranslation. However, this approach cannot detect errors that arise purely in the translation component, independent of ASR. An alternative approach is "back-translation", in which the system re-translates the output of a source- to target-language translation back into the original source language. The idea is that if the back-translation is reasonably close to the original source language input, the speaker can have confidence that he was translated correctly. Possible objections to back-translation, however, are that 1) it might frequently produce garbage even for good translations, and 2) even if garbage is not produced, the results of back-translation may yet be misleading, as merely using the translation model in the reverse direction may serve to mask errors in translation.

The most obvious way to evaluate the efficacy of back-translation would be to run two complete sets of live evaluations, one with back-translation and one without, and compare the results on measures such as concept transfer, rate of concept transfer, user satisfaction, and the like. However, such evaluation is expensive to conduct and non repeatable. We must therefore look for an offline method for evaluating back-translation.

Note that we are not interested in predicting the actual value of the Likert rating for the forward translation, but rather in simply predicting whether or not the forward translation's Likert rating is above a certain threshold of acceptability. Therefore, we seek to use the back-translation for binary classification, rather than regression. As is pointed out for another context by [12], even a poor approximation of a function may be adequate for classification, if all we care about is the sign of the function's value and not its magnitude.

First, we choose a specific minimum acceptable Likert score *F* for the forward translation (say a score of 4). We then test various minimum thresholds *B* for the back-translation Likert score. In particular, for utterances whose back-translation score is at or above the threshold *B*, we test the prediction that the utterance's forward translation Likert score will be at or above the threshold *F*, and thus acceptable. Below *B*, we predict that the forward translation Likert will be below *F*, and therefore unacceptable. We compute precision, recall, and F-measure for each such threshold.

There are of course different kinds of costs for false acceptance (a failure of precision) vs. false rejection (a failure of recall). A false acceptance, by allowing an incorrect or garbled forward translation to be sent to the FLE, incurs the risk that the FLE will misunderstand or be confused. A false rejection, on the other hand, means that the user will have to repeat or rephrase his utterance unnecessarily, which not only wastes a dialog turn, but also opens up the possibility that the retry itself will be mistranslated. Different application domains may impose different weights on these types of costs, which can be straightforwardly taken into account by computing the F-measure with a weighted harmonic mean.

To test the methodology outlined above, we used a set of 779 English utterances that were spoken to our system by SME users during the TransTac live evaluation in June 2008, conducted by the US government's National Institute of Standards and Technology (NIST). In this evaluation, active-duty military personnel played the part of the SMEs. Native speakers of Iraqi Arabic living in the US were recruited to play the part of FLEs. Participants worked through a set of specified scenarios in which each was briefed ahead of time with either the information he was to try to obtain (in the case of the SME), or the information he was to give (in the case of the FLE). The FLE wore noise-masking headphones so that he was not able to hear any of the English spoken by the SME, and thus had to rely on the system output only. The utterances of both parties, and the system's ASR and MT outputs, were recorded for later analysis.

The Iraqi translation output produced by the system for these utterances was Likert-scored by a native Arabic speaker experienced in the application domain. To produce the back-translations, we ran (offline) our Arabic-to-English MT on the system's Iraqi translation outputs. The back-translations thus produced were Likert-scored by a native English speaker knowledgeable in the application domain. For comparison, the same ranker Likert-scored the output of our system's English ASR for these same 779 utterances.). Half-scores (e.g. 4.5) were also allowed.

Some examples of back-translations and their Likert rankings are: "Turn off your vehicle" (for "Turn your vehicle off"), ranked 5; "Construction prior experience do you have" (for "Do you have prior construction experience"), ranked 4; and "How many subcontracting work" (for "How many subcontractors work for you") ranked 3. Table 7 shows the mean Likert scores for each of the conditions, namely, forward translation, back-translation, and ASR output of Likert scores for the back-translation.

Forward	BackTrans	ASR
4.42	3.99	4.64

Table 7: Mean Likert Scores

As can be expected, the highest mean Likert scores were produced on ASR output, which tends to overestimate the true (forward) Likert score, while the lowest were associated with back-translation output, which tends to underestimate it. Both were approximately equally well-correlated with forward Likert score, however, with a correlation coefficient of approximately 0.60. Table 8 gives a more detailed breakdown of the score distributions.

The large number of utterances scoring 5 for the ASR ranking is partly due to the low WER and low utterance rate for this set. The English ASR WER obtained on this corpus was 6.2%. And the English-to-Arabic BLEU score on this ASR output was 56.7%.

Likert	Forward	BackTrans	ASR
5.0	0.47	0.27	0.72
4.5	0.22	0.08	0.03
4.0	0.14	0.26	0.10
3.5	0.06	0.17	0.08
3.0	0.07	0.16	0.05
2.5	0.03	0.02	0.00
2.0	0.01	0.03	0.01
1.0	0.01	0.01	0.00

Table 8: Likert Score Distributions

To obtain results on back-translation efficacy, we set the desired forward translation Likert score threshold *F* to be 4.0. This may be considered a good minimum acceptable score for our purposes, as scores below 4.0 are associated with "semantic damage" to the translation. Table 3 gives acceptance rate, false rejection rate, false acceptance, F-measure, and precision-weighted F-measure for different back-translation Likert score cutoffs *B*. Each row of this table can be interpreted as a prediction rule, which predicts that an utterance whose back-translation Likert score is at or above the cutoff will have a forward translation whose Likert score will be 4.0 or higher.

Cutoff	Acpt	FlsRej	FlsAcc	FMsr	WFMsr
5.0	0.27	0.68	0.02	0.48	0.58
4.5	0.35	0.59	0.03	0.57	0.67
4.0	0.61	0.29	0.04	0.81	0.86
3.5	0.78	0.14	0.08	0.89	0.90
3.0	0.94	0.02	0.14	0.92	0.90
2.5	0.96	0.02	0.15	0.91	0.89
2.0	0.99	0.00	0.16	0.91	0.88
1.0	1.00	0.00	0.17	0.91	0.88

Table 9: Precision and Recall for Back-Translation

For many S2S applications, a false acceptance can be regarded as worse than a false rejection, because of the possibility of confusing the respondent, etc. For example, one might decide that a false acceptance is twice as bad as a false rejection. The rightmost column of Table 9 gives F-measure computed with these weights (0.67 vs. 0.33).

The results in Tables 8 and 9 seem to show that the worst fears regarding back-translation are not realized. Back-translation does not yield garbage all the time, nor is it a totally faithless guide. Indeed, for cutoffs of 4.0 or higher, its false acceptance rate is actually quite low. This precision does come at the expense of recall, however, and in particular at a cutoff of 4.0 fully 39% of SME utterances would be rejected and have to be retried. A better strategy might be a slightly less strict cutoff of 3.5, which yields a low false acceptance rate of 8%, while falsely rejecting only 14%. This rule corresponds to a back-translation which subjectively seems rather poor, but which is not completely deficient.

A key question to be addressed, however, is whether back-translation is better than the strategy of simply reading back the system's English ASR output. To address this question, Table 11 repeats the above experiment, but with Likert rankings on the system's English ASR output. It may be regarded as an experiment in which we pretend that the ASR output itself is the back-translation.

Cutoff	Acpt	FlsRej	FlsAcc	FMsr	WFMsr
5.0	0.72	0.19	0.07	0.86	0.88
4.5	0.75	0.17	0.08	0.88	0.89
4.0	0.86	0.07	0.10	0.92	0.91
3.5	0.94	0.02	0.13	0.92	0.90
3.0	0.99	0.00	0.16	0.91	0.89
2.5	0.99	0.00	0.16	0.91	0.89
1.0	1.00	0.00	0.17	0.91	0.88

Table 11: Precision and Recall for ASR Output

The false acceptance rate is higher than for back-translation, but the false rejection rate is much lower, yielding good F-measure scores at all values of the cutoff. For most cutoff values, the ASR read-back strategy even slightly out-performs back-translation on weighted F-measure. It might seem from this analysis that ASR read-back is therefore a superior strategy. It should be noted, however, that ASR read-back on this dataset has a floor of 7% false acceptance, below which it cannot possibly go. The back-translation strategy, by contrast, can go as low as 2% false acceptance, albeit at the price of a very high false rejection rate. If rather than seeking to maximize F-measure, one were to instead stipulate a certain maximum allowable rate of false acceptance – say 8% – the back-translation strategy could be seen as slightly superior, resulting in a 14% false rejection rate as opposed to ASR read-back's 17%.

6. CONCLUSIONS

We have presented our speech-to-speech translation system, TransTalk, and outlined several techniques used for automatically evaluating its performance. Among these techniques is an algorithm STER that highlights machine translation errors in a linguistically more meaningful way than other approaches. STER also correlates better with human judgment than does TER. We have also presented detailed analyses of TransTalk's performance. In particular, we have presented a method by which we can apportion blame to different MT error phenomena, and shown that word sense errors are the largest contributor to error of all categories. We have also presented a method for evaluating the effectiveness of the back-translation approach to user confirmation, and shown that back-translation provides higher precision than the simple strategy of reading back the ASR, at the expense of recall.

7. REFERENCES

[1] Stallard, D., et al. 2007. “The BBN 2007 Displayless English/Iraqi Speech-to-Speech Translation System,” Proc. EUROSPEECH, ICSA, Antwerp, Belgium, Sept. 2007.

[2] Prasad R., et al. 2005. “The 2004 BBN/LIMSI English Conversational Telephone Speech Recognition System,” Proc. EUROSPEECH, ISCA, Lisbon, Portugal, Sept. 2005.

[3] Nguyen, L, and Schwartz, R. 1997. "Efficient 2-pass N-best Decoder," Proc. EUROSPEECH, ISCA, Rhodes, Greece, Sep. 1997

[4] Koehn, P., Och, F., and Marcu D. 2003. “Statistical Phrase-Based Translation”, NAACL/HLT 2003, Proc. NAACL/HLT, Edmonton, Canada, May 27--June 1 2003

[5] Brown, P, Della Pieta, S, Della Pietra V, and Mercer, R. 1991. “The mathematics of statistical machine translation: parameter estimation,” Computational Linguistics, 19(2), 263 - 311, 1991

[6] Och, F and Ney, H. 2003. “A Systematic Comparison of Various Statistical Alignment Models,” Computational Linguistics, 29(1), 19 - 51, 2003

[7] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. 2002. “BLEU: a method for automatic evaluation of machine translation”. Proc. ACL-2002: 40th Annual meeting of the Association for Computational Linguistics, 2002.

[8] Banerjee, S. and Lavie A. 2005.. “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization.

[9] Fellbaum, C., ed. 1998. “WordNet: An Electronic Lexical Database”, MIT Press, May 1998.

[10] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. 2006. “A Study of Translation Edit Rate with Targeted Human Annotation,” Proceedings of Association for Machine Translation in the Americas, 2006.

[11] Subramanian, K., Stallard, D., Prasad, R., Saleem S., Natarajan, P. 2007. “Semantic translation error rate for evaluating translation systems”. Proc. ASRU 2007.

[12] McCallum, A. and Nigam, M. 1998. “A Comparison of Event Models for Naïve Bayes Text Classification,” in *Proc. AAAI-98 Workshop on Learning for Text Categorization*, 1998.

The Impact of Evaluation Scenario Development on the Quantitative Performance of Speech Translation Systems Prescribed by the SCORE Framework

Brian A. Weiss
National Institute of Standards and Technology
100 Bureau Drive, MS 8230
Gaithersburg, Maryland 20899
301-975-4373
brian.weiss@nist.gov

Craig Schlenoff
National Institute of Standards and Technology
100 Bureau Drive, MS 8230
Gaithersburg, Maryland 20899
301-975-3456
craig.schlenoff@nist.gov

ABSTRACT

The Defense Advanced Research Projects Agency's (DARPA) Spoken Language Communication and Translation for Tactical Use (TRANSTAC) program is a focused advanced technology research and development program. The intent of this program is to demonstrate capabilities to quickly develop and implement free-form, two-way, speech-to-speech spoken language translation systems allowing speakers of different languages to communicate with each other in real-world tactical situations without the need for an interpreter. The National Institute of Standards and Technology (NIST), with support from the Mitre Corporation and Appen Pty Limited, has been funded by DARPA to evaluate the TRANSTAC technologies since 2006. The NIST-led Independent Evaluation Team (IET) has numerous responsibilities in this ongoing effort including collecting and processing training data, designing and implementing performance evaluations, and analyzing the test data. In order to design and execute fair and relevant evaluations, the NIST IET has employed the System, Component and Operationally-Relevant Evaluation (SCORE) framework. The SCORE framework is a unified set of criteria and tools built around the premise that, in order to gain an understanding of how a technology would perform in its intended environment, it must be evaluated at both the component and system levels and further tested in operationally-relevant environments while capturing both quantitative and qualitative performance data. Since an evaluation goal of the TRANSTAC program is to capture quantitative performance data of the translation technologies, the IET developed and implemented SCORE-inspired live evaluation scenarios. The two developed forms of live evaluation scenarios have unique impacts on the quantitative performance data. This paper presents the TRANSTAC program and SCORE methodology, as well as the evaluation scenarios and their influence on system performance.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Machine translation, Speech recognition and synthesis, Text analysis.*

General Terms

Design, Experimentation, Languages, Measurement, Performance.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA. ACM 978-1-60558-747-9/09/09

Keywords

SCORE, TRANSTAC, Speech-to-Speech Translation System, Performance Metrics, Evaluation

1. INTRODUCTION

The Spoken Language Communication and Translation for Tactical Use (TRANSTAC) program is an advanced technology research and development program managed by the Defense Advanced Research Projects Agency[1] (DARPA) [3]. The objective of the TRANSTAC program is to demonstrate capabilities to rapidly develop and field free-form, two-way, speech-to-speech spoken language translation technologies that allow speakers of different languages to communicate with each other in real-world tactical situations without the need for an interpreter [7] [11]. To date, several prototype systems have been developed for various language domains in Iraqi Arabic, Mandarin, Farsi, Dari, Pashto, and Thai. Systems have been demonstrated on PDAs (Personal Digital Assistants), laptop-grade platforms, and compact, ruggedized laptop systems[2] with varying performance.

The primary use case of the TRANSTAC technology involves US military personnel and foreign language speakers engaging in a range of civilian and tactical dialogues. The anticipated concept of operation is that the English-speaking personnel will be trained in advance to use the technology, while it is assumed that the foreign language users will have little to no opportunity to become familiar with the system.

DARPA has funded the National Institute of Standards and Technology (NIST) to lead the evaluation of the TRANSTAC technologies, with support from the Mitre Corporation and Appen Pty Limited. As the Independent Evaluation Team (IET), NIST was tasked with capturing the required language training data, designing and implementing multiple evaluations to capture both

[1] The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

[2] Certain commercial products and software are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the products and software identified are necessarily the best available for the purpose.

technical performance and end-user utility assessment, and analyzing the data. This included the IET collecting technical performance data from the TRANSTAC systems under live test conditions. The IET utilized the System, Component, and Operationally-Relevant Evaluation (SCORE) framework to produce test scenarios that English and foreign language speakers used as the backbone of their dialogues between themselves while using the TRANSTAC technology [4]. These test scenarios directly impacted the metrics generated from measures captured from these test dialogues. To date, NIST has primarily evaluated English/Iraqi Arabic two-way systems along with English/Dari two-way systems.

This paper will discuss the following: Section 2 will provide background on the SCORE framework; Section 3 will present the high level concept transfer metrics that the evaluation sought to output; Section 4 will discuss the live evaluations including relevant technical performance metrics and test scenarios; Section 5 will discuss the impact of the test scenarios on the performance data[3]; Section 6 will offer a glimpse of future scenario design; and Section 7 provides conclusions.

2. SCORE METHODOLOGY

The SCORE framework is a design methodology that is built around the premise that, in order to get a true picture of how a system performs in the field, it must be evaluated at the component level, the capability level, the system level, and in operationally-relevant environments [3] [10].

SCORE is a cohesive suite of criteria and software tools employed to design performance evaluations for complex intelligent systems. It stipulates an extensive evaluation plan that is capable of both assessing technical performance through variable isolation and manipulation along with collecting end-user utility across a range of test environments.

SCORE sets itself apart from other methodologies since:

1. It can be applied to a broad range of technologies from manufacturing to defense systems
2. Elements of SCORE can be decoupled and customized based upon specific goals
3. It can evaluate a technology at varying stages of development, from conceptual to the final iteration
4. It combines the results of targeted evaluations to produce a comprehensive representation of a technology's capabilities, performance, and utility.

This framework has provided proven techniques to facilitate performance evaluations of numerous intelligent systems since it was conceived. To date, it has driven five TRANSTAC evaluations and six test events for DARPA's Advanced Soldier Sensor Information System and Technology (ASSIST) program [8] [12]. Likewise, the SCORE framework was employed to produce the initial designs for the RoboCup Rescue Virtual Robots Competition and Virtual Manufacturing Automation Competition (VMAC) [1] [2] [4].

2.1 Evaluation Goal Types

The SCORE framework has evolved over the years to define five evaluation goal types [10].

- *Component Level Testing – Technical Performance* – This evaluation type decomposes a system into components to isolate those subsystems that are critical to system operation. Ideally, all of the components taken together should include all facets of the system and yield a complete evaluation. This level of testing has occurred in past TRANSTAC evaluations where the three major components of speech-to-speech systems, Automatic Speech Recognition (ASR), Machine Translation (MT), and Text-to-Speech (TTS), were evaluated independently from one another.

- *Capability Level Testing – Technical Performance* – This type of evaluation involves identifying and isolating individual capabilities of a system and measuring their technical performance. A system can have one or more capabilities. This test type has also occurred in previous TRANSTAC evaluations when the IET designed and executed tests focused on the systems' capability of correctly translating proper names.

- *System Level Testing – Technical Performance* – This evaluation type is intended to assess the complete system, but in a controlled environment where test variables can be separated and influenced. The benefit is that tests can be performed using a combination of variables and parameters, where relationships can be determined between system behavior and these variables and parameters based upon the technical performance analysis. This test type inspired the live TRANSTAC evaluations and their driving scenarios which will be discussed in Section 4.

- *Capability Level Testing – Utility Assessments* – This evaluation type assesses the utility of an individual capability (where the complete system is made up of multiple capabilities), where utility is defined as the value the application provides to the end-user. Additionally, usability is assessed; this includes effectiveness, and user attitude towards the system. This test type also influenced several TRANSTAC evaluations where the IET captured end-user assessments of the technologies' ability to translate proper names.

- *System Level Testing – Utility Assessments* – This evaluation type assesses a system's utility and has inspired numerous live TRANSTAC technology evaluations. These include tests where Marines, Soldiers and foreign language experts provided utility feedback about the systems after using them in a range of tests.

It is important to note that even though the last two test types focus on extracting the technology users' utility assessment, it is virtually impossible to prevent the users' perceptions from being influenced by the technologies current level of technical performance. The users' utility is based upon the current state of the technology and is expected to change as the technical performance improves over future test events.

[3] Due to DARPA restrictions, the performance data captured using these test scenarios cannot be published. Instead, this paper will focus on the approach and impact as opposed to the results.

2.2 Evaluation Elements

The following evaluation elements must be identified for each goal type in order to generate relevant, reasonable, and appropriate evaluations [10].

- *Identification of the system or component to be assessed*
- *Definition of the goal/objective(s)/metrics/measures*
    - *Goal* – For a particular assessment, the goal is influenced by whether the intent of the evaluation is to inform or validate the system design.
    - *Objectives* – Evaluation objectives are used to separate evaluation concerns. These concerns also include identifying how different variables impact system performance.
    - *Metrics/Measures* – Depending upon the type of evaluation, either technical performance metrics and/or utility metrics are specified.
- *Specification of the testing environment* – Selecting the testing environment is influenced by a range of aspects including intended use-case environments, system maturity, etc.
- *Identification of Personnel* – This includes selecting the direct technology users, the test participants who will be indirectly interacting with the technology as role-players in the environment and evaluation personnel who will be directing role-players and/or capturing measures.
- *Specification of the personnel training* – All three personnel groups identified above must be given appropriate training and adequate practice time to become proficient in their test responsibilities.
- *Specification of the data collection methods* – Data capture methods, equipment, and/or instrumentation must be identified as measures and metrics are specified.
- *Specification of the use-case scenarios* – Test scenarios must be devised that are appropriate to the system or component being tested and the test end-users.

This paper is focused on the element of *Specification of the use-case scenarios* as designed and implemented within the *System Level Testing – Technical Performance* goal type. Prior to discussing these use-case scenarios, it is important to present the metrics that drive the scenario generation and implementation.

3. High Level Concept Transfer Metrics

Before discussing metrics specific to this work it is important to define both metrics and measures with respect to their usage by SCORE. Metrics are defined as the interpretation of one or more contributing elements, e.g. measures or other metrics that correspond to the degree to which a set of attribute elements affects its quality [6]. Likewise, a measure is defined as a performance indicator that can be observed, examined, detected, and/or perceived either manually or automatically [6]. For example, suppose it is desired to capture the velocity of a new vehicle under test. Examples of measures would be timing how long it takes a car to travel from one point to another and measuring the exact distance traveled. The velocity metric would be generated using the distance and time measurements where *velocity = distance/time*. Note that in some cases, a metric can be directly measurable. Using the same example, radar (or some other capture device) can directly capture the velocity of the vehicle making the measurement equal to the metric. Discussion will now follow of some of the technical performance metrics generated and/or captured during the TRANSTAC evaluations.

One of the key metrics that DARPA specified for evaluating the TRANSTAC technologies was the capture and analysis of *High Level Concept Transfer* Metrics. This suite of metrics reflects the goal of the TRANSTAC program which is the deployed use of the speech-to-speech machine translation technology to enable consistently successful communication between English-speaking and foreign language personnel [11].

Specifically, *High Level Concept Transfer* metrics consist of bilingual judges determining whether the meaning of a human-spoken utterance was conveyed during the machine translation. These metrics include the number of utterances that were successfully translated per ten minutes (with failed utterances not directly scored except for taking up time) so these metrics are assessments of both efficiency and accuracy. Additional *High Level Concept Transfer* metrics include:

- *Number of questions per 10 minutes* - Number of questions correctly translated in ten minutes as spoken by the English speaker
- *Question Percentage* - Percentage of questions that were correctly translated divided by the total number of questions asked
- *Number of attempts per question* - As spoken by the English speaker
- *Number of answers per 10 minutes* – Number of answers correctly translated in ten minutes as spoken by the foreign language speaker
- *Answer Percentage* – Percentage of answers that were correctly translated divided by the total number of answers stated
- *Number of attempts per answer* – As spoken by the foreign language speaker

It should be noted that these metrics are considered normalized since they can be computed using data from evaluation scenarios regardless of how much time it took to conduct each scenario.

Now that the evaluation type's required metrics are known, additional evaluation elements can be specified including the *Specification of the use-case scenarios.* To attain the *High Level Concept Transfer Metrics*, specific live evaluation scenarios have been designed and implemented across many of the TRANSTAC evaluations. These scenarios are discussed in the following section.

4. LIVE EVALUATIONS

A majority of each TRANSTAC evaluation features live scenarios performed by English-speaking Soldiers or Marines (also known as Subject Matter Experts or SMEs) and Foreign Language Experts (FLEs). These evaluations took place in both the lab (set up as an indoor, controlled environment where speakers remained stationary) and the field (outdoor, simulated tactical environments where the speakers were mobile and background noise was present) [7] [9] [11]. Figure 1 depicts a live field evaluation from a recent TRANSTAC test event.

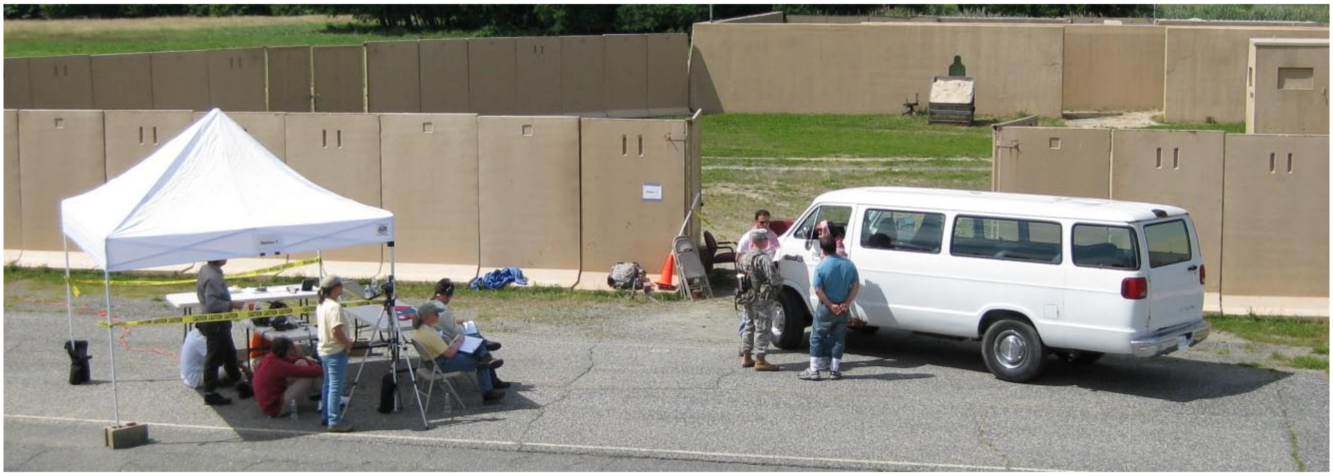


Figure 1. Live evaluation in the field environment at a recent TRANSTAC test event

Both of these test environments support the capture of quantitative and qualitative data and have featured two scenario types to attain these metrics: structured scenarios and spontaneous scenarios. The following sub-sections will present both types of scenarios and how they have been employed in the TRANSTAC evaluations. A final sub-section presents how the *High Level Concept Transfer* metrics are obtained from performing the two scenario types.

4.1 Structured Scenarios

Structured scenarios were intended to prompt the SME to ask the FLE questions (or convey information, in some instances) in order to obtain information from the FLE. The concept of this scenario type is that both speakers are told exactly what pieces of information they need to collect and/or convey. However, the speakers have the latitude to phrase their question and/or statement using whatever wording they choose so they can maximize their chances of a successful dialogue. A structured scenario is composed of two separate documents: the SME version and the FLE version. The SME version contains:

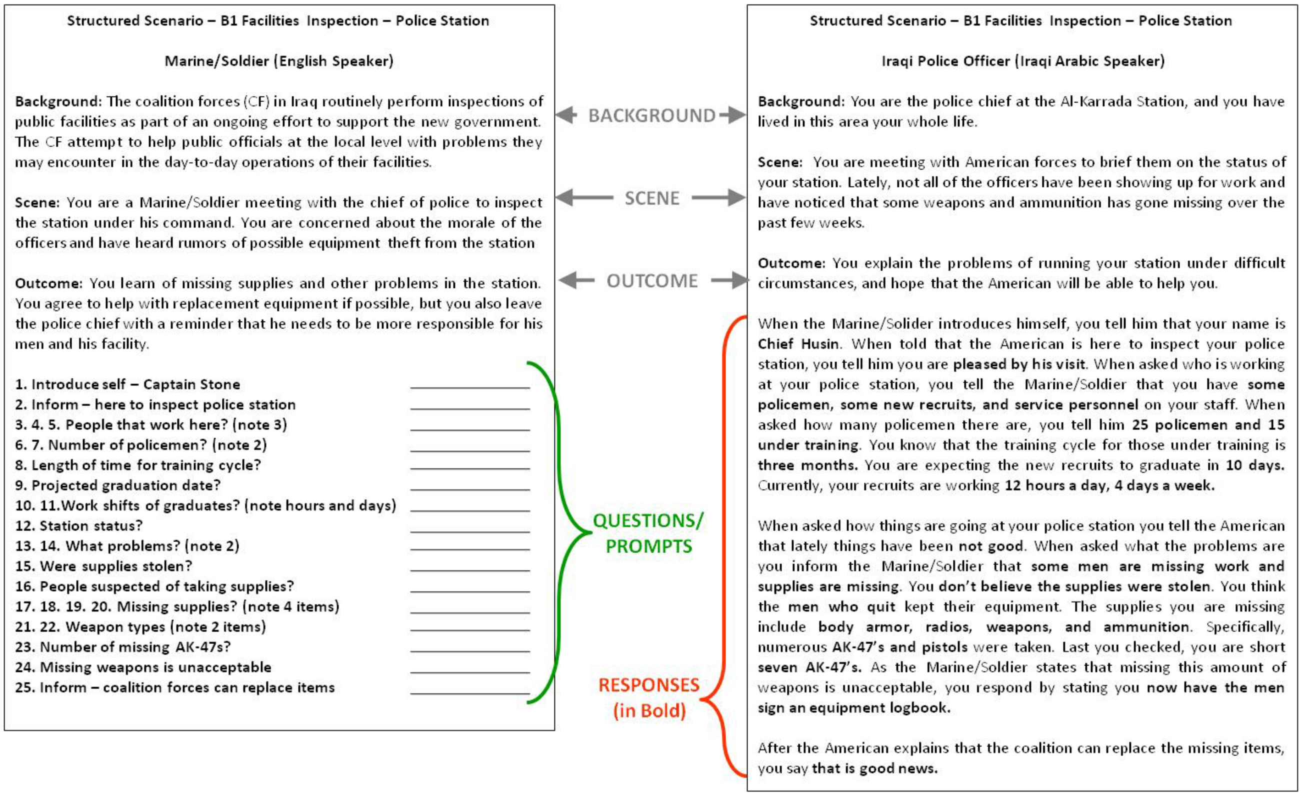
- **Background** – Specific information to put the SME in the appropriate mindset. This often includes high level goals and/or a snapshot of the current state of affairs.
- **Scene** – Describes the immediate situation and specific goals.
- **Outcome** – Presents the expected result of the conversation (as stated in the structured dialogue).
- **Questions/Prompts** – Numbered list of specific pieces of information the SME is to ask of the FLE or to convey to the FLE. Note that questions with multiple numbers indicate to the SME that there are multiple concepts to be obtained from the FLE.

Likewise, the FLE version contains **Background**, **Scene**, and **Outcome** elements, but they are stated from the FLE's perspective making them unique from the SME's version. Instead of **Questions/Prompts**, the FLE version contains **Responses** comprised of informational paragraphs. These include key pieces of information in bold throughout the paragraphs. An example of a structured scenario, showing both the SME and FLE (written in English) versions, is shown in Figure 2.

The evaluation protocol for the structured scenarios begins with the SMEs and FLEs each receiving their respective versions. After reviewing their dialogues separately, the SME and FLE practice their scenario together in their native languages through an interpreter (taking the place of a TRANSTAC system). After the training session is complete, the speakers participate in the evaluation. At this point, the SME is trained on the specific TRANSTAC technology they are about to use. However, the only training the FLE receives on the technology is in the form of TRANSTAC system spoken instructions that are played by the SME immediately before the evaluation dialogue begins. As the speakers are conversing through the TRANSTAC systems according to the structured format, the SMEs are informing an IET member of the concepts they perceived from the technology. For example, if a SME asks a FLE how many children he has and the FLE responds with "I am proud to have two sons," then the SME would simply report "two sons" to the IET.

Each structured scenario was conducted by a SME/FLE pair within a ten minute window. Since the scenarios were designed with more concepts than the speakers could reasonably get through in ten minutes, the speakers never reached the end of their structured scenario dialogues.

It should be noted that the content of each structured scenario is derived from audio dialogues that were collected by the IET ahead of each evaluation [7] [9] [11]. These 20 to 25 minute interpreter-mediated dialogues occurred between Marines or Soldiers and foreign language speakers within a recording studio. These dialogues were inspired by tactically-relevant data collection scenarios that the IET developed for the data collection efforts.



Structured Scenario – B1 Facilities Inspection – Police Station

Marine/Soldier (English Speaker)

Background: The coalition forces (CF) in Iraq routinely perform inspections of public facilities as part of an ongoing effort to support the new government. The CF attempt to help public officials at the local level with problems they may encounter in the day-to-day operations of their facilities.

Scene: You are a Marine/Soldier meeting with the chief of police to inspect the station under his command. You are concerned about the morale of the officers and have heard rumors of possible equipment theft from the station

Outcome: You learn of missing supplies and other problems in the station. You agree to help with replacement equipment if possible, but you also leave the police chief with a reminder that he needs to be more responsible for his men and his facility.

1. Introduce self – Captain Stone ______________
2. Inform – here to inspect police station ______________
3. 4. 5. People that work here? (note 3) ______________
6. 7. Number of policemen? (note 2) ______________
8. Length of time for training cycle? ______________
9. Projected graduation date? ______________
10. 11.Work shifts of graduates? (note hours and days) ______________
12. Station status? ______________
13. 14. What problems? (note 2) ______________
15. Were supplies stolen? ______________
16. People suspected of taking supplies? ______________
17. 18. 19. 20. Missing supplies? (note 4 items) ______________
21. 22. Weapon types (note 2 items) ______________
23. Number of missing AK-47s? ______________
24. Missing weapons is unacceptable ______________
25. Inform – coalition forces can replace items ______________

Structured Scenario – B1 Facilities Inspection – Police Station

Iraqi Police Officer (Iraqi Arabic Speaker)

Background: You are the police chief at the Al-Karrada Station, and you have lived in this area your whole life.

Scene: You are meeting with American forces to brief them on the status of your station. Lately, not all of the officers have been showing up for work and have noticed that some weapons and ammunition has gone missing over the past few weeks.

Outcome: You explain the problems of running your station under difficult circumstances, and hope that the American will be able to help you.

When the Marine/Solider introduces himself, you tell him that your name is **Chief Husin**. When told that the American is here to inspect your police station, you tell him you are **pleased by his visit**. When asked who is working at your police station, you tell the Marine/Soldier that you have **some policemen, some new recruits, and service personnel** on your staff. When asked how many policemen there are, you tell him **25 policemen and 15 under training**. You know that the training cycle for those under training is **three months.** You are expecting the new recruits to graduate in **10 days.** Currently, your recruits are working **12 hours a day, 4 days a week.**

When asked how things are going at your police station you tell the American that lately things have been **not good**. When asked what the problems are you inform the Marine/Soldier that **some men are missing work and supplies are missing**. You **don't believe the supplies were stolen**. You think **the men who quit kept their equipment**. The supplies you are missing include **body armor, radios, weapons, and ammunition**. Specifically, numerous **AK-47's and pistols** were taken. Last you checked, you are short **seven AK-47's.** As the Marine/Soldier states that missing this amount of weapons is unacceptable, you respond by stating you **now have the men sign an equipment logbook.**

After the American explains that the coalition can replace the missing items, you say **that is good news.**

Figure 2: Structured scenario outlining a police station inspection dialogue between a Marine/Soldier and Iraqi police officer

4.2 Spontaneous Scenarios

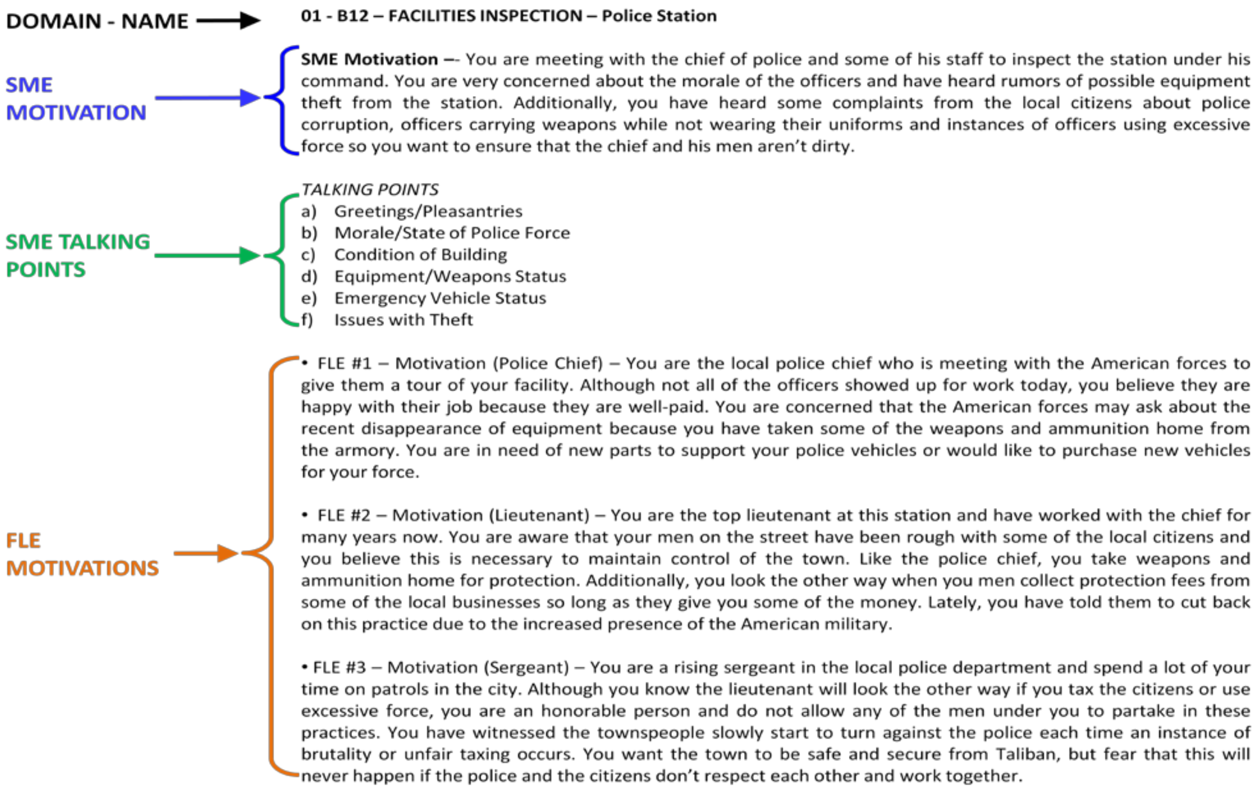
The spontaneous scenarios provided the SMEs and FLEs with more freedom and latitude in their dialogues by not laying out specific questions and answers as compared to the structured scenarios.

A spontaneous scenario begins with specifying the overall domain (six tactical domains were commonly identified for the Iraqi Arabic and Dari systems). For each domain, multiple SME motivations were generated that included some background and situational information along with the mindset the SME should take in the conversation. Additionally, each SME motivation was paired with numerous talking points with the intent of giving the SME topics they could include in their dialogues, but not limit them to specific questions. In turn, each scenario provided the FLE with a specific motivation including some background information. These can be seen in the example shown in Figure 3.

Since these scenarios have been used in evaluations involving a single SME and multiple FLEs per conversation, it was important to generate multiple FLE motivations corresponding to a single SME motivation. Each FLE motivation was designed to be unique from one another even though they applied to the same scenario. However, the FLE motivations built upon one another where each FLE's information either supported one another, created a broader picture, or purposefully contradicted one another. An example of this can be seen in Figure 3.

An additional consideration in creating the spontaneous scenarios was the environment where they were employed. Dialogues will naturally play out differently given the environment and specific props available for the speakers to comment and discuss. Using the police station facilities inspection scenario noted in Figure 3 as an example, it is possible to have drastically different dialogues if this scenario were performed in a very old, run-down building as compared to conducting the same scenario in a brand-new, pristine facility. The more realistic the evaluation environment, the more representative the dialogues will be when driven by spontaneous scenarios.

The evaluation protocol began with each speaker being given their own motivation and unable to see their counterpart's. The SMEs and FLEs were trained separately from one another with IET assistance. Their training covered possible dialogue directions along with how to interact with one another in the simulated tactical environments set up for the evaluation. Since the scenarios required the SMEs to have a tactical background in the areas they would be discussing, the IET considered their individual experiences when devising scenario assignments. In some instances, SMEs were paired with scenarios that they were unfamiliar with so they worked with other SMEs and IET members to better understand the domains. SMEs and FLEs received comparable technology training as if they had been doing structured scenarios. The SMEs received extensive training on the systems prior to the evaluation while the FLE was played verbal instructions from the system immediately before their evaluation dialogues began.



01 - B12 – FACILITIES INSPECTION – Police Station

SME Motivation –- You are meeting with the chief of police and some of his staff to inspect the station under his command. You are very concerned about the morale of the officers and have heard rumors of possible equipment theft from the station. Additionally, you have heard some complaints from the local citizens about police corruption, officers carrying weapons while not wearing their uniforms and instances of officers using excessive force so you want to ensure that the chief and his men aren't dirty.

TALKING POINTS

a) Greetings/Pleasantries
b) Morale/State of Police Force
c) Condition of Building
d) Equipment/Weapons Status
e) Emergency Vehicle Status
f) Issues with Theft

- FLE #1 – Motivation (Police Chief) – You are the local police chief who is meeting with the American forces to give them a tour of your facility. Although not all of the officers showed up for work today, you believe they are happy with their job because they are well-paid. You are concerned that the American forces may ask about the recent disappearance of equipment because you have taken some of the weapons and ammunition home from the armory. You are in need of new parts to support your police vehicles or would like to purchase new vehicles for your force.

- FLE #2 – Motivation (Lieutenant) – You are the top lieutenant at this station and have worked with the chief for many years now. You are aware that your men on the street have been rough with some of the local citizens and you believe this is necessary to maintain control of the town. Like the police chief, you take weapons and ammunition home for protection. Additionally, you look the other way when you men collect protection fees from some of the local businesses so long as they give you some of the money. Lately, you have told them to cut back on this practice due to the increased presence of the American military.

- FLE #3 – Motivation (Sergeant) – You are a rising sergeant in the local police department and spend a lot of your time on patrols in the city. Although you know the lieutenant will look the other way if you tax the citizens or use excessive force, you are an honorable person and do not allow any of the men under you to partake in these practices. You have witnessed the townspeople slowly start to turn against the police each time an instance of brutality or unfair taxing occurs. You want the town to be safe and secure from Taliban, but fear that this will never happen if the police and the citizens don't respect each other and work together.

Figure 3. Spontaneous scenario outlining a police station inspection dialogue between a Marine/Soldier and Afghan police officer

The evaluations then commenced and the SMEs and FLEs role-played their dialogues. The spontaneous scenarios ran differently than the structured scenarios in that the speakers had 15 to 35 minute windows to speak based upon the evaluation schedule. Since all scenarios ran for unique amounts of time, the normalized metrics (discussed in Section 3) applied in the structured scenarios were also applicable here. This enabled the evaluation team to conduct a more "apples-to-apples" comparison of the data given the varying scenario times.

It should be noted that these scenarios stemmed directly from corresponding data collection scenarios, but were augmented to support the evaluation [9]. Like the structured scenarios, the spontaneous scenarios were also based upon the audio dialogues collected at IET-led data collections.

4.3 Metrics Generated from Scenario Data

Both the structured and spontaneous scenarios served their purpose of enabling the evaluation team to generate *High Level Concept Transfer* metrics from the live conversations between English and foreign language speakers using the TRANSTAC technologies.

Since the structured scenarios provided the IET with the concepts that were conveyed by the speakers before the evaluation began, scoring spreadsheets were devised ahead of time to support the data analysis. Once the evaluation concluded, the IET enlisted the support of ten bilingual judges to assess the accuracy of the machine translations as compared to the human speech from the SMEs and FLEs. Between three to six judges assessed each evaluation dialogue which entailed viewing and listening to the recorded scenario, noting how many attempts a speaker made to convey a concept, and scoring how successful the technologies were in translating the spoken concepts. At the conclusion of the bilingual judges' analysis, the IET averaged out all of the judgments for each scenario and calculated the metrics discussed in Section 3.

Analyzing the data from the spontaneous scenarios was similar with the exception of one time-consuming and critical difference; since the scenarios were spontaneous in nature and the concepts to be conveyed were not known ahead of time, the IET had to transcribe the evaluation conversations and identify the concepts that the speakers were attempting to transfer. Ultimately, both scenarios produced the same desired metrics to assess this aspect of the TRANSTAC technologies' technical performance.

5. SCENARIO IMPACT ON METRICS

Both the structured and spontaneous scenario types impacted the evaluation dialogues which in turn, impacted the *High Level Concept Transfer* metrics. The following sub-sections present the specific impacts and how these affected the metrics.

5.1 Impacts

When conducted across multiple evaluations, the structured scenarios allowed the following with each having unique effects.

- The same structured scenarios using the same concepts were used across multiple evaluations.
  EFFECT – Direct technical performance comparisons were drawn across multiple technologies over multiple evaluations enabling more "apples-to-apples" assessments.
- SMEs and FLEs did not need firsthand knowledge of a particular scenario to be effective as long as they had sufficient training to become familiar with the concepts.

EFFECT – It was easier to obtain some repeatability across multiple speakers who performed the same scenarios.

- SMEs and FLEs were forced to attempt specific concepts, some of which were not easily understood by the technology.

  EFFECT – Technologies had to attempt varying targeted and challenging vocabulary that would have not been otherwise attempted.
- SMEs and FLEs were given little flexibility in their dialogues so it was easy for them to become disengaged in their conversations.

  EFFECT - Speakers were more prone to speaking less-naturally leading to a decrease in the ability of the technology to recognize their speech.

  EFFECT – Speakers were prone to reading concepts verbatim from the scenarios, as opposed to rephrasing, even when they had to repeat them due to miscommunications.

The spontaneous scenarios counteracted some of the negative consequences of the structured scenarios while producing some other effects, as well. These scenarios allowed the following producing the noted affects.

- Speakers used the system in the anticipated manner in which it would be deployed in more relevant, use-case environments.

  EFFECT – The output metrics provided a more representative gauge of how the system would perform in actual use-case environments.
- The same scenarios using the same talking points could be used across multiple evaluations but would still ultimately produce very distinct dialogues.

  EFFECT - It would be very challenging to make direct technical performance comparisons across multiple evaluations.
- SMEs and FLEs had great flexibility in their dialogues as long as their responses stay consistent and they remain within the scope of the scenario.

  EFFECT – The speakers made the scenarios "their own" thereby becoming more engaged and enthusiastic.
- SMEs must have firsthand knowledge of a scenario's tactical domain to effectively role-play the conversation during the evaluation.

  EFFECT - All of the dialogues were unique since they were based upon the SMEs' distinct experiences.
- SMEs and FLEs must improvise during their conversations in the event that the TRANSTAC systems were having difficulties with specific areas of dialogue.

  EFFECT – Dialogues easily stalled if the speakers did not change their wording or conversation direction based upon the systems' vocabulary capabilities.

5.2 Impact Analysis

After analyzing the *High Level Concept Transfer* metrics from multiple evaluations that were supported by structured and spontaneous scenarios, the following observations were made:

- On average, scores across all of the *High Level Concept Transfer* metrics were lower for those scenarios that forced the speakers to use specialized vocabulary, i.e. the scenarios performed within the medical domain scored lower as compared to the overall averages
- On average, the *Number of Attempts per Question* and *Number of Attempts per Answer* were higher for evaluations supported by the spontaneous scenarios as compared to the structured scenarios
- On average, the *Number of Questions per 10 minutes* and the *Number of Answers per 10 minutes* were lower for evaluations supported by the spontaneous scenarios as compared to the structured scenarios
- On average, the *Question Percentage* (number of questions correctly translated over the number of total questions asked) and the *Answer Percentage* were lower for those evaluations supported by the spontaneous scenarios as compared to the structured scenarios
- Since the FLEs had more flexibility in the spontaneous scenarios and weren't constrained to specifying multiple concepts per response, as they were in the structured scenarios, the average ratio of questions to answers was lower in this scenario type as compared to the structured scenarios resulting in less answer opportunities

It is important to note that the scenarios were not the only significant factor contributing to the disparity in results of metrics when applied to data from structured and spontaneous scenarios. All of the *High Level Concept Transfer* metrics captured using the structured scenarios have resulted from evaluations testing the Iraqi Arabic (IA) TRANSTAC systems. In contrast, the spontaneous scenarios have only been applied to the most recent evaluation which tested the Dari versions of the TRANSTAC technology. Additionally, the technology developers have had access to the IA data for a much longer period of time as compared to the Dari data. Also there is much more IA conversation data available to support training and development efforts as compared to the limited amount of available Dari data.

6. FUTURE EFFORTS

The IET is expecting to deploy another round of spontaneous scenarios to support the October 2009 evaluation. The overriding factor in the selection of spontaneous scenarios over structured scenarios is that the spontaneous scenarios enable the speakers to use the system in the expected manner in which it would ultimately be deployed, thereby providing an indication of the technology's current performance level under these conditions. This is critical considering it is desired to provide this technology to Soldiers and Marines operating within tactical environments in the near future.

The October 2009 evaluation will test the TRANSTAC research teams' two-way, English/Pashto systems to capture both technical performance (including the discussed *High Level Concept Transfer* metrics) and end-user qualitative assessments. The IET is exploring ways to augment the spontaneous scenario including the addition of suggested pieces of information to capture, i.e. presenting the SME with structured scenario-like prompts that they could optionally ask. Ultimately, the SME would still be free to take the conversation in any direction within the scenario's scope, but would have the fallback option to ask some (or all, at their discretion) of the IET-specified questions. However, the FLEs' scenarios would remain unchanged. Their dialogue would still be governed by their scenario-driven motivation where they would respond with answers relevant and consistent with the scenario.

7. CONCLUSION

SCORE has proven to be an invaluable evaluation design generation tool in formulating appropriate performance tests for DARPA's TRANSTAC technologies. This framework inspired the creation and implementation of the structured and spontaneous scenarios across multiple test events. Each scenario type has yielded vast amounts of data to support the suite of *High Level Concept Transfer* metrics necessary to the IET's evaluation. To date, SCORE has driven the development of 11 DARPA evaluations including six for the ASSIST program and five for the TRANSTAC program along with providing design inspiration to the VMAC and RoboCup Rescue Virtual Robot competitions. Based upon the success of these evaluations including the comprehensive levels of data generated, the IET envisions using this framework to support future evaluations of advanced technologies and other intelligent systems under test.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the DARPA TRANSTAC program manager, Dr. Mari Maeda, and the members of the NIST IET for their continued support.

9. REFERENCES

[1] Balakirsky, S., Carpin, S., Dimitoglou, G., and Balaguer, B. 2009, "From Simulation to Real Robots with Predictable Results: Methods and Examples," in *Performance Evaluation and Benchmarking of Intelligent Systems*, New York: Springer Science & Business Media, 2009, pp. 113-137.

[2] Balakirsky, S. and Madhavan, R. 2009. Advancing Manufacturing Research Through Competitions. In Proceedings of the SPIE Defense Security and Sensing Conference (Orlando, Florida, USA, April 13 – 17, 2009).

[3] DARPA. 2009. Spoken Language Communication and Translation System for Tactical Use (TRANSTAC). http://www.darpa.mil/IPTO/programs/transtac/transtac.asp

[4] Intelligent Systems Division – National Institute of Standards and Technology. 2009. Measurement Science for Intelligent Manufacturing Robotics and Automation Program. http://www.nist.gov/mel/isd/si/msimra.cfm

[5] Intelligent Systems Division – National Institute of Standards and Technology. 2009. System, Component and Operationally-Relevant Evaluations (SCORE). http://www.isd.mel.nist.gov/projects/score/

[6] Schlenoff, C., Steves, M.P., Weiss, B.A., Shneier, M. and Virts, A. 2007. Applying SCORE to Field-Based Performance Evaluations of Soldier Worn Sensor Technologies. Journal of Field Robotics – Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems, vol. 24 (Sept 2007), pp. 671 – 698.

[7] Schlenoff, C., Weiss, B.A., Steves, M.P., Sanders, G., Proctor, F., and Virts, A. 2009. Evaluating Speech Translation Systems: Applying SCORE to TRANSTAC Technologies. To Appear In Proceedings of the 2009 Performance Metrics for Intelligent Systems (PerMIS) Workshop (Gaithersburg, Maryland, USA, September 21 – 23, 2009).

[8] Schlenoff, C., Weiss, B.A., Steves, M., Virts, A, and Shneier, M. 2006. Overview of the First Advanced Technology Evaluations for ASSIST. In Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop (Gaithersburg, Maryland, USA, August 21 – 23, 2006).

[9] Weiss, B.A. and Menzel, M. 2009. Development of Domain-Specific Scenarios for Training and Evaluation of Two-Way, Free Form, Spoken Language Translation Devices. In Proceedings of the 2009 International Test and Evaluation Association (ITEA) Symposium (Baltimore, Maryland, USA, September 28 - October 1, 2009).

[10] Weiss, B.A. and Schlenoff, C. 2008. Evolution of the SCORE Framework to Enhance Field-Based Performance Evaluations of Emerging Technologies. In Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop (Gaithersburg, Maryland, USA, August 19 - 21, 2008).

[11] Weiss, B.A., Schlenoff, C., Sanders, G.A., Steves, M.P., Condon, S., Phillips, J., and Parvaz, D. 2008. Performance Evaluation of Speech Translation Systems. In Proceedings of the 6th edition of the Language Resources and Evaluation Conference (Marrakech, Morocco, May 28 – 30, 2008).

[12] Weiss, B.A., Schlenoff, C., Shneier, M., and Virts, A. 2006. Technology Evaluations and Performance Metrics for Soldier-Worn Sensors for ASSIST. In Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop (Gaithersburg, Maryland, USA, August 21 – 23, 2006).

Probability of Successful Transfer of Low-level Concepts via Machine Translation: A Meta-evaluation

Gregory A. Sanders
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-8940
(301) 975-4451

gregory.sanders@nist.gov

Sherri Condon
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102

scondon@mitre.org

ABSTRACT

In this paper, we present one of the important metrics used to measure the quality of machine translation in the DARPA TRANSTAC program. The metric is stated as either the probability or the odds of a machine translation system successfully transferring the meaning of *content words* (nouns, verbs, adjectives, adverbs, plus the most important quanitifiers and prepositions). We present the rationale for the metric, explain its implementation, and examine its performance. To characterize the performance of the metric, we compare it to utterance level (or sentence-level) human judgments of the semantic adequacy of the translations, obtained from a panel of bilingual judges who compare the source-language input to the target-language (translated) output. Language pairs examined in this paper include English-to-Arabic, Arabic-to-English, English-to-Dari, and Dari-to-English.

Categories and Subject Descriptors

D 2.8 **[Metrics]** Performance Measures.

I 2.7 **[Natural Language Processing]** Machine Translation.

:General Terms

Languages, Measurement.

Keywords

TRANSTAC, low-level concept transfer

Copyright

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA. Copyright © 2009 ACM 978-1-60558-747-9/09/09...$10.00

1. INTRODUCTION

In this paper, we present one of the important metrics used to measure the quality of machine translation in the DARPA TRANSTAC program. TRANSTAC is intended to develop speech-to-speech machine translation that can be used by U.S. soldiers and marines who speak only English but who need to communicate with civilians in other countries who speak only other languages. The metric is stated as either the probability or the odds of a machine translation system successfully transferring the meaning of *content words* (nouns, verbs, adjectives, adverbs, plus the most important quanitifiers and prepositions) from the source-language spoken input, into the target-language output. TRANSTAC systems take spoken input via automatic speech recognition (ASR) pipelined with machine translation (MT) and then speak out the translation via a text-to-speech (TTS) process.

We refer to the identified content words as *low-level concepts.* The metric is explained in more detail in section 4 of this paper. Transfer of the low-level concepts is scored one concept at a time, as successfully transferred, deleted, or substituted (judges can also identify inserted concepts). Sanders, et al., (2008) provides some earlier analyses of the metric.

In this paper, we present the rationale for the metric, explain its implementation, and examine its performance. To characterize the performance, we compare it to utterance level (or sentence-level) human judgments of the semantic adequacy of the translations, obtained from a panel of bilingual judges who compared the source-language input to the target-language (translated) output.

Language pairs examined in this paper include English-to-Arabic, Arabic-to-English, English-to-Dari, and Dari-to-English. Correspondingly, we will examine the low-level concepts in English, Dari, and Arabic. The Arabic dialect is Iraqi. The languages reflect countries in which many U.S. military people are deployed.

2. MOTIVATION FOR THE METRIC

One important motivation was to create a metric whose values would be intuitively meaningful. Another comes from the fact that the target languages in which translations are to be assessed in TRANSTAC are quite dissimilar. English, Dari, and Arabic come from different families of languages. Arabic is a “central semitic” language very different from Indo-European languages. Dari, one of the two dominant languages spoken in Afghanistan, is a dialect of the Persian language. Dari is part of the Indo-Iranian group of Indo-European languages and is distinctly different from English, although not as different as Arabic. In the future, we expect to also deal with Pashto, the other dominant language of Afghanistan. The fact that the languages are so varied presents challenges for common automated metrics for machine translation, most of which were designed to assess translations into English and implicitly assume linguistic characteristics of English or similar languages.

Most automated metrics for machine translation look at “*n*-gram co-occurrence statistics.” For example, the commonly used BLEU metric (BiLingual Evaluation Understudy, Papineni, et al.; 2001, 2002) compares a machine translation output to a set of (usually four) independent high-quality translations created by human translators. The BLEU metric looks at co-occcurrences of individual words (also called unigrams), pairs of words (called 2-grams or bigrams), triples of words (3-grams or trigrams), and 4-grams. We refer to these, collectively, as *n*-grams. The BLEU metric does a weighted average of the fractions of the unigrams, bigrams, trigrams, and 4-grams in the machine translation output that also occur in one or more of the human translations (reference translations). Looking at bigrams, trigrams, and 4-grams makes sense for English, because it is a measure of getting words in the right order, which is an important aspect of English syntax. Getting words in the same order as a human translator is also a useful indicator of *fluency* in English. But other languages communicate “who did what to whom” by means of *affixes* on words (for example, “whom” communicates that it is an object rather than subject by means of an “m” affixed to “who”). In scoring translations into languages whose syntax relies more heavily on affixes rather than on word order, scoring the longer n-grams that BLEU examines can be counter-productive.

A related problem with the longer *n*-grams is that a paragraph-length passage has about the same number of 4-grams as of unigrams. BLEU and similar metrics that consider longer n-grams (such as 3-grams and 4-grams) were designed with paragraph-length texts in mind. But for speech-to-speech machine translation, one is normally scoring a sentence-length utterance, which will have proportionately few 3-grams and 4-grams. For example, consider the greeting, “Good morning, my name is Joe.” That greeting has six unigrams, but only three 4-grams. Correspondingly, the probability that the 4-grams will co-occur is lower, making the value returned by the metric be somewhat affected by the length of the utterance.

Finally, the significance of a unigram error differs for languages that rely heavily on affixes. Consider the English phrase “to the program”, which consists of three words. Each of the three may independently co-occur (or not) between the machine translation output and a reference translation. Looking at only unigrams, the translation “to a program” has two unigrams correct and one unigram (the vs. a) wrong. The Arabic equivalent of “to the program” is للبرنامج (llbrnAmj) which has three elements: *l*- ‘to’, *Al*- ‘the’, and *brnAmj* ‘program’. The Arabic equivalent of “to a program is لبرنامج (lbrnAmj). But the single error will cause automated metrics to in effect count all three elements as wrong in Arabic, where in English they would count only one as wrong. Condon, et al., (2009) discussed relevant aspects of Arabic morphology.

All this led to the desire for a metric whose numeric values would be independent of the utterance length, comparable across very different language pairs (combinations of source language and target language), and intuitively meaningful. Extensive discussions within the TRANSTAC research community made it clear that some measure of low-level concept transfer could have those desired properties. Many possible schemes were proposed for what to take as those low-level concepts. Some of the proposals looked at structural relations among concepts and some proposals included the notion of more-important vs. less-important concepts. Arriving at a scheme that would have similar concept counts in very different languages was difficult. Agreement as to what to do did not materialize, however.

At some point, Sherri Condon had suggested just using the content words as the low-level concepts. When circumstances finally forced a choice in order to be able to proceed, that was the approach that was chosen. A key goal of this paper is to examine how well using the content words has worked out.

3. RATIONALE FOR CONTENT WORDS

Content words occur in essentially all languages, and the number of content words in the source language input and the target language output tends to be quite similar. For that reason, a numeric measure of the probability that the meaning of the source language content words will be correctly transferred by the translation is somewhat language-independent. This was desired because we wanted to be able to say whether translations into one language were about as good as translations into another—for example, whether the translations from English to Arabic were about as good as the translations from English to Dari. We calculate the probability of success because probability is a linear scale that lends itself to calculating correlations to other metrics. We state the value as odds of success (the number transferred correctly, divided by the number of errors) because we can conveniently describe progress from one evalaution to the next as an odds ratio, which is a familiar statistic that is well understood.

We chose to weight the content words equally because there is no agreed on way to measure the relative importance of different content words. Further, the usual metric for speech recognition accuracy (word error rate) weights all words equally, and that metric has worked out well in practice. This choice to weight the low-level concepts equally was somewhat controversial, so we want to look at how well that choice has worked out.

But what are the advantages of using content words as our low-level concepts? There are three. First, it puts an upper-bound on the number of low-level concepts: one concept per content word. And that limit directly reflects choices made by the speaker. Second, it is a fact about human languages that any piece of meaning expressed by function words, syntax, context, and so forth can be expressed by a content word, should the speaker choose to give it greater prominence by doing so. Thus, the content words reflect the speaker’s choice of what to make prominent. Third, content words tend to *lexicalize* entire complexes of meaning. Consider the noun “steer.” Saying or writing that word conveys, in one word, that it is a male bovine that is castrated by humans, being kept by humans, and destined to be slaughtered for meat. A bilingual human judge can decide whether the meaning intended by the speaker’s choice of the source language word “steer” ended up being adequately conveyed by the target language translation. Thus, using the content words directly reflects the speaker’s choices of how to say things. The probability or odds of a translation successfully transferring the meaning of the content words is a directly quantitative measure of the transfer of the low-level elements of meaning in each utterance.

4. IMPLEMENTING THE METRIC

We had an analyst who is a native speaker of each source language identify the low-level elements of meaning (low-level concepts) in the input utterances from several representative excerpts of conversations where both directions were translated by the TRANSTAC speech-to-speech machine translation systems. We then asked a panel of five bilingual judges to tell us which pre-identified low-level concepts were successfully transferred into the target-language output (where failures are deletions, substitutions, or insertions of concepts).

4.1 Source-language Annotation

To standardize the annotation of the source-language concepts, we created a fairly extensive guidelines document explaining what to mark as a concept. The guidelines document includes rules for what to group as a single concept. The following are some examples of groups of words that count as just one concept.

□ words that make up a number: "four hundred and twenty seven"

□ words that make up a date: "June 12, 1979"

□ words that make up an address: "1313 Mockingbird Lane"

□ words that make up a verb tense in English: "would have been known"

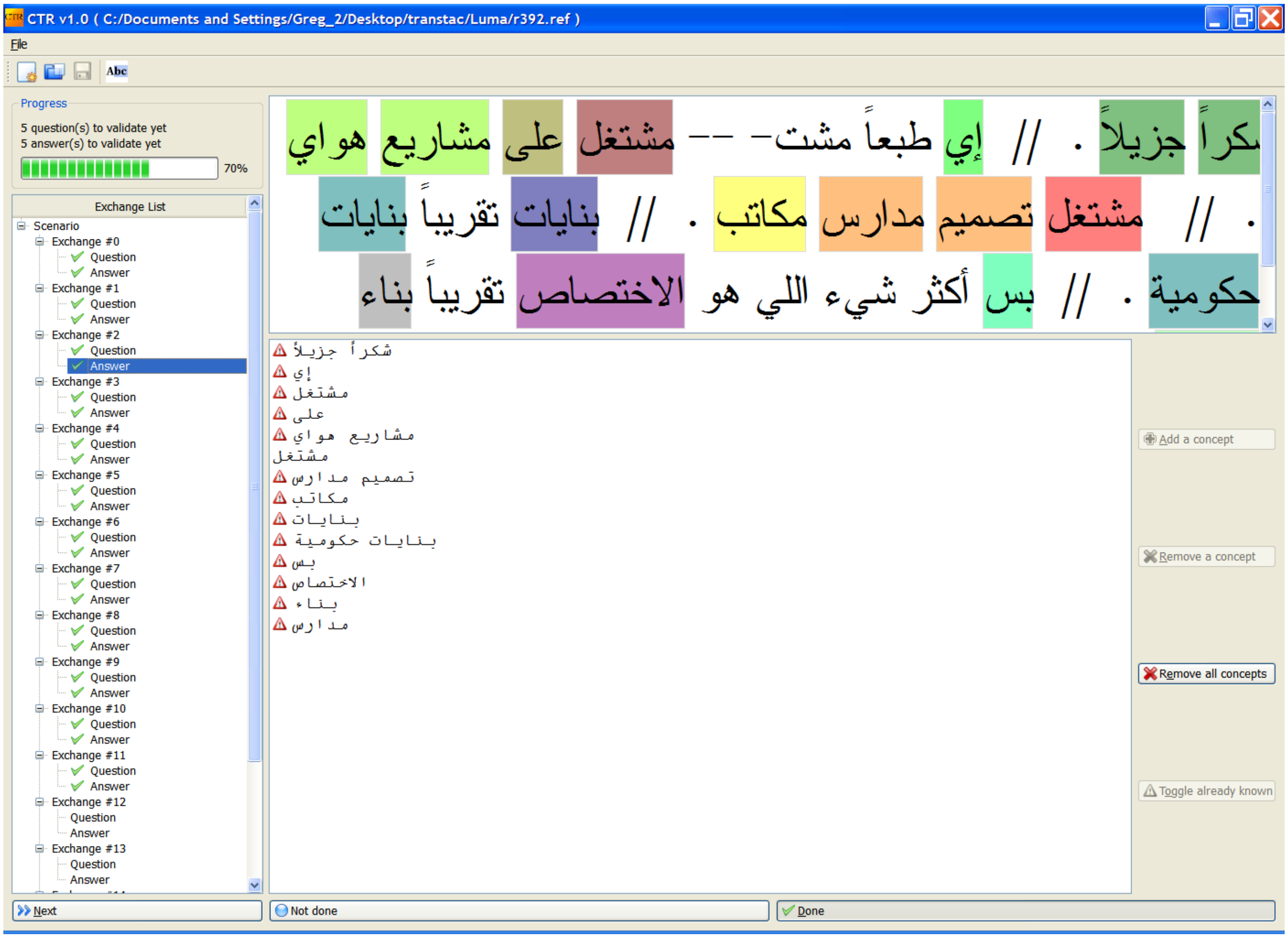
□ noun-noun constructions: "tea napkin" or "Abrams tank"

□ phrasal verbs: "look up" (e.g., a word in a dictionary).

As a concession to make things easier for Arabic judges, we chose to group possessive pronouns with the thing possessed, for example "his car."

Correspondingly, for source languages that rely heavily on affixes, the source-language annotator must sometimes mark affixes as independent low-level concepts. In some languages, including Arabic, "two locks" would be a single word in *dual* number, which should be marked as a separate source-language concept.

The source language annotators used a software tool called CTR, written by Sébastien Bronsart.



CTR v1.0 (C:/Documents and Settings/Greg_2/Desktop/transtac/Luma/r392.ref)

5 question(s) to validate yet
5 answer(s) to validate yet
70%

كراً جزيلاً . // إي طبعاً مشت– –– مشتغل على مشاريع هواي
. // مشتغل تصميم مدارس مكاتب . // بنايات تقريباً بنايات
حكومية . // بس أكثر شيء اللي هو الاختصاص تقريباً بناء

⚠ شكراً جزيلاً
⚠ إي
⚠ مشتغل
⚠ على
⚠ مشاريع هواي
مشتغل
⚠ تصميم مدارس
⚠ مكاتب
⚠ بنايات
⚠ بنايات حكومية
⚠ بس
⚠ الاختصاص
⚠ بناء
⚠ مدارس

Figure 1. CTR tool, in source language annotation mode.

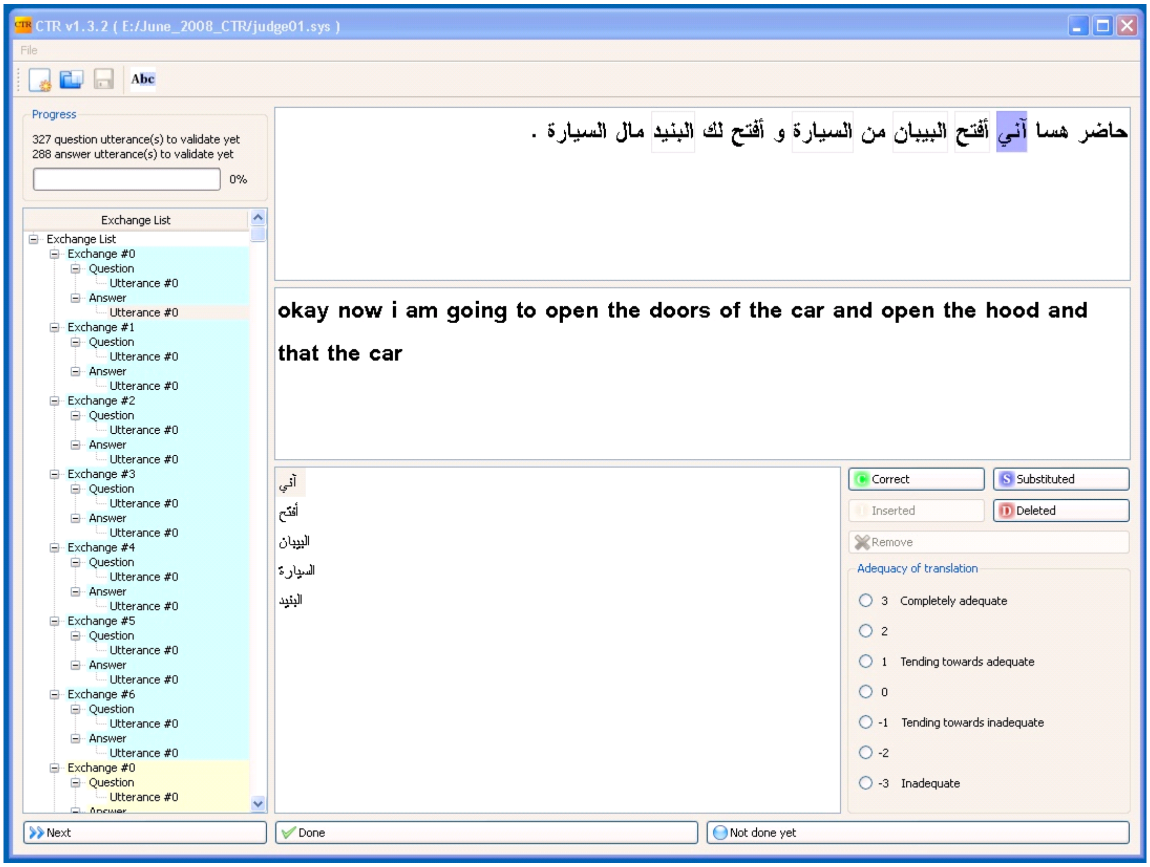
In figure 1, low-level Arabic concepts identified by the annotator appear as a vertical list in the bottom-center of the CTR tool.

No matter what the language, the choices made by the source-language annotator were always reviewed in full detail with the first author (Sanders), in order to ensure that all issues became known and to promote uniformity across languages.

4.2 Target-language Judging

Later (after the evaluation had been run, generating the machine translation outputs from the systems) we had a panel of five bilingual judges compare the machine translation outputs (taken from the system logfiles) to these lists of pre-identified low-level concepts in each source language input utterance. The bilingual judges were all native speakers of the foreign language, reasonably current on the spoken language (not someone who had been in the U.S. for several decades), and had a high level of English proficiency. The judges were given detailed written guidelines for how to do the analysis and were trained by NIST.

The bilingual judges did their analyses using the CTR analysis tool, in an "output scoring" mode. The analyst worked through an utterance at a time. The reference transcription of the spoken input was presented across the top of the screen (much as the Arabic example in the screenshot above), with the machine translatiion output below. The list of pre-identified concepts appeared as a vertical list below that (just as in the screenshot above). The analyst worked down that vertical list of concepts, inputting their judgments using a relevant set of clickable buttons at the right. For each low-level concept, the analyst decided whether it was transferred successfully, was completely missing (a deletion error), or had morphed into something else (a substitution error). The analysts also identified concepts that had been inserted by the system (somewhere in the pipelined combination of ASR+MT). Immediately after scoring the low-level concepts for an utterance, the judge assigns a Likert-type judgment, as can be seen in the set of radio buttons in the lower-right corner of the following screenshot, where the judge is beginning to score an utterance.



CTR v1.3.2 (E:/June_2008_CTR/judge01.sys)

Progress

327 question utterance(s) to validate yet
288 answer utterance(s) to validate yet
0%

حاضر هسا آني أفتح البيبان من السيارة و أفتح لك البنيد مال السيارة .

okay now i am going to open the doors of the car and open the hood and that the car

آني
أفتح
البيبان
السيارة
البنيد

Adequacy of translation

○ 3 Completely adequate
○ 2
○ 1 Tending towards adequate
○ 0
○ -1 Tending towards inadequate
○ -2
○ -3 Inadequate

Figure 2. CTR tool in output scoring mode.

The result was, therefore, a count of correctly transferred concepts and a count of the three types of errors. We computed our metric (Odds of Successful Transfer of a Low-level Concept) by dividing the number of successes by the number of errors. If, for example, there were 100 reference concepts, 75 transferred successfully, and there were no insertion errors, then the odds of success would be 3.0, which would also be the case if 79 transferred successfully and there were 5 insertion errors. In probability and statistics, the word "odds" refers to the odds of the desired outcome — for example, if you roll one six-sided die, the odds of rolling something other than a six are 5 to 1).

Comparisons across repeated evaluations can be stated as an Odds Ratio (the odds of success in the new evaluation, divided by the odds in the previous evaluation), which we believe will be easily understood. To continue our example from the immediately preceding paragraph, if the previous evaluation had 90 of the 100 concepts successfully transferred and there were 10 errors (none insertion errors) then the odds of successful transfer are 90 / 10 or 9.0. If the current evaluation has 95 of the 100 concepts successfully transferred, with 5 errors (none insertion errors), then the odds of success in the current eval were 95 / 5 or 19.0. Of course, odds can be re-stated as a probability, which we call "adjusted probability of correct" — adjProb(*corr*) amounts to

numCorrectlyTransferred / (*totalNum* + *numInsertionErrors*).

5. LIKERT-TYPE JUDGMENTS

Perhaps the most widely accepted benchmark for the quality of machine translation is to get judgments of semantic adequacy from a panel of bilingual judges. "Semantic adequacy" means one asks the judges whether the translation conveys the meaning.

As has been mentioned, immediately after scoring the low-level concepts for an utterance, the judge assigns a Likert-type judgment of semantic adequacy (as can be seen in the set of radio buttons in the lower-right corner of the screenshot in figure 2). Because the judge has just worked through the low-level concept analysis, the judge should have digested the machine translation output and thought about what the *hearer* of that translation would understand it to mean.

In addition, the judges were given a substantial set of exemplars of each of the four Likert categories that have textual anchors, so as to calibrate their expectations to be as harsh/forgiving as the set of exemplars. These examplars came from judgments done by previous panels of judges. In practice, it has been necessary to pay considerable attention to getting all the judges to accept the sets of exemplars as correctly graded, and to not make their own judgments be more harsh or forgiving than the sets of exemplars.

6. ASSESSING OUR METRIC

Our metric is the probability of successful transfer of low-level concepts (content words). In general, the low-level concept transfer metric should be relatively independent of the identity of the source language. We have done what we could to ensure that source-language annotation of low-level concepts was done comparably in all source languages. And of necessity, the source language inputs are different for the different languages, being idiomatic spoken conversations with native speakers of the foreign language, intended to be representative of how U.S. soldiers and marines would use the systems in country.

To assess the quality of our metric, we ask how closely its values correlate with the Likert-type judgments of semantic adequacy. At the utterance level, we have had three different machine translation systems in each evaluation, and we have assessed tranaslations of sixty to eighty utterances. The utterances were input to the systems as audio recordings, so that all three systems would have exactly the same inputs. The translations were judged as explained in the preceding sections of this paper. We then calculated Pearson correlations (Pearson's *R*) between the utterance-level Likert-type judgments of semantic adequacy and the values for probability of successful transfer of low-level concepts in each utterance. from each system. The resulting correlation values between our low-level concept transfer metric and the Likert-type judgments from our bilingual judges are substantially higher than the correlations of those judgments with the typical automated metrics (BLEU, METEOR, and so forth).

For translations into English, the utterance-level correlations between our metric and the Likert-type judgments were usually above $R = 0.7$

For translations into Dari, the utterance-level correlation between our metric and the Likert-type judgments was $R = 0.72$

For translations into Arabic, the utterance-level correlation between our metric and the Likert-type judgments was consistently above $R = 0.8$ (with $R^2 \approx 0.7$) This raises the question as to why the correlations are higher than for the other two target languages. We really do not know why. One possibility is a wider range of difficulty in the source-language data; we don't think that is the case. Another possibility is that it could be the case that for translations into Arabic any low-level concept error has a more direct effect on the actual semantic adequacy of the translation. We suspect that may be true, but we do not know. The Likert-type judgments for translation into Arabic are higher than for any other language pair that we have examined, so it also appears that the quality of these translations is better, and this could also be a factor.

Why are the correlation values above (between our low-level concept transfer metric and the Likert-type judgments of semantic adequacy) not higher? That question is difficult to answer. We think much of the answer lies in the fact that our low-level concept metric gives the same weight to all content words. But some errors are far more serious than others. To provide an aritificial example, suppose a soldier asks a local collaborator about the state of the road to some nearby town. Suppose the collaborator says, "There are new mines under that road." If the automated speech recognition (speech-to-text) component of the system mis-hears this, the translation could easily be, "There are no mines under that road." Our low-level concept transfer metric, (as well as all the typical automated machine translation metrics) will find only one word incorrect and therefore give that translation an excellent score, where a bilingual human judge would recognize that the translation is "inadequate."

7. REFERENCES

[1] Condon, S., Sanders, G. A., Parvaz, D., Rubenstein, A., Doran, C., Aberdeen, J. and Oshika, B. 2009. Normalization for Automated Metrics: English and Arabic Speech Translation. Proceedings of Machine Translation Summit XII, (Ottawa, Canada).

[2] Papineni, K., Roukos, S., Ward, T., and Zhu, W. 2001. BLEU: A Method for Automatic Evaluation of Machine Translation. (IBM research report available at http://domino.watson.ibm.com/library/cybergig.nsf/)

[3] Papineni, K., Roukos, S., Ward, T., and Zhu, W. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In, Proceedings of ACL 2002, 311–318.

[4] Sanders, G., Bronsart, S., Condon, S., and Schlenoff, C. 2008. Odds of ssuccessful transfer of low-level concepts: A key metric for bidirectional speech-to-speech machine translation. In, Proceedings of LREC 2008

Automated Metrics for Speech Translation

Sherri Condon, Mark Arehart, Christy Doran, Dan Parvaz, John Aberdeen, Karine Megerdoomian, and Beatrice Oshika

The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
00-1-703-983-5522

{scondon/marehart/cdoran/dparvaz/aberdeen/karine/bea}@mitre.org

ABSTRACT

In this paper, we describe automated measures used to evaluate machine translation quality in the Defense Advanced Research Projects Agency's Spoken Language Communication and Translation System for Tactical Use program, which is developing speech translation systems for dialogue between English and Iraqi Arabic speakers in military contexts. Limitations of the automated measures are illustrated along with variants of the measures that seek to overcome those limitations. Both the dialogue structure of the data and the Iraqi Arabic language challenge these measures, and the paper presents some solutions adopted by MITRE and NIST to improve confidence in the scores.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics – *performance measures*
I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *machine translation*

General Terms

Measurement, Performance, Experimentation

Keywords

Speech translation evaluation, automated translation metrics, machine translation

1. INTRODUCTION

While human judgments are considered to be the gold standard for evaluating translation performance, it is the development of automated evaluation metrics that has facilitated significant advances in machine translation technology during the last decade. Unlike evaluation methods that involve human judgments, automated measures provide rapid, reliable feedback with relatively low cost. Both human judgments and automated metrics are limited in ways that are still not fully understood, and this report reveals some additional characteristics concerning the application of automated measures to speech translation between English and Iraqi Arabic.

The Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) program has experimented with several evaluation strategies and metrics. Since the inception of the Defense Advanced Research Projects Agency (DARPA) tactical two-way speech translation programs, a MITRE team has coordinated with system developers to design collection methods for training data and evaluation methods to measure progress. More recently, The National Institute of Standards and Technology (NIST) has directed these efforts, and the MITRE team has focused on automated metrics.

The evaluations have focused on the basic functionality of speech recognition and machine translation, and a major goal has been tests that incorporate users and domains which are representative of the military uses for which the systems are designed. Consequently, a significant challenge of developing useful evaluation methods for the TRANSTAC program has been the conflict between replicability and authenticity. Test conditions resembling real-world conditions require spontaneous interaction between representative users with meaningful goals in realistic situations and environments. However, these conditions are not repeatable due to the inevitable variation in human behavior.

The strategy adopted for TRANSTAC evaluations has been to conduct two types of evaluations: live evaluations in which users interact with the translation systems according to several different protocols and offline evaluations in which the systems process audio recordings and transcripts of interactions. The inputs in the offline evaluation are the same for each system, and the automated measures used to evaluate system performance on those inputs produce scores in the same way each time they are computed. Therefore, the same tests can be repeated as the systems mature. Automated measures such as BiLingual Evaluation Understudy (BLEU) [10], Translation Edit Rate (TER) [13], and Metric for Evaluation of Translation with Explicit word Ordering (METEOR) [1] have been developed and widely used for translations of text and broadcast material, which have very different properties than dialogue. The TRANSTAC evaluations provide an opportunity to explore the applicability of automated metrics to translation of spoken dialogue and to compare these metrics to human judgments from a panel of bilingual judges.

The evaluations also offer a chance to study the results of applying automated MT metrics to languages other than English. Studies of the measures have primarily involved translation to English and other European languages related to English. The TRANSTAC data present some significant differences between the automated measures of translation into English and Arabic, and our research has provided some insights into the reasons for these differences.

Approved for Public Release: 09-3822. Distribution Unlimited. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.
PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.

2. AUTOMATIC TRANSLATION METRICS

2.1 The BLEU Measure

A fundamental problem of translation evaluation is that there are many possible translations from a source language input to a target language output. The IBM researchers who developed BLEU in 2001 provided a partial solution to this problem by creating test sets with more than one translation for each input. The machine translation output is then compared to these reference translations, and a score is computed based on the number of n-grams in the output that match the references. For example, Figure 1 provides a sample machine translation from Iraqi Arabic to English along with 4 reference translations.

In Figure 1, 11 of the 12 words in the system output can be matched to words in the reference translations, producing a score of 11/12 for unigram matches. There are 11 bigrams (sequences of 2 words) in the system output, and 5 of them correspond to bigrams in the reference translations: *he has, stomach pain, pain and, and always,* and *pain in*. Therefore, the bigram score is 5/11. The trigram score is 1/10: only *pain and always* can be matched to the references, and there are no matching 4-grams. The BLEU score is computed by micro-averaging [4] the n-gram scores of all the outputs in the test corpus, for n = 1, 2, 3, and 4. Then the geometric mean of the four n-gram averages is computed. Finally, the result is multiplied by a "brevity penalty."

The brevity penalty is assessed because without it, the score would not reflect portions of the reference translations that were completely missed. For example, suppose we added *and I don't know what to do* to each of the reference translations. Without the brevity penalty, the BLEU score would not be affected. The brevity penalty lowers the BLEU score in proportion to the difference between the number of words in the system outputs and the number of words in the reference translations whose lengths are closest to the lengths of the outputs (combined across the entire test corpus).

The example illustrates some of the limitations of the BLEU metric. Although the system output is not fluent English, the meaning expressed in the reference translations is easily inferred from the system output. The BLEU score cannot discriminate between a translation like the system output in Figure 1 and a translation like (1), which has the same number of matching n-grams.

(1) *he has some abdomen and always my and he says in his*

The n-gram matching treats all words equally, regardless of their significance for the meaning. In the extreme case, a semantically loaded word like *not* is treated no differently than an optional conjunction like *and*.

It has been observed that BLEU and measures derived from BLEU have become de facto standards in the machine translation community [7]. As automated measures are used more extensively, researchers learn more about their strengths and shortcomings, which allows the scores to be interpreted with greater understanding and confidence. Some of the limitations that have been identified for BLEU are very general, such as the fact observed earlier that the measure primarily reflects the accuracy of the words that the system produced with only a brevity penalty to assess what the system may have missed. This makes the measure more like a document similarity measure [9]. In fact, researchers often use information retrieval terms to describe this problem. BLEU scores measure *precision*: the proportion of words or documents that were correctly translated or retrieved compared to the total words or documents that were translated or retrieved. BLEU scores do not measure *recall*: the proportion of words or documents that were correctly translated or retrieved compared to the total words or documents that should have been translated or retrieved.

2.2 The METEOR Measure

Researchers have proposed dozens of alternative measures that seek to improve on BLEU, while retaining the basic insight of comparing system outputs to multiple reference translations. Many of these measures were compared in the NIST Metrics for Machine Translation 2008 Evaluation (MetricsMATR08) [8]. In addition to BLEU, the TRANSTAC program uses METEOR to score translations of the recorded dialogues. METEOR incorporates a unigram recall score that can yield higher correlations with human judgments than BLEU scores [1].

METEOR also addresses another problem that has been associated with BLEU. The ability of BLEU to take into account many possible translations for a given segment of language depends solely on the number of reference translations that are available for comparison. In contrast, METEOR accepts synonyms defined in a resource called WordNet [17], allowing additional options that are not present in reference translations. For example, METEOR would recognize the equivalence of *pain* and *ache* in Figure 1. METEOR also uses stemming to remove inflectional affixes that may prevent translations from matching due to minor variation. For example, after stemming, METEOR would match *cries* and *crying* in Figure 1 because they are both forms of the verb *cry*. However, these enhancements are available only for English: there is no equivalent of WordNet for Iraqi Arabic, and Arabic affixes are often ambiguous out of context, making it difficult to stem words accurately.

The METEOR score is computed by aligning the system output to the closest reference translation as in Figure 2. After stemming, *cries* and *crying* are considered a match, as are *saying* and *says*. In Figure 2, three words of the reference translation (in boldface) are not matched to the system output, and three words of the system output (not boldface) do not match the reference translation.

Ref 1: he has some pain in his stomach and always cries and complains about stomach pain

Ref 2: he has some pain in his stomach and he always cries and says I have a stomach pain

Ref 3: he has some stomach pain and always cries saying my stomach hurts

Ref 4: he has a stomach ache and he always cries and says my stomach hurts

System: he has stomach pain and always crying he says pain in stomach

Figure 1: Sample Reference Translations and System Output

Ref 3:	he has	**some**	stomach pain and always	cries		saying		**my**		stomach	**hurts**
System:	**he has**		**stomach pain and always**	**crying**	he	**says**	pain		in	**stomach**	

Figure 2: METEOR Alignment of System Output and Reference Translation

Therefore, recall is 9/12 and precision is 9/12. A weighted F-score (harmonic mean of recall and precision) is computed with a penalty if any of the words have been aligned out of order. Recall is weighted more heavily than precision, though this can be adjusted by the user.

Unlike the BLEU score, the METEOR score for the significantly poorer translation in (1) is lower than for the system output in Figures 1 and 2. For (1), recall is 8/12, precision is 7/12, and then the score is lowered by a penalty that applies to the match of *my* because *my* occurs in a different position in the two sequences.

2.3 The TER, STER and HTER Measures

Another limitation of the BLEU metric is that it only indirectly captures sentence-level properties such as word order by counting n-grams for values of *n* that are greater than one. But syntactic variation can produce translation variants that may not be represented in reference translations, especially for languages that have relatively free word order [2,15]. For example, in the sample in Figures 1 and 2, the word *always* could appear in a variety of positions as illustrated in (2) for reference #4.

(2) a. he has a stomach ache and he always cries and says my stomach hurts (original reference)
b. he has a stomach ache and he cries always and says my stomach hurts
c. he has a stomach ache and always he cries and says my stomach hurts

Although (2b) and (2c) may seem to be slightly less natural, they are certainly acceptable English forms. In other languages, word order is much freer than in English so that 3 or 4 reference translations will provide only a fraction of the options. METEOR allows users to adjust the word order penalty, but each word that must be moved or shifted in order to align with the reference translation is penalized separately. Therefore, when entire phrases in a language can freely occur in several positions, the translation is penalized for each word in the phrase.

The TRANSTAC program has also experimented with the TER metric to measure translation quality. Unlike METEOR, TER allows any number of contiguous words to shift positions in a single move. Computation of the TER score is based on the Levenshtein edit distance measure for string matching [3], which counts the number of insertions, deletions, and substitutions required to transform one string into another. Figure 3 shows how the alignment in Figure 2 would be edited to transform the system output into the reference translation. The deletions and substitutions that transform *he says pain in* into *saying my* could have been aligned differently with no effect on the number of deletions and substitutions.

The edit distance score is usually normalized by dividing the number of edits by the length of one of the strings, which would produce a score of 7/12 in Figure 3. When more than one reference translation is available, the denominator is the average length of the reference translations.

Levenshtein edit distance does not allow for the possibility of aligning words that are out of order, as TER does. TER permits movement of words or contiguous sequences of words in order to align them, and the shifts are counted as edits along with insertions, deletions, and substitutions. With a slightly different reference translation, Figure 4 shows how allowing a shift produces a lower edit distance score. The TER score in figure 4 is 7/13, whereas the Levenshtein edit distance score treats one *he* as a deletion and the other as an insertion, yielding a score of 8/13 for the same pair. (Lower TER scores reflect better performance.)

TER does not recognize synonyms, though the Semantic Translation Error Rate (STER) does use WordNet to align synonyms [17]. Instead, the inventors of TER introduced a variant that requires human intervention: Human Translation Error Rate (HTER). TER and HTER were developed for another DARPA machine translation program, Global Autonomous Language Exploitation (GALE), for which the machine translation evaluation is also conducted by NIST. In order to compute HTER, a human "post editor" edits the system output to produce a new reference translation that is maximally similar to the system output, while preserving the meaning of the reference translation. For example, a maximally similar reference for the system output in Figures 1-3 is (3).

(3) he has stomach pain and always **cries** he says **I have** pain in **my** stomach

Computing TER using the reference translation in (3) results in a score of 4/15 (errors are in boldface), which is a significant improvement over the TER score computed using any of the reference translations in Figure 1. The lower error rate seems appropriate given our intuition that the meaning of the reference translations can easily be inferred from the system output. In contrast, consider the much poorer translation in (1), which is repeated as (4a).

(4) a. he has some abdomen and always my and he says in his
b. he has some abdomen pain and always cries and he says my stomach hurts

Ref 3:	he has	some	stomach pain and always	cries		saying		my	stomach	hurts
System:	**he has**		**stomach pain and always**	**crying**	**he**	**says**	**pain**	**in**	**stomach**	
Edits:		insertion		substitution	deletion	substitution	deletion	substitution		deletion

Figure 3: TER Alignment of System Output with Reference Translation and Edits

Ref 3:	he has	some	stomach pain and	**he**	always	cries		saying		my	stomach	hurts
System:	**he has**		**stomach pain and**		**always**	**crying**	**he**	**says**	**pain**	**in**	**stomach**	
Lev Edits:		insertion		insertion		substitution	deletion	substitution	deletion	substitution		deletion
TER Edits:		insertion		shift$_1$		substitution	[1]	substitution	deletion	substitution		deletion

Figure 4: TER vs. Levenshtein Edit Distance

(4a) receives a TER score of 8/12 when compared to the closest reference translation in Figure 1, which is a higher error rate than the system output's score of 7/12. The HTER score results in an even greater difference between the two translations: compared to the maximally similar reference translation in (4b), the HTER score for (4a) is 5/14 compared to 4/15 for the system output.

The HTER measure does not need to use WordNet or stemming because the human post editor can incorporate synonyms and adjust inflections. Also, unlike human judgments of translation quality, HTER does not require bilingual judges. Monolingual post editors can produce the customized reference translations from a single reference translation. Although the HTER measure appears to be more sensitive than the TER measure, it requires human intervention. Therefore, the significant advantages that automated measures obtain by eliminating the time, expense, and variability of human evaluations are lost. Consequently, the TRANSTAC program has not used the HTER metric.

2.4 More Issues for Automated Metrics

One shortcoming of automated measures of translation quality is shared by human judgments, which are typically obtained by asking bilinguals to rate system outputs on a scale that ranges from poor to perfect. Neither automated measures nor human judgments provide feedback that is diagnostic or that specifies the problems in less-than-perfect translations. In fact, BLEU is designed to be computed on an entire test corpus, using micro-averaging and calculating the brevity penalty based on all of the references and system outputs in the test set. NIST micro-averages the HTER scores when reporting evaluation results for GALE [Le, personal communication]. The claim is often made that automated measures cannot be expected to correlate well with human judgments at a sentence or utterance level: the high correlations that are reported compare corpus-level scores among translation systems so that the statistic is typically based on only a dozen or fewer data points. The MetricsMATR08 evaluation computed both utterance and corpus level correlations, and the former were much lower [9].

Another issue that is relevant to TRANSTAC evaluations concerns the quantity of data required for reliable automated measures. TRANSTAC training data is difficult to collect (see Section 3) so that it is important to hold as little as possible back for evaluation. Fortunately, some recent work suggests that samples as small as 300 sentences can be sufficient to correctly detect significant differences between systems, though bootstrap sampling is recommended to assess the significance of differences in scores [15].

A related concern is the length of the inputs, which has particular importance for TRANSTAC data because spoken utterances tend to be shorter than written ones. For example Turian, Shen, & Melamed report that samples of reference translations from TIDES corpora averaged about 31 words per sentence [15], whereas 30 words is considered a maximum for inputs to the TRANSTAC speech translation systems. All of the automated measures of translation quality have been developed and tested using text data, whereas TRANSTAC data is speech data, which is structured very differently. In the next section the data collected for TRANSTAC systems is described, and additional features of those data that might affect automated metrics are discussed.

3. TRANSTAC TEST DATA

3.1 Data Collection

Initially, TRANSTAC stakeholders agreed that domains and use cases should be narrowly defined in order to provide realistic goals for the speech translation systems. However, it quickly became clear that even the most routine interactions can easily veer out of domain when, for example, the driver at a checkpoint tries to explain why he has a sack of money in the trunk. Interviews with veterans of military operations in Iraq and Afghanistan initially resulted in about 50 scenarios that were used to elicit interactions in 6 domains, including checkpoints, searches, infrastructure surveys (sewer, water, electricity, trash, etc.), and training. Later, another 30 scenarios were developed with more diverse topics such as medical screening, inspection of facilities, and recruiting for emergency service professionals. Eventually scenarios were consolidated into six broad categories: checkpoints, civil affairs, facility inspections, medical, training, and joint operations.

Scenarios provide each role-player with a description that sets the scene, identifies the role of the speaker, provides some background and motivation for the speaker, and may describe an outcome for the encounter. For example, the military speaker might be asked to imagine that he is at a checkpoint, that a car driven by a young man has approached, that a search of the car revealed a large bag of cash in the trunk, and that the man is detained for further questioning. Scenarios included an example interaction or suggested topics for discussion. Role-players were coached to prepare for their roles before recording.

A variety of protocols were used in order to take advantage of role-players available at different data collection events and to maximize the number of interactions that were recorded. Large amounts of Iraqi Arabic data can be collected if Arabic speakers interact in Arabic. For authentic military English, dialogues were recorded in which an American soldier or Marine interacted with an Iraqi Arabic speaking civilian via a bilingual interpreter. This protocol made it possible to obtain a maximum amount of speech from the very limited time that we had access to military personnel. In earlier data collection events, an inoperable telephone handset or similar prop was passed to each role-player before he or she could begin to talk, which minimized overlap among the speakers. Later, lights were used to signal when participants could begin to speak. Additional data were collected by eliciting answers to prerecorded questions from native Iraqi Arabic speakers, and one of these collections was designed to elicit names of people, places, and organizations.

All of the interactions were transcribed orthographically, and the transcriptions were translated into the other language (English to Arabic or Arabic to English) by professional transcribers and translators. Transcription and translation conventions were developed with input from developers, NIST, the Linguistic Data Consortium (LDC), and MITRE. Portions of the Arabic data were transcribed phonetically, and diacriticized lexica were created. Transcriptions included timestamps at the beginning and end of each segment. Some recordings, transcriptions, and translations were not distributed to the developers so that they could be used for evaluation. These data are referred to as the *reserved* data (see section 3.2).

The data collection protocols resulted in speech that differs from the inputs that users produce when interacting with speech translation devices. Users communicating via a speech translation device quickly realize that they must speak clearly, avoid false starts and filler expressions such as ‘uh,’ and keep inputs short and simple. In contrast, the training data resembles ordinary conversation with high frequencies of filler expressions, pauses, breaths, and unclear speech as well as lengthy utterances. Some examples are provided in (5).

(5) a. then %AH how is the water in the area what's the -- what's the quality how does it taste %AH is there %AH %breath sufficient supply?

b. the -- the first thing when it comes to %AH comes to fractures is you always look for %breath %AH fractures of the skull or of the spinal column %breath because these need to be* these need to be treated differently than all other fractures.

c. would you show me what part of the -- %AH %AH roughly how far up and down the street this %breath %UM this water covers when it backs up ?

The examples in (5) illustrate the filler expressions such as ‘um’ and ‘uh,’ which are transcribed ‘%UM’ and ‘%AH,’ and false starts, which are represented by dashes, in the data.

Another source of mismatches between training data and live evaluation inputs is in the transcription. Transcribers were instructed to divide sequences of speech from a single speaker into smaller units at reasonable logical break points. The guidelines indicate that there has been ongoing clarification of this directive, and it is clear that divisions were inconsistently applied. For example, the single segment in (5a) contains four separate questions, and (5b) was divided in the middle of a sentence where the asterisk appears in the text. There can be good reasons not to separate every distinct sentence-like unit in a steady stream of speech. If speakers do not pause between these units, then the speech cannot be divided cleanly due to co-articulation.

3.2 Selection of Evaluation Data

The TRANSTAC offline evaluations have primarily used two types of recorded dialogues. Reserved test data are subsets of the training data that are held back for evaluation instead of delivered to researchers for system development. Although reserved sets can be maximally representative of the training data, they are not ideal test sets because systems have been exposed to the voices and speech patterns of the speakers during training. Therefore, a special data collection using speakers who do not appear in any training data was conducted in order to create a test set that is sequestered for re-use.

Training data were collected, processed, and released as separate corpora based on the data collection events at which they were produced. In order to identify a representative reserved set from each collection, the vocabulary in each dialogue was analyzed to provide the following information:

1. Total word tokens and word types in the dialogue
2. Number of tokens and types that are unique to the dialogue
3. Percentage of tokens and types in the dialogue that occur in other dialogues
4. Number of times a word in the dialogue appears in the corpus: average for all words

From the dialogues that were in the mid-range for the percentage of word types that occurred in other dialogues, reserved dialogues were chosen so that each scenario topic was covered, a variety of speakers were represented, and the score in (4) above was maximized. Approximately 10% of the recordings were reserved.

Before each evaluation event, the sets of reserved dialogues were analyzed, and a summary of information relevant to selecting the test dialogues was produced. This information included the scenario topics, gender of the speakers (most were male), the number of English and Arabic utterances, and information about the lengths of utterances in the scenarios. Selection of specific audio inputs for the offline evaluation requires several passes through the pool of dialogues available for the offline corpus. In the first pass, complete dialogues for the offline evaluation are selected based on the authenticity of the content, the range of scenarios, and the variety of speakers.

From the selected dialogues, individual utterances were identified as candidates for the offline audio inputs. Utterances were selected to satisfy the following goals:

1. Proportions of male and female speakers are similar to proportions in the training set
2. Utterance lengths do not exceed 30 words with preference for 5 - 15 words in length
3. Minimize the frequency of false starts, pauses and filled pauses
4. Avoid utterances that do not preserve structural and semantic coherence
5. Avoid utterances that appear to overlap with other utterances according to the timestamps
6. At least 400 utterances in each language

After an initial pass through the dialogues to select utterances for an initial count, a second pass finalized the choices by eliminating additional utterances that were less desirable according to the criteria, while still preserving the goal of at least 400 inputs per language. In order to preserve the content and coherence of the dialogues, only the worst offenders of criteria 2-4 were excluded. As more data was collected, the number of utterances was raised to 600. The sequestered test set was selected in a similar manner. It includes 810 English utterances and 664 Arabic utterances.

Timestamps were used to segment the audio recordings into a separate clip for each input. In addition, text inputs were produced from the transcriptions of the selected segments in order to provide measures of translation quality that were independent of speech recognition. Consequently, offline evaluations produced a set of results that included speech recognition word error rate (WER) for each language and BLEU, TER, and METEOR translation scores for spoken inputs as well as BLEU, TER, and METEOR scores for textual inputs.

4. CORRELATIONS AMONG MEASURES

Speech recognition performance is important because recognition errors usually result in translation errors. The speech recognition word error rate was measured using the NIST SCLite scoring software, which computes a score derived from Levenshtein edit distance by comparing system recognition outputs to transcriptions of the speech [12]. To address the variation that occurs in speech, NIST modifies the reference transcriptions, replacing each occurrence of an English contraction with the most likely expansion for that occurrence in its context. Further, words such as *gonna*, *wanna, 'em* and *'cause* that represent phonological reduction are replaced by the unreduced equivalent. Compound words that are usually written as a single word are replaced by that form. Hyphenated words are rewritten as multiple words (replacing hyphen by space). Similar re-writes are done to the system output, except that contractions are replaced by an alternation, so that either version can match the reference. The net result of normalizing the system output and reference transcription files is to increase the number of matches (lowering the WER), make fairer comparisons among systems, and increase repeatability.

For each evaluation, a sample of approximately 100 English-to-Arabic and 100 Arabic-to-English translations from the offline test data was also scored using two methods that involved human judgments. In one method, which will be referred to as *Likert judgments*, bilinguals classified the translations as *completely adequate*, *tending adequate*, *tending inadequate* and *inadequate*. More recently these judgments have been modified to the 7-point scale in Figure 5. The same translations were scored using another method, developed by NIST, in which each open class content word (c-word) in the source utterance was identified, and bilingual judges determined whether the word had been successfully translated, deleted, or substituted in the target utterance. The measure, which NIST refers to as *low-level concept transfer,* is computed as an odds score by dividing the number of c-words successfully translated by 1 minus the number of insertions, substitutions or deletions in the target [11].

+3 Completely adequate
+2
+1 Tending adequate
0
–1 Tending inadequate
–2
–3 Inadequate

Figure 5: Seven-value scale for semantic adequacy

Tables 1 and 2 show how the system scores from automated measures correlate with each other, with the human Likert judgments, and with the low level concept scores. Because TER and WER scores are error rates, they are subtracted from 1 to allow a positive correlation. "Concept Odds" refers to the low level concept measure described above, while "%Adequate" is the percent of utterances that were judged completely adequate in the Likert judgments. The correlations are typical of the correlations that developers of automated metrics of translation quality report. They are very high, but are based on only 5 systems and only on the samples of approximately 100 translations for each direction.

Figures 6 and 7 present the scores obtained for each automated measure and each human-judged measure, including the live dialogues. In the latter, military English speakers and Iraqi Arabic speakers were asked to role play scenarios using the translation systems. To maintain consistency in the content of the unscripted interactions as they were repeated for each system, the same speakers were required to obtain and provide the same specific information using each system. Scores were based on a binary human judgment of translation adequacy for inputs produced in 20 ten-minute dialogs [16]. The figures show similar patterns for all of the automated measures and for the human judged measures based on the offline data. The pattern for the live data is somewhat different, but for the most part, systems A-C score higher than D and E.

	BLEU	METEOR	1 - TER	Concept Odds	%Adequate	1 - WER
BLEU	1					
METEOR	0.994	1				
1 - TER	0.994	0.993	1			
Concept Odds	0.955	0.919	0.928	1		
%Adequate	0.969	0.937	0.951	0.994	1	
1 - WER	0.958	0.968	0.974	0.872	0.888	1

Table 1: English to Arabic Pearson Correlations among Measures for January 2007 Systems

	BLEU	METEOR	1 - TER	Concept Odds	%Adequate	1 - WER
BLEU	1					
METEOR	0.974	1				
1 – TER	0.982	0.945	1			
Concept Odds	0.978	0.990	0.972	1		
%Adequate	0.979	0.988	0.930	0.971	1	
1 - WER	0.813	0.906	0.756	0.847	0.880	1

Table 2: Arabic to English Pearson Correlations Among Measures for January 2007 Systems

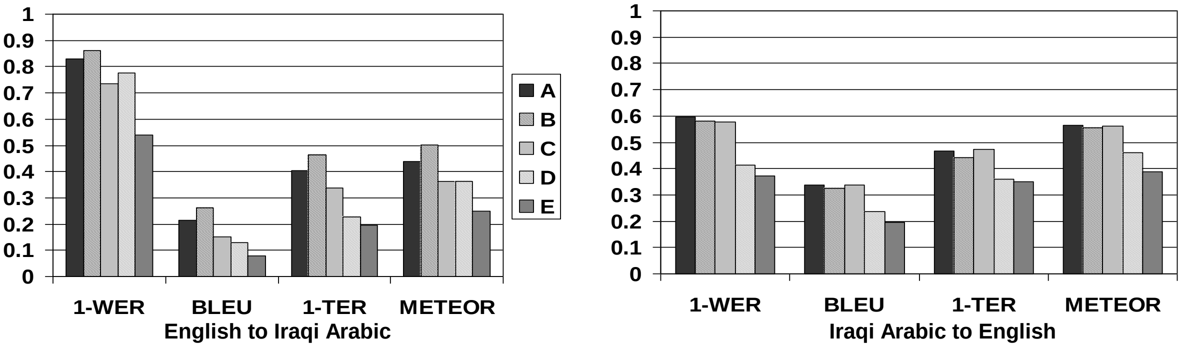


Figure 6: Automated Measures for Translations and Speech Recognition for January 2007 Systems A - E

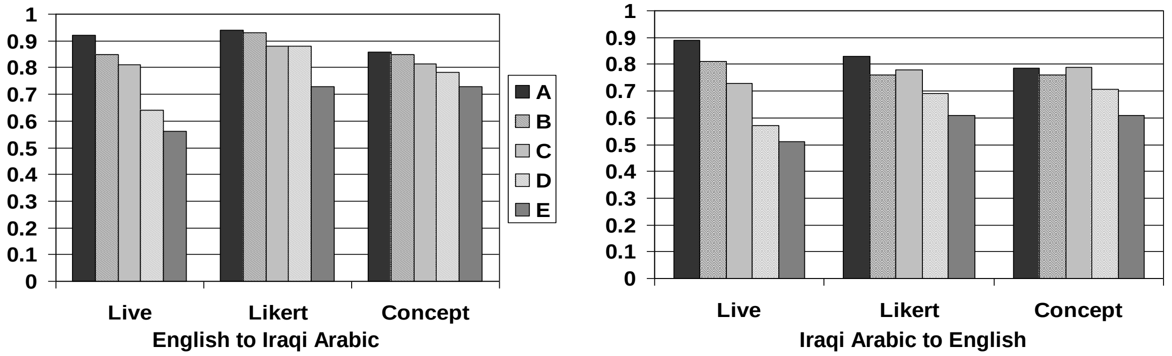


Figure 7: Translation Quality Measures Involving Human Judgments for January 2007 Systems A – E

5. CHALLENGES FROM ARABIC

One fact about the patterns of scores has persisted in subsequent evaluations. Although the human judgments consistently suggest that translation from English to Arabic is more successful than translation from Arabic to English, the automated measures consistently suggest the opposite. Moreover, the WER for English is much lower than for Arabic, which should also make translation more accurate, as suggested by the human judgments, but not the automated measures. It cannot be expected that scores from automated metrics will be comparable across languages, but the concern is that the measures may be less indicative of translation performance for languages like Arabic.

Several properties of Arabic challenge assumptions of automated measures. For example, it is assumed that words can be separated by spaces or punctuation, but six high-frequency words in Arabic, including the equivalents of 'the' and 'and' are attached to the word that follows them in Arabic orthography. The orthography of Arabic is extremely variable, with diacritic elements frequently omitted, so that string matching may fail due to a minor difference that would not obstruct understanding. Also, word order is freer in Arabic than in English. Furthermore, Arabic is more highly inflected than languages like English, though these differences have little effect on meaning. The examples in (6) illustrate that even in the absence of context, errors in inflectional morphology do not prevent communication of the sender's message.

(6) a. two book (two books)
b. Him are my brother. (He is my brother)

BLEU scores computed with reference to the correct versions in parentheses would be very low because the inflected forms do not match. METEOR provides a stemming operation that addresses this problem for English, but for many Arabic strings, complete stemming is not possible because the forms are ambiguous. Instead we experimented with light stemming, which has proven to be helpful in information retrieval tasks [6].

While NIST's normalization of references and outputs for computing WER has been uncontroversial, similar processes had not been proposed for automated measures of translation quality. However, normalization appears to be a simple way of handling superficial variation that would adversely affect accurate scoring of translations, just as it does for scoring WER. The TRANSTAC program has introduced normalization procedures for both English and Arabic to reduce variability before scoring with automated metrics. Norm1 performs rule-based normalization such as replacing contractions with full forms in English and removing all diacritics in Arabic. Norm2 performs word-based normalization such as the spellings of Arabic names in English. We experimented with two consequences of light stemming in Arabic: Norm2a separates the affixes, but does not delete them, while Norm2b deletes the affixes.

We also experimented with an option in the BLEU metric that uses only the unigram scores to allow for the freer word order in Arabic. We used the human judged subset of the June 2008 evaluation consisting of 109 English utterances (1431 words) and 96 Iraqi Arabic utterances (1085 words) in excerpts from 13 dialogs, each including about 7 exchanges. Table 1 provides Pearson's correlations among all the measures we have discussed for the English to Iraqi Arabic translations. Each correlation is computed over 39 data points (scores from 3 systems on excerpts from 13 dialogs). Correlations to the word-error-rate (WER) from

								Likert	Content
	English input	BLEU 1	BLEU 4	BLEU 1	BLEU 4	BLEU 1	BLEU 4	Semantic	Word
	WER Norm2	Norm2	Norm2	Norm2a	Norm2a	Norm2b	Norm2b	Adequacy	AdjProbCor
WER Norm2	1								
BLEU_1 Norm2	-0.23	1							
BLEU_4 Norm2	-0.03	0.81	1						
BLEU_1 Norm2a	-0.33	0.77	0.63	1					
BLEU_4 Norm2a	-0.18	0.81	0.89	0.79	1				
BLEU_1 Norm2b	-0.43	0.82	0.51	0.80	0.61	1			
BLEU_4 Norm2b	-0.38	0.76	0.63	0.64	0.66	0.84	1		
Likert Sem Adeq	**-0.63**	**0.50**	0.19	**0.60**	0.41	**0.75**	0.63	1	
Adj Prob Correct	**-0.67**	**0.35**	0.07	**0.59**	0.30	**0.67**	0.48	**0.86**	1

Table 1: Pearson's *R* Correlations among the Metrics and Normalizations: June 2008 English to Iraqi Arabic

automated recognition of the English speech input are included in the first column. Next are correlations of Norm2, Norm2a, and Norm2b computed with BLEU_1 (BLEU with unigrams only) and with BLEU_4 (the more usual version with unigrams through 4-grams). Correlations with the two human-judgment metrics are highlighted with grey background: "AdjProbCorrect" is based on the low-level concept transfer score described in section 4.

The highest correlation in Table 1 is between the two types of human judgments. Also, it appears that WER is a good predictor of translation quality for the TRANSTAC systems. There is a steady increase in correlation from Norm2 to Norm2a to Norm2b. Norm2b scores correlate with the human judgments considerably more strongly than is the case for the Norm 2 and Norm2a scores. We believe this shows that human judges are more sensitive to errors on content words than to errors on the functional elements that are removed from Norm2b, but are only separated in Norm2a.

6. CONCLUSIONS

This report describes automated measures of translation quality, their limitations, and the issues encountered when applying the measures to speech translation data and to Arabic data. The report contributes to the research community's understanding of these measures, which have significantly advanced the development of machine translation systems. Results are based on a small sample of data, and additional data may change the patterns observed.

7. REFERENCES

[1] Banerjee, S. and A. Lavie. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, pp. 65-73.

[2] Chatterjee, N., Johnson, A., and M. Krishna. (2007). Some Improvements over the BLEU Metric for Measuring Translation Quality for Hindi. In Proceedings of the International Conference on Computing: Theory and Applications 2007, pp. 485-90.

[3] Cohen, W., Ravikumar, P, and Fienberg, S. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks. *IIWeb 2003: 73-78*.

[4] Internet Information. Course Wiki. Accessed August, 2009. http://ilps.science.uva.nl/Teaching/II0607/twiki/bin/view/Main/MeetingFeb22#1_1_1_Averaging.

[5] Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In Proceedings of EMNLP 2004.

[6] Larkey, L. and Connell, M. 2007. Light stemming for Arabic information retrieval. In Arabic Computational Morphology: Knowledge-based and empirical method, A. Soudi, A. van den Bosch, and G. Neumann, Eds,, Springer Verlag.

[7] Lita, L.V., Rogati, M., and A. Lavie. (2005). BLANC: Learning Evaluation Metrics for MT. In Proceedings of HLT/EMNLP, pp. 740–747.

[8] Metrics for Machine Translation Evaluation (Metrics MaTr08). Accessed August, 2009. http://www.itl.nist.gov/iad/mig/tests/metricsmatr/2008.

[9] Owczarzak, K., van Genabith, J., and A. Way. (2007). Dependency-Based Automatic Evaluation for Machine Translation. In Proceedings of HLT-NAACL 2007 AMTA Workshop on Syntax and Structure in Statistical Translation.

[10] Papineni, K., Roukos, S., Ward, T., and W-J. Zhu. (2002). Bleu: A method for automatic evaluation of machine translation. In Proceedings of ACL 2002, pp. 311-318.

[11] Sanders, G., Bronsart, S., Condon, S., and C. Schlenoff, (2008). Odds of successful transfer of low-level concepts: A key metric for bidirectional speech-to-speech machine translation in DARPA's TRANSTAC program. In Proceedings of LREC 2008.

[12] SCLite. NIST Multi-Modal Information Group. Accessed August, 2009. http://www.itl.nist.gov/iad/mig/tools/

[13] Snover, M., Dorr, B., Schwartz, R., Makhoul, J., and L. Micciula. (2006). A Study of Translation Error Rate with Targeted Human Annotation. In Proceedings of AMTA 2006, pp. 223-231.

[14] Subramanian, K., Stallard, D., Prasad, R., Saleem, S., and Natarajan, P. Semantic translation error rate for evaluating translation systems. IEEE Workshop on Automatic Speech Recognition & Understanding, 2007, pp. 390-395.

[15] Turian, J.P., Shen, L. and I. D. Melamed. (2003). Evaluation of Machine Translation and Its Evaluation. In Proceedings of MT Summit 2003, pp. 386-393.

[16] Weiss, B., Schlenoff, C., Sanders, G., Steves, M., Condon, S., Phillips, J., and Parvaz, D. (2008). Performance Evaluation of Speech Translation Systems. In Proceedings of LREC 2008.

[17] WordNet. Accessed August, 2009. http://wordnet.princeton.edu.

Utility Assessment in TRANSTAC: Using a set of complementary methods

Michelle Potts Steves and Emile Morse
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20899-8940 USA
1 301-975-{3537, 8239}

msteves@nist.gov, emorse@nist.gov

ABSTRACT

This paper describes the methods used during formative utility assessments for speech-to-speech, real-time translation systems intended for tactical use. The systems, while still prototypical, had hardware platforms that could be exercised indoors or outdoors by study participants with pertinent backgrounds and skills that are similar to the intended target users. English-speaking and foreign language subjects participated in exercises using the systems exchanging information without the aid of a human interpreter. Feedback on subjects' experiences during the dialog interactions was collected via two primary methods: survey questionnaires and semi-structured interviews. In this paper, we describe our human-centered approach to utility assessment, how we combined the use of two feedback gathering methods, and discuss findings from use of this approach.

Categories and Subject Descriptors

H.5.2 [Information Systems. Information Interfaces and presentation. User Interfaces] Evaluation/methodology

General Terms

Measurement, Performance, Human Factors

Keywords

Speech-to-speech translation systems, evaluation, questionnaire, semi-structured interview, utility assessments, complementary assessment methods

1. INTRODUCTION

This paper describes the approach we have evolved to perform formative utility assessments for speech-to-speech translation systems. It describes briefly what utility is and the techniques used to assess utility for these prototype systems, namely, the combination of two methods: survey questionnaires and semi-structured interviews. Further, we describe how we used these techniques during our most recent evaluation and provide some observations on the use of the approach.

Copyright

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA, Copyright © 2009 ACM 978-1-600558-747-9/09/09…$10.00

2. BACKGROUND

2.1 TRANSTAC Program

The systems being evaluated were developed by research teams participating in the Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) program. TRANSTAC is a Defense Advanced Research Projects Agency (DARPA) advanced technology research and development program[1]. The goal of the TRANSTAC program is to demonstrate capabilities to rapidly develop and field free-form, two-way speech-to-speech translation systems that enable speakers of different languages to communicate with one another in real-world tactical situations without an interpreter. To date, several prototype systems have been developed for force protection and medical screening domains in Iraqi Arabic (IA), Mandarin, Farsi, and Pashto. Systems have been demonstrated on both handheld and laptop-grade platforms with varying performance. Figure 1 shows one such prototype in use during a recent evaluation. Evaluations have focused on assessments of both technical performance data and value to the intended end user.

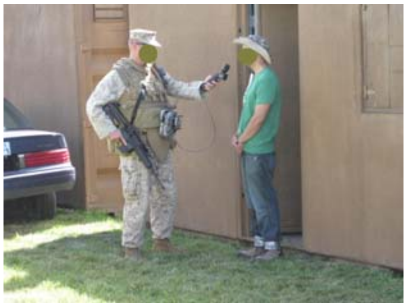


Figure 1: TRANSTAC system prototype

2.2 Utility Assessments for TRANSTAC

Utility is an extension of usability [4] and describes the value that a system provides to the end user; usability, in contrast, addresses whether the system can be used. Utility and usability fall under the general heading of usefulness. For a detailed discussion of utility in the context of testing early prototype systems, see [3]. Utility assessments in TRANSTAC were intended to provide both quantitative and qualitative data about various aspects of these end-to-end systems. The interfaces between the systems and the users comprise both hardware and speech. Users are expected to

[1] The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

use a microphone and speaker/headphones as the hardware and use speech as the interaction mode. Both types of interactions are novel. More usual systems require keyboard input and text understanding.

We believed that user-centered methods would allow us to collect both quantitative and qualitative measures of utility that could drive further development of the TRANSTAC systems.

2.2.1 Techniques Used

Over the course of the last three years, there have been five TRANSTAC evaluations. From the earliest of these evaluations, we have employed questionnaires to elicit opinions and attitudes of the subjects. Instruments have been developed for English speakers who are usually Marines and Soldiers, referred to as Subject Matter Experts (SMEs), and for Foreign Language Speakers (FLEs), who have always been bilingual in English and the target language.

The second type of utility elicitation method that has been used is some form of interview. In early studies, observers often engaged in ad hoc debriefs with individual subjects. As the evaluation series evolved, we structured meetings with small groups (2-4 subjects) of subject matter experts and a single facilitator at one time, and large group meetings (8-10 subjects) with 1-2 facilitators for the foreign language speakers. Additionally, sessions in earlier TRANSTAC evaluations were essentially unstructured, in that the interviewer would rely on his/her recent observations of the subjects engaged in their tasks with the speech-to-speech translation system. More recently we have adopted a process that is referred to as semi-structured interviews [6]. This transition was made possible by our experience over time with the other interview methods and a focusing of areas of inquiry after learning the kinds of feedback subjects provided.

Although the questionnaires that we used provided space for the subjects to make open-ended comments related to an issue, we found that since each subject completed many of these surveys over the course of a week of testing, the quantity of comments decreased and the quality of the content became repetitious. Semi-structured interviews gave the subjects an opportunity to interact with one or more other subjects and with the facilitator; this situation seem to stimulate new ideas and the group tended to become newly motivated and to feed off each other. The structure provided by a core set of questions also allowed different researchers to act as facilitators. This latter aspect was important since several activities were running in parallel throughout the evaluation period and a single facilitator would never have been able to interact with all interview groups without severe changes to the master plan.

2.2.2 Method

This section gives an overview of the testing procedures used in the most recent study. Prior evaluations had similar procedures.

SMEs and FLEs were trained just prior to the testing period. Training for both groups included an overview briefing on the program and its goals, the assessment goals and procedures for this testing period, administrative topics, and scenario topics. During scenario topic review, SMEs and FLEs were given some background information and suggested talking points for each of their assigned scenarios. Additionally, they were given the opportunity to step into their respective scenario roles to practice and try out various talking points with members of the evaluation team prior to interacting with a TRANSTAC system. Additionally, SMEs were provided TRANSTAC system-specific training just prior to a testing period. Four days of testing followed the training day. See Figure 2: Overall schedule.

Day 1	Day 2	Day 3	Day 4	Day 5
SME and FLE Training	SME system training	SME system training	SME system training	SME system training
	AM: testing	AM: testing	AM: testing	AM: testing
	PM: testing	PM: testing	PM: testing	PM: testing

Figure 2: Overall schedule

Figure 3: Daily testing schedule example shows a more detailed schedule of activities for subjects interacting with one research team's TRANSTAC system, e.g., Team C, for the highlighted areas in Figure 2. Each of the other 2-3 research teams, their associated prototype systems, and assigned subject participants had similar schedules each of the 4 days of testing.

Tuesday						
TRANSTAC Research Team C						
Start	**End**	**Station**	**Activity**	**SME**	**FLE**	**Obs.**
8:00	8:45	N/A	Morning Briefing	ALL	ALL	
8:45	10:00	N/A	Sys. Training	7, 8, 9	n/a	
8:45	10:00	N/A	Scenario Review	as needed	ALL	
10:00	10:40	Station 3	Scenario 3	7	7, 8	C
10:40	11:20	Station 1	Scenario 1	8	1, 2, 3	C
11:20	12:00	Station 2	Scenario 2	9	4, 5, 6	C
12:00	12:45	Area 3	Semi-Struct. Int	7, 8, 9	n/a	C
12:00	12:45	Area 4	Semi-Struct. Int	n/a	ALL	D
12:45	13:45	N/A	LUNCH			
13:45	14:25	Station 3	Scenario 6	8	7, 8	A
14:25	15:05	Station 1	Scenario 4	9	1, 2, 3	A
15:05	15:45	Station 2	Scenario 5	7	4, 5, 6	A
15:45	16:30	Area 3	Semi-Struct. Int	7, 8, 9	n/a	A
15:45	16:30	Area 4	Semi-Struct. Int	n/a	ALL	D

Figure 3: Daily testing schedule example

3. TECHNIQUE IMPLEMENTATION

The next section describes the detailed questionnaires that were developed for quantitative and qualitative subjective feedback. After that we describe the range of interview methods that were used in this study. Generally speaking, there are three main opportunities in a study for using questionnaires [1],[4] – pre-test, post-task, and post-test. We administered a pre-test survey to gather demographic data about our subjects; the content of the pre-test form is not shown here. The details in section 3.1 refer specifically to the post-task survey. SMEs completed a post-task survey roughly 8 times over the course of the 4-day evaluation; whereas the Foreign Language Speakers completed the post-task survey approximately 24 times during the 4-day evaluation period. The FLEs participated in more interactions, although typically two thirds of their interactions were of a secondary nature. We expected the observations of those who participated in a secondary manner to also be of value. There was no post-test survey in this study, but rather we used interview techniques to elicit the subjects' feedback after all tasks had been completed.

3.1 Survey Questionnaires

Questionnaires were administered to English speakers and to foreign language speakers. During the first TRANSTAC evaluation exercise, our subjects filled in paper surveys. All subsequent studies were done on-line. The switch to on-line had several advantages: 1) elimination of data errors due to post-study transcription; 2) ability to enforce submission of data in required fields; and 3) subjects appeared to prefer interacting with a system rather than with paper. The first subsection describes survey items that produced quantitative data. The section after that addresses qualitative, open-ended comments.

3.1.1 *Quantitative assessment areas*

There were several areas that we wanted to probe and each category is detailed in the following subsections. Related sets of questions were administered in the order shown. Best practice in questionnaire design recommends that instruments be organized so that the most important information is elicited first and that more detailed information is requested later. Getting the most important things first enhances the likelihood of getting a subject's freshest impressions. Asking detailed questions later tends to prevent a subject from being bored. Five-point Likert scales were used for each test item. Except for grouping related items, no headers, labels, or other indicators were used in the questionnaires themselves.

3.1.1.1 *Interaction factors*

The questions in this section are related to general usability of the speech-to-speech translation. We wanted to know at a high-level how the subject felt about the general context of the just-completed task. While some of the items are highly correlated with one another, analysis of the data shows that subjects discriminate the nuanced meanings of the items. For example, on average, subjects might give high ratings to a system on the first three questions but significantly lower ratings for other questions.

- I found the system easy to understand in this interaction.
- What the system said made sense to me.
- The system made it easy to have this interaction.
- In this interaction, it was easy to get the information I needed.
- I knew what I could say or do at each point of this interaction.
- The system worked the way I expected it to in this interaction.

All of the questions had response choices of 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A using a scale of 1 to 5.

3.1.1.2 *Speech characteristics*

Each characteristic was assessed by the SME with respect to English and the FLE with respect to the other language being assessed. The following characteristics were assessed:

- The pronunciation of the <*language*> was clear and understandable. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*
- The <*language*> words were put together in a way that was coherent and comprehensible. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*
- When the system didn't understand me, it was easy to correct. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*
- How widely will the <*language*> spoken by the system be understood by other <*language*> speakers? *[Response choices were 5=Nearly all, 4=Most, 3=Many, 2=Few, 1=Almost None, and N/A, using a scale of 1 to 5.]*
- How easily will other <*language*> speakers be able to speak in a way that is understood by the system? *[Response choices were 5=Effortlessly, 4=With little effort, 3=Some effort, 2=With difficulty, 1=Nearly impossible, and, N/A using a scale of 1 to 5.]*
- How appropriate is the <*language*>spoken by the system to the situations simulated in today's tests? *[Response choices were 5=Completely appropriate, 4=Rather appropriate, 3=Minimally acceptable, 2=Somewhat inappropriate, 1=Completely inappropriate, and N/A, using a scale of 1 to 5.]*

3.1.1.3 *Satisfaction ratings*

Satisfaction, along with efficiency and effectiveness, are the primary measures of usability [2]. In some sense all questions on a survey are trying to assess a user's subjective satisfaction. This set of questionnaire items particularly targets the 'desirability' aspect of the user's interaction with the system. The first question regarding the speaker's confidence was asked only of the native English speakers, while the other three questions were asked of all subjects. Since the English speaker was viewed in earlier TRANSTAC evaluations as the primary user of the system, it made sense to concentrate on his/her opinion about confidence. We have come to see that the opinions of the foreign language speaker diverge from those of English speakers and provide alternative viewpoints. Therefore, questionnaires in future evaluations will ask both speaker types about their confidence.

- How confident were you in the system's ability to help you communicate effectively? *[Response choices were 5=Strongly Confident, 4=Confident, 3=Neutral, 2=Doubtful, 1=No confidence, and N/A, using a scale of 1 to 5.]*
- I would use this system in the field in its current state of functionality. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*
- Based on my experience in this interaction, I would recommend this system for future similar interactions. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*
- If the system had worked as intended in this interaction, I would recommend this system for future similar interactions. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*

3.1.1.4 *Problem detection and categorization*

Subjects were asked to characterize the causes of any problems they experienced during each interaction. This was assessed with the following question: If you experienced problems in the last dialog-interaction as reported above, please indicate the cause(s) of these problems. *[Response choices were "Translation system problems", "Human partner problems", and "Other, please describe".]*

3.1.1.5 Changing speech to match the system

When we observed subjects using the speech-to-speech systems, it was clear that they tried a variety of ways of adjusting their speech input, especially when their utterances were not producing good results from their perspective. We asked our participants to tell us how they tried to deal with the system by changing their speech patterns. The list of possible changes was based on our observations and things subjects mentioned during interviews.

- Towards the end of the conversation, I was interacting with the system in a different manner than at the beginning. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*
- If you changed the way you interacted with the system, it involved the following (check all that apply):
  - □ Speaking more quickly
  - □ Speaking more 'naturally' for me
  - □ Speaking more slowly
  - □ Speaking more clearly
  - □ Speaking more loudly
  - □ Speaking more softly
  - □ Waiting more time to speak once my 'speech button' is pressed
  - □ Waiting less time to speak once my 'speech button' is pressed
  - □ Using simpler and or shorter sentences phrases
  - □ Using longer and or more complex sentences phrases
  - □ Other (Please describe)

3.1.1.6 Miscellaneous

Except as noted above, both English speakers and bilingual speakers were asked the same questions. One aspect of the test plan that differed for these two types of subjects was the training that people received. Since the English speakers were imagined to be the ones who would carry the physical system in actual use situations, they needed to be trained how to operate the physical device and given guidance on how to place it on their person. Training was provided by the developers of a system immediately before first use on each day that the system was tested.

Foreign-language speakers played the role of citizens of their native country who might need to engage in conversation with the device-carrying English speaker. In such a scenario, there is no reason to provide training to these people. It would be important, however, for the foreign-language speaker to understand how he should act in the conversation. This information was provided by a set of instructions in the foreign language that the English speaker could invoke to be played from the TRANSTAC system for his intended conversational partner.

The survey for foreign-language speakers had a five-point Likert item that addressed the adequacy of the instructional material.

- The system instructions provided a clear understanding of how to use the system. *[Response choices were 5=Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree, and N/A, using a scale of 1 to 5.]*

3.1.2 Qualitative assessment areas

The answers to all the questions in section 3.1.1 were entered by the subject using a radio button set or a drop-down list. The questions in this section are those that presented the user with a text box that accepted open-ended input.

- What did you like most about this system, or what was most helpful and useful?
- What did you like least about this system?
- What would you change about this system?
- What situations can you imagine that might best match this system's capabilities?
- If the translation system seemed to have problems with certain words or phrases, please list any you can recall.
- If you changed the way you interacted with the system, it involved the following "other" [strategies]:
- What would you suggest to improve the training you received before starting the interactions?

3.2 Interview topics

Interviews were done each time that an English speaker completed a series of interaction scenarios using a single system. The schedule of activities was constructed so that two identical systems were in use by different English speakers. This situation allowed us to interview pairs of speakers who had had experience with the same system under very similar conditions. The next section shows topics that each interviewer covered during the interview session. Topics for FLEs follow in the next section.

3.2.1 SME Topics

- Things SMEs liked
- Things SMEs didn't like
- Things SMEs would change
- Overall reliability and robustness (software/hardware) impressions
- Form Factor Aspects
  - Microphone Use
  - Sound Output
  - Appropriateness
  - Comfort
  - Control(s)
- Speech Quality Aspects
  - Communication Successes/Failures
  - Speed Perception
  - Accuracy Perception
  - Error Issues
- Use of confirmation
- Frustration: SME and FLE
- System feedback
- Training/Instruction
- Any other feedback

3.2.2 FLE topics

Foreign language speakers are not government employees but are considered members of the general public from a legal perspective. This fact complicates the use of ad hoc questions during evaluations. Due to restrictions imposed by the requirement for NIST to adhere to the Paperwork Reduction Act (PRA), all questions that NIST researchers would like to ask subjects from the general public must receive approval through a process managed by the Department of Commerce. The questions on surveys described in Section 3.1 have become standardized over time, it was possible to obtain permission to use those survey items. However, since interview topics are of a more ad hoc nature, it was not possible to get timely approval. Therefore, the only question we could pose to the foreign-language speakers during this evaluation was one requesting any feedback.

4. DISCUSSION

Each method has its own strengths. Questionnaires provide opportunities to gather both quantitative and qualitative feedback. Additionally all topics of inquiry were presented to each subject for feedback in the same way after each interaction and subjects could express their views without peer scrutiny. Interviews provide a way to modify the topics of inquiry as new points of interest arise. The interview facilitators had been scenario interaction observers, and were able to make just-in-time modifications to feedback topics relevant to interaction issues the participants had just experienced. Also, facilitator prompting and the group dynamics promoted deeper topic investigation and kept interviews fresher and less repetitive for participants.

The use of complementary techniques gave us the opportunity to gain insights regarding the participant experience that each data gathering technique, used in isolation, would not have provided. In past evaluations, we attempted to use the qualitative responses from the surveys to provide context for the quantitative results. To some extent this was possible, but was completely dependent on the quality of the feedback provided by subjects prompted by the survey questions. We found that the semi-structured interview data greatly enriched the qualitative feedback we received from participants. This, in turn, provided a more informed context to interpret the quantitative findings.

A striking example of how the semi-structured interview data provided a rich context for understanding quantitative results surfaced during the last evaluation. This example has to do with differences between SME and foreign language users regarding interaction factors for each of three systems assessed. During the course of the series of evaluations, we have seen that FLEs were often much more critical of the technologies than were the SMEs. This observation has held over time and has been an on-going question of interest for us. Figure 4 shows a graph of SME and FLE quantitative responses from a recent evaluation to our group of interaction factors questions listed below (also in 3.1.1.1).

- I found the system easy to understand in this interaction.
- What the system said made sense to me.
- The system made it easy to have this interaction.
- In this interaction, it was easy to get the information I needed.
- I knew what I could say or do at each point in the interaction.
- The system worked the way I expected it to in this interaction.

All responses are on a five-point Likert scale, with 5 being the best response. SME responses are in blue and FLE responses in red as graphed. The general trends represented here illustrate the differences of opinions between the two major groupings of participants.

Qualitative data from the questionnaires and FLE interviews indicated FLEs had significant dissatisfaction with the sentence construction, verb plurality, dropped words and concepts, among other issues, for all of the systems involved. The SMEs who possessed some knowledge of the target language were aware of these issues for the FLEs and noted them both in qualitative responses to survey questions and the interviews, however, the systems worked fairly well for the English speakers. On the surface, this seemed to explain the discrepancy between SME and FLE ratings for the systems. However, further probing by interview facilitators uncovered that SMEs are so keenly aware of the need for technology-assisted translation in the field that they were willing to take a chance on the current prototype systems. FLEs, on the other hand, being conversationally fluent in both languages, had no such real-world need and therefore gave a less biased judgment of system performance. In this case, the interview data provided a much richer context for understanding the quantitative results.

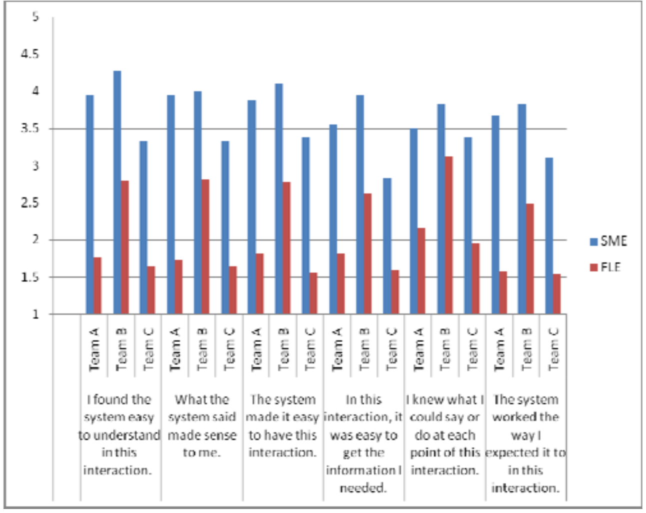


Figure 4: SME and FLE responses to interaction factors questions

Additionally, the interview feedback was much richer than the qualitative data provided in survey responses. We believe some of this is impact from method (discussed above) and some of it may be particular to our subject groups. For example, especially rich feedback was provided by the foreign language speakers from the one open-ended question posed: *Any other feedback*. The FLEs had high self-reported comfort levels with writing in English and use of computers. Yet despite this, verbal feedback was significantly richer. After a bit of reflection, this is not surprising… asking someone to respond verbally in a foreign (2nd, 3rd or 4th) language vs. asking someone to write in that foreign language on a computer, of course, verbal responses will typically be richer. For example, during the interviews, FLEs introduced and discussed topics relating to how the technologies might be received in their native cultures and other cultural impact on expected concepts of operation and system form factors. These topics were not raised by FLEs in qualitative survey feedback. Similarly, Soldiers and Marines were more willing to discuss implications of system issues verbally than in writing. Likewise, they performed more brainstorming regarding system improvements during interviews than in survey feedback.

Finally, the ability to modify topics of inquiry during the evaluation is a large bonus that the interview method provides. In the most recent evaluation, all systems used several forms of

(English) confirmation. Only a small group of survey questions had been selected prior to the evaluation regarding back-translation for inclusion in the survey instrument. The semi-structured interview method provided the ability to probe the SMEs regarding all forms of confirmation and was very useful in providing pertinent feedback to the research teams on this topic of interest.

As with all assessment efforts, there is a cost. In this case, adding the semi-structured interviews required additional time and effort, both in facilitator time to observe each of the interactions, and also time by both facilitators and subjects to participate in the interviews. We felt the time was well-spent, given the current stage of development of these systems, where utility is of interest as well as technical performance.

5. CONCLUSION & FUTURE DIRECTIONS

Early evaluations in this series focused heavily on technical performance of the systems. This was appropriate given the maturity level of the prototype systems and the understanding that insufficient speed and accuracy of translations would dominate any utility feedback during early stages of development. In accordance with this, we conducted data gathering using the survey questionnaires and ad-hoc interviews to provide some feedback to the research teams while refining our topics of inquiry and question sets.

As the systems improved in translation speed and accuracy (technical performance) and form factor, a shift was made towards more emphasis on utility for the intended end user. When this occurred, semi-structured interviews were added to the post-task feedback gathering methods. While increasing the time required to gather utility feedback somewhat, the additional utility data provided new insights into the quantitative data gathered via the survey questionnaires and richer qualitative data overall. The use of these complementary methods improved the quality of the assessments that were collected and drawn from the data, and therefore provided richer feedback to the research teams to help direct their efforts in improving their systems as well as programmatic feedback on intended-user propensities relating to utility and acceptance.

6. ACKNOWLEDGMENTS

We especially want to thank the participants of these evaluations who strive to help move these systems toward deployment with their willingness to work with prototype systems and thoughtful feedback.

7. REFERENCES

[1] Dumas, J.S. and Redish, J.C. 1999. A Practical Guide to Usability Testing. Intellect, Portland OR.

[2] ISO/IEC. 9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 14 Menu dialogues, ISO/IEC 9241-14: 1998 (E), 1998.

[3] Schlenoff, C., Steves, M. P., Weiss, B. A., Shneier, M., and Virts, A. 2007. Applying SCORE to field-based performance evaluations of soldier worn sensor technologies: Field Reports. *J. Field Robot.* 24, 8-9 (Aug. 2007), 671-698.

[4] Scholtz, J. 2006. Methods for evaluating human information interaction systems. Interacting with Computers, Vol. 18 (4), July, 507-527.

[5] Tullis, T. and Albert, B. 2008. Measuring the User Experience. Morgan-Kauffmann, Burlington, MA.

[6] Wood, L.E. 1997. Semi-structured interviewing for user-centered design. Interactions: 4(2): 48-61.

Performance measurement and its role in advancement for intelligent systems: discussion points

Danil Prokhorov* and Yasuo Uehara
Toyota Research Institute NA
1555 Woodridge Dr.
Ann Arbor, MI 48105, USA
*dvprokhorov@gmail.com

ABSTRACT

This brief paper discusses the main theme of the PerMIS 2009 Workshop. It overviews three major areas of autonomous intelligent systems and evaluates different issues of their performance testing.

Keywords

Autonomous intelligent systems, UGV, UAV, home robot

1. INTRODUCTION

The main theme of the PerMIS'09 Workshop is "Does performance measurement accelerate the pace of advancement for intelligent systems?" This simple question apparently may not have an equally straightforward answer.

The simple answer is no. Performance measurement is just a process to evaluate a system, and it is supposed to be unbiased with respect to any higher-level intention, including the intention to accelerate the pace of advancement for intelligent systems (IS).

However, the answer may also be yes because the higher-level intention is self-evident as far as IS advancement goes (one needs constantly advancing IS for accomplishing ever more complex missions without human intervention), and we need to be able to measure the progress of such advancement.

If the reader agrees with the affirmative answer (or at least, admits that "yes" is more useful than "no"), then the reader is welcome to proceed with reading further.

We change the question above slightly, hopefully without loosing its value and importance, and turn it into a more general discussion about the role of performance measurement in advancement for intelligent systems. Our focus is on autonomous IS as exemplified by Unmanned Aerial Vehicles (UAV), Unmanned Ground Vehicles (UGV) and house robots. We do not consider unmanned systems which operate under human operator control as in telerobotics (e.g., when a UAV is flown by a team of human operators).

The next section introduces examples of performance measurement for autonomous IS. Section 3 summarizes our opinion about performance testing, including the use of simulators, followed by our final thoughts in Conclusion.

2. SPECIFICS AND EXAMPLES OF PERFORMANCE MEASUREMENT FOR AUTONOMOUS IS

Not every performance measure helps to accelerate the progress of IS. Performance measurement seems to be only useful for this purpose if it allows us to evaluate decision making capabilities of IS. For example, just measuring maximum speed of an intelligent vehicle implies little about its decision making capabilities, except maybe that the vehicle needs some computing time to make appropriate decisions. It is therefore our opinion that, instead of performance measurement, we should employ performance test (a sequence of appropriate measurements in a form of scenario or procedure).

Performance test may or may not include components which allow us to understand how to advance IS to the next level of intelligence. The Turing test is well known among proposed tests for intelligence. The original test itself is probably not useful *for autonomous IS*, however its automotive variation has been suggested in [1] which is relevant to a class of autonomous IS. We briefly discuss below other tests in the context of autonomous flying, driving and home robots and their corresponding performance metrics.

To be useful for advancing IS, performance tests should contain clear directional information, i.e., information on which components to improve in order to achieve the goals of the test. For example, recognizing objects of interest with the same accuracy as that of the trained human observer is essential to passing the test in any target recognition challenge. Matching or exceeding human recognition accuracy constitutes an obvious direction to advance IS because humans often expect higher performance from robots than from themselves (see, e.g., [2]).

2.1 Autonomous Driving

Though certainly not yet at the human level of driving, autonomous surface vehicles have already demonstrated impressive achievements by advancing in just three years from the level of DNF (did not finish) to the undeniable success as evident from the series of DARPA Grand Challenges [3] (Figure 1). Such challenges are examples of performance tests, which greatly facilitated technology development for

PerMIS'09 September 21-23, 2009, Gaithersburg, MD, USA.
Copyright 2009 ACM 978-1-60558-747-9/09/09 ...$10.00.

autonomous driving. The next logical steps are autonomous driving in less restricted and more complex environments and races against human drivers.

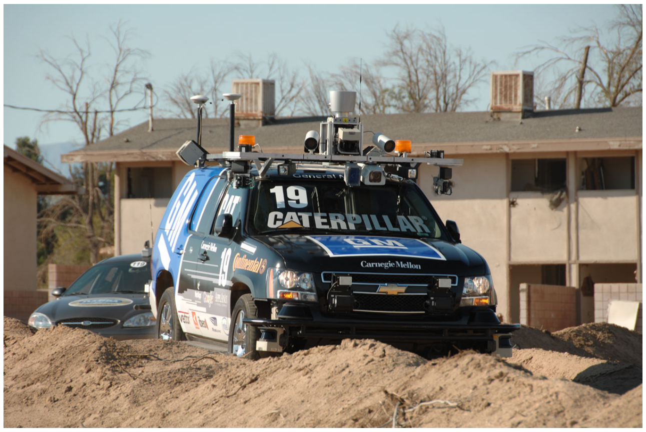


Figure 1: Boss, the robot of the Tartan Racing team is in action during the DARPA Urban Challenge.

In terms of durability (not getting tired, distracted, etc.), the autonomous robots will be able to challenge humans even sooner than their designers usually want to admit. Indeed, autonomous driving across the NA continent will simply not be possible for a single human driver because of the need for rest, while a robot can drive essentially continuously (except for short stops for fueling/electric battery replacement). It is also interesting to ponder whether a conservatively driving robot (one which is slow and stops often, letting other non-robotic vehicles pass it, etc.) can still beat a human driver which needs to rest in a race across the continent – a modern version of the tortoise and the hare fable. By the way, this example is another case in which performance metrics is less of a typical one which measures speed of motion and other physical quantities, but emphasizes safety.

2.2 Autonomous Flying

Similar to autonomous ground vehicles, autonomous flying vehicles have been making impressive advances as exemplified by aerial robotic competitions [4].

It is well known that many existing planes fly on autopilot majority of time, especially during long flights. Even take off and landing can be accomplished by robots on their own (Figure 2). In fact, this unsophisticated level of automation is already greater than what is currently available in the automotive world. Such automation level might be approachable in highway driving for automotive vehicles equipped with next-generation adaptive cruise control (ACC) and lane keeping assist (LKA) systems. Many people would have an issue calling such automatic systems intelligent, for even specialized intelligence is generally believed to imply something much more sophisticated than ACC/LKA.

2.3 Home Robots

Similarly, mobile robots with manipulators such as the one shown in Figure 3 may some day reach the level at which they can measure up against humans in performing various useful tasks. Of course, prior to challenging humans in everyday tasks, home robots should pass much simpler tests of reliably navigating in home environments, if only for the purpose of re-charging their own batteries, as well as finding and fetching appropriate objects based on their verbal descriptions, doing dishes without breaking them, etc. Various universities are pursuing their in-door robotic research programs (e.g., [5]). New robotic challenges are contemplated [6].

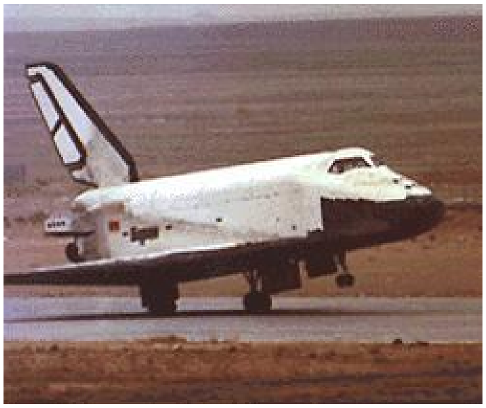


Figure 2: The Soviet Space Shuttle "Buran" touching down automatically upon returning from space in its first and only test flight in 1988.

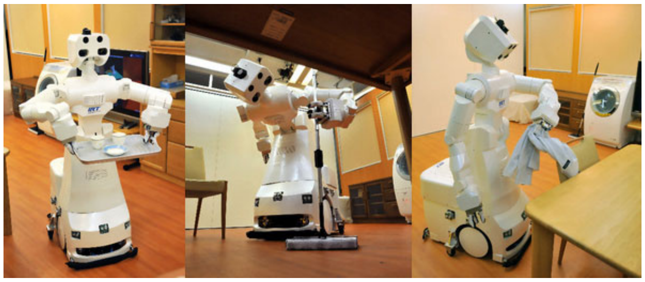


Figure 3: The home robot under development by Toyota and University of Tokyo.

3. COMMON ISSUES OF TESTING AUTONOMOUS IS

UGV, UAV and home robots generally operate in quite different environments. UGV in the form of civilian robotic cars will essentially operate in 2D (i.e., on the ground "plane") which is a safer environment than 3D operational environment of civilian UAV, especially when contrasted with the risk of falling from the sky due to a mistake or a malfunction. However, robotic cars will have to deal with broader variety of situations on the road than their counterparts flying in 3D. Indeed, it is usually easier to collide with something on the road than when flying due to higher density of potential obstacles in 2D. Though formally also operating in 3D, home robots will have their own operation specifics, with even greater variety of situations to handle than those of robotic cars.

Yet, there are common issues of performance testing which will apply to all of them. Several common issues are listed below:

1) For autonomous IS, the performance is probably best to be tested against humans whose functions such IS are

supposed to replace. It seems most useful to do so in a succession of tests of increasing complexity, which will naturally facilitate the pace of IS advancement. As mentioned above, the specific performance measurements such as average speed en route may or may not be of essence for the test when contrasted with the IS operation safety.

2) An appropriate quantifiable metrics should provide a clear indication of how well the IS accomplished the test (e.g., the deviation from a planned route, the total number of missed targets, etc.) [7]. Each test will certainly have specific metrics to measure the degree of success, but it remains to be seen what quantifiable metrics to use when comparing with human-like behavior. For example, expert human operators can always tell the difference between them and a less skilled operator but their assessment that something just does not feel right is often hard to quantify. Moreover, operation differences such as feeling not-human-like must not influence the performance assessment of the IS if they are not essential to the chosen performance metrics.

Conceptually, imitating a human operator is easy: observe inputs to the system under control of a human operator and match them with the system outputs using an appropriate machine learning method as part of the IS. This may or may not be so easy in actual implementation, depending on the control problem specifics. For example, like a human operator any machine learning method must generalize to similar but not exactly the same system conditions as observed during its training to imitate human behavior. This is usually achieved by providing sufficient variety of training examples from a human to the IS, but what constitutes sufficiency and how to guarantee adequate generalization are subjects of on-going machine learning research. Sometimes it is more effective to approximate a human evaluation (“reward”) function than a human controller itself from a modest number of human demonstrations via inverse reinforcement learning (see, e.g., [8]).

3) Performance testing in high-fidelity simulators is worthwhile to supplement testing in real-world environments, especially in early stages of IS development (e.g., simple scenarios, IS subsystem development, etc.). For example, there is not much point in risking an expensive UAV if its attitude control system has not been tested thoroughly in high-fidelity simulations.

Simulators should also be very helpful to test the behavior of autonomous IS in rare/unusual situations because 1) their very low rates of occurrence greatly complicate verification of the IS performance in the real world, and 2) rare situations test for the IS ability of “exception handling” – the quality usually attributed to human intelligence. For example, air or road accidents are rare events, but they can be modeled if we understand what has happened and could describe it in computable terms. Such events as accidents are not only rare but also prohibitively expensive, which makes their high-fidelity simulation even more valuable as a tool for IS performance assessment. The reader is referred to [9] and [10] for examples of IS development simulators.

4. CONCLUSION

We think that performance measurement does help to advance the pace of IS development, especially if performance tests are structured to have gradually increased level of difficulty and to compare different IS against each other and against humans in tasks which humans would otherwise do anyway.

We briefly discussed three classes of autonomous IS: UGV, UAV and home robots. While seemingly different, they all share the same set of performance testing issues as discussed in Section 3. The main systems and subsystems of these IS may be first subjected to tests in high-fidelity simulators, followed by initially simple tests in real-life environment. Gradually, tests will become more complex, with the purpose of verifying more advanced functions of IS. It is in this progression of tests where we see the opportunity to accelerate the pace of IS advancement.

5. REFERENCES

[1] S. Kalik and D. Prokhorov. Automotive Turing Test. In *Proc. PerMIS’07*.

[2] R. Murphy and B. Argrow. UAS in National Airspace System: Research Directions. Final Report of the NSF/AUVSI/FAA/DHS Wokrshop. *Unmanned Systems*, June 2009, pp. 23–28.

[3] `http://www.darpa.mil/grandchallenge/index.asp`, DARPA Urban Challenge 2007.

[4] `http://www.auvsi.org/competitions/`, AUVSI Student Competitions.

[5] `http://stair.stanford.edu/`, STAIR: STanford Artificial Intelligence Robot.

[6] C. Jenkins. The Future of AAAI Robotics. Challenges and directions for building upward. `http://robotics.cs.brown.edu/ijcai09/aaai_challenges_mar2009.pdf`

[7] E. Messina, A. Jacoff, and H. Scott. Performance Evaluation of Autonomous Mobile Robots. In R. Madhavan, E. Messina, J. Albus (Eds.), *Intelligent Vehicle Systems: A 4D/RCS Approach*, pp. 247–282, Nova Science Publishers, NY, 2006.

[8] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. International Conference on Machine Learning*, 2004.

[9] `http://usarsim.sourceforge.net/`, USAIRSim homepage.

[10] `http://www.tno.nl/downloads/PreScanBrochureDec2008.pdf`, PreScan brochure.

Collective Intelligence: Toward Classifying Systems of Systems

Alan J. Ramsbotham, Jr.
Orion Enterprises, Inc.
2300 Fall Hill Avenue, Suite 212
Fredericksburg, VA 22401 USA
+1.540.373.6025
ramsboth@oei-tech.com

ABSTRACT

Intelligent systems are moving from science to more widespread engineering development and deployment. The objectives of this paper are to suggest design-oriented attributes that may provide a useful basis for classifying systems of systems. The discussion extends existing concepts, such as ALFUS, to complex ad hoc systems of systems wherein the individual elements can be geographically-dispersed and highly and independently mobile, and where the functions normally considered to comprise "intelligence" are distributed across the system. While not fully developed, the suggested extensions frame a discussion of how knowledge is obtained and distributed in such a system. Finally, the paper addresses some of the key challenges in predicting the performance of such complex intelligent systems of systems.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence, Multiagent systems

General Terms

Performance, Design, Theory

Keywords

Intelligent systems, collective intelligence, metrics, taxonomy, system of systems, complexity, receiver operating curve

1. INTRODUCTION

The following discussion draws heavily on the past efforts of others in autonomous and intelligent systems, unmanned systems, and knowledge management. [5][6][7] The purpose is to explore extensions of these existing concepts to move toward an effective ontology [13] for design and characterization of future "systems of systems" that exhibit collective intelligence.

The benefit of a well-defined ontology is that it facilitates understanding in development and provides a framework for standardization in implementation and operation. The challenge is that an effective ontology depends upon a common understanding of its semantics—that is the meanings and relationships of the terms selected as labels in context. [2] No one size fits all. Even within a given set of operational requirements, conceptual design solutions, and implementation options there will always be outliers and exceptions. The set of principles used to classify systems of systems must be flexible enough to accommodate design trade-offs and changes. These criteria may sound intuitively obvious and simplistic. In practice, meeting them can prove to be a daunting challenge. It is essential to begin with some vision of the nature of the systems we will ultimately build. Specifically, the discussion addresses systems of systems, and specifically, systems that in aggregate exhibit intelligent behavior.

2. BASIC OBSERVATIONS AND ASSERTIONS

There exists within society a range of diverse applications that, for various reasons, cannot be (or are best not) performed by human operators. In some cases the human simply cannot survive in the operating environment—for example, inspection of six-inch pipes or operation in a highly radioactive environment. In others, the human may be present, but unable to perform essential functions safely or effectively—for example, control of high performance fly-by-wire aircraft with "relaxed stability." [15]

Across the broad spectrum of unmanned and machine-assisted applications systems are required to perform more or less autonomously; that is, without human intervention. Some level of autonomy is arguably essential for machine intelligence. However, autonomy does not equate to intelligence. Intelligence implies an ability to perceive and adapt to external environments in real-time, to acquire and store knowledge regarding problem solutions, and to incorporate that knowledge into system memory for future use. The simple examples cited above arguably do not exhibit intelligence, per se. At the current state of the art, pipe inspection systems are generally tele-operated. In the high performance aircraft the control laws, although complex, time-critical, and adaptive, are pre-programmed. Both incorporate human intelligence in their basic design or operation. These observations lead to a few highly interdependent assertions.

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
Copyright © 2009 ACM 978-1-60558-747-9/09/09...$10.00

A system need not be intelligent to solve complex problems and perform useful functions.

A system can be both autonomous and adaptive without exhibiting intelligence. What matters is that the mission/task be performed adequately. As the complexity of the task increases and the system becomes more adaptive and autonomous in how it responds, the line between system complexity and machine intelligence becomes less distinct and meaningful.

For the foreseeable future, systems will depend on human intelligence to perform their tasks.

The dependence may be direct, as in the case of tele-operation, or it may be preprogrammed in the form of knowledge and rules of behavior. As we design and deploy systems that are capable of autonomous learning and self-modification, the distinction between system design and operation will blur. In a very real sense, we will design systems to re-design themselves. Regardless, human reasoning (or the fruits of human reasoning) will be essential elements of the system.

All higher intelligence is collective in nature.

Except for the most primitive instinctive functions, behaviors are acquired through observation of or imparted by others. In the case of autonomous intelligent systems, knowledge may be acquired through sensory perception and experience, and behaviors modified based on results of actions. Alternatively, knowledge/intelligence may be communicated from others. The rules and procedures whereby behaviors are modified may themselves evolve. But, as with human intelligence and collective behavior, they will evolve within practical constraints.

The idea of "collective intelligence" is gaining importance in the field of information systems, and is a useful concept for designing and characterizing intelligent systems of systems. Also useful is the concept, articulated by the evolutionary biologist Gregory Stock, of a "superorganism" of humans and machines that adapts and responds to its environment in an organic manner. [20]

For the foreseeable future, humans will be components of the system.

It may be useful from an engineering design perspective to approach hardware and software as something that humans use. However, except for the most tightly-constrained tasks, humans will remain essential components of systems, if for no other reasons than those of legal liability.

The ALFUS (Autonomous Levels For Unmanned Systems) provides a clear framework for characterizing human interaction with the mechanistic elements of intelligent systems (see Figure 1). Significantly, for this discussion, the highest level does not postulate full autonomy, but specifies "near-zero human-robot interaction (HRI)." From a system of systems engineering standpoint, however, human operators are functional components integral to the system.

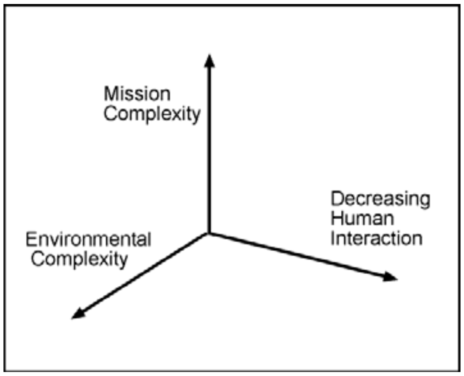


Figure 1. Basic conceptual framework of the ALFUS levels of autonomy.

3. CONCEPT OF A "SYSTEM OF SYSTEMS"

The term "system of systems" is a relatively new term. Its growing use is in large part a result of the U.S. Department of Defense's affection for the term in recent years. Much of the discussion is found in books, conference presentations, and symposia, as opposed to refereed journals. [8] While it is a truism that one man's system is another man's subsystem, the term is suggestive of a concept that is useful. It is a useful concept for this paper simply because some of the most promising uses for intelligent systems—transportation, military operations, and information infrastructure operations—will be systems of systems.

For purposes of this discussion we will make several key assertions about those systems:

- A system of intelligent systems will, by definition, exhibit some level of intelligent behavior and be an "intelligent system." A system of systems may exhibit intelligent behavior, even if none of the identifiable subsystems are "intelligent."
- The functional elements of the system of system may be geographically-separated, and if not mobile, physically or electronically transportable/transferable between different locations.
- While the maximum and minimum configurations of the system may be defined, at any given moment the operational configuration can be indeterminate and in many cases, indeterminable.

For purposes of discussion, we will also postulate that the extent to which our system of systems will exhibit intelligent behavior with respect to both its external and internal environment is also an important design feature. The ability to modify and evolve its own internal environment autonomously is the essence of learning, and has been addressed in detail and with considerable technical rigor in recent applications of the development of the 4D/RCS model architecture to learning. [1] It is the interfaces between the external environment and the intelligent systems that present the greatest design challenges and uncertainties.[19] To that end it is useful to decompose the functions comprising intelligent behavior into elements that correspond to specific hardware and software characteristics. For purposes of this discussion those will be to Sense, Perceive, Attend, Apprehend, Comprehend, and Effect Action (SPA^{2}CE$^{(sm)}$)[1], as follows:

- Sense: to generate a measurable signal (usually electrical) from external stimuli. A sensor will often employ techniques (for examples, bandpass filtering or thresholding) such that only part of the theoretical response of the transducer is perceived.
- Perceive: to capture the raw sensor data in a form (analog or digital) that allows further processing to extract information. In this narrow construct perception is characterized by a 1:1 correspondence between the sensor signal and the output data.
- Attend: to select data from what is perceived by the sensor. To a crude approximation, analogous to feature extraction.
- Apprehend: to characterize the information content of the extracted features. Analogous to pattern recognition.
- Comprehend: to understand the significance of the information apprehended in the context of existing knowledge--in the case of automata, typically other information stored in electronic memory.
- Effect action: to interact with the external environment or modify the internal state (e.g., the stored information comprising the "knowledge base" of the system) based on what is comprehended.

These are, in effect, a modest expansion of OODA (Observe, Orient, Decide, and Act). To a crude first approximation, the first four of these (sensing through attention, and in some configurations apprehension) comprise what we generally refer to as a sensory perception system, and thus to observation. Apprehension through comprehension map to orientation and decision-making. Effecting action corresponds to action, but includes, as an element of action, changes to the internal knowledge base and functions of the system. The distinction is one of focus and perspective: OODA speaks to the critical mission functions—the functions the end user (or in this case more accurately, the system of systems acting as an agent of the end user) must perform. The SPA^{2}CE framework defines functional elements relating to hardware/software design considerations.

From a systems engineering standpoint, the owner does not care about the system architecture or its intelligence—only that the mission be successfully completed.

Much of what we do as humans involves a combination of conditioned reflex and cognitive thought, depending on the nature of the action required. Intelligent systems will need to exhibit the same characteristic. The choice of which functions require automated reflexive action and which require cognitive decision-making, and the allocation of those functions will be important considerations in system of systems design.

The priorities that the system assigns to attending to specific sensors or data may be driven by preprogrammed design, sensor inputs, or human intervention. Similarly the knowledge that the system relies upon will, again, for the foreseeable future, reside in the combined memories of the human operators and the information systems. Regardless of the source of knowledge the ability to optimize system performance will depend upon (and will at times be critically limited by) availability of computational and communications services.

For the system of systems designer the choice of architecture and the details of functional allocations have major ramifications. These choices will drive trade-offs among: the cost and sophistication of individual nodes; the performance (bandwidth, latency, and availability) of the mobile ad hoc networks required to disseminate data and command and control; and mission performance and survivability.

Collective mission performance can be optimized in many diverse ways. Table 1 provides a framework of examples in rough order of functional and architectural complexity. Less complex group behaviors include leader-follower and collective swarming. More sophisticated swarms exhibit individuated behavior where each member of the swarm is able to interact with its local environment to self-optimize its behavior. This implies situational awareness and adaptability on the part of system, but not necessarily at the level of the individual node. For example, the architecture may incorporate an all-knowing “dictator node” that directs and optimizes the actions of other nodes in the system.

Basic learning behavior in artificial intelligence and swarming systems have been studied extensively. [3][10][11][12][17] This work forms a sound framework for designing and evaluating individual intelligent systems, homogeneous systems, and systems in which the functions comprising systems intelligence reside in a relatively small number of nodes.

In an ad hoc mobile system of systems the number of possible architectural variations is practically unlimited. In the case of more complex ad hoc systems of heterogeneous mobile systems distribution of awareness and decision-making capabilities may be required. The holy grail of the intelligent system of systems is the case where individual nodes will sacrifice (i.e., sub-optimize individuated behavior) to optimize collective mission performance or survivability of the group.

An example is the concept of future unmanned combat systems. As envisioned these will comprise of networks of unmanned smart sensors and robotic mobility platforms equipped for a wide range of land, air, and sea mission functions. The system is envisioned as reacting organically to maximize the forces operational effectiveness under any scenario. In this example an unattended surveillance sensor may optimize performance for a specific set of targets in a geographically-limited space at the expense of overall performance. Platforms may be "sacrificed" (that is placed in tactical situations where probabilities of survival are minimal) to achieve overarching objectives. As noted previously, in the specific tasks required of the integrated system and its component parts and the operational configuration of the system may be indeterminable at any point in time.

[1] The acronym SPA^{2}CE is a service mark of Orion Enterprises, Inc.

Table 1. Examples of "Systems of Systems"

Type of System	Defining Characteristics
Leader-Follower	Intelligent behavior exhibited by single node, and replicated (sometimes with minor adaptation) by other nodes.
Swarming (simple)	Loosely structured collection of interacting agents, capable of moving collectively.
Swarming (complex)	Loosely structured collection of interacting agents, capable of individuated behavior to effect common goals.
Homogenous intelligent systems	A relatively structured collection of identical (or at least similar) agents, wherein collective system performance is optimized by optimizing the performance of individual agents.
Heterogeneous intelligent systems	A relatively structured heterogeneous collection of specialized agents, wherein the functions of intelligence distributed among the diverse agents to optimize performance of a defined task or set of tasks.
Ad hoc intelligent adaptive systems	A relatively unstructured and undefined heterogeneous collection of agent, wherein the functions comprising intelligence are dynamically distributed across the system to adapt to changing tasks.

3. MOVING TOWARD A SYSTEM OF SYSTEMS ONTOLOGY

Table 1 provided a notional framework for characterizing intelligent behaviors in systems of systems. An important step will be to understand how knowledge is acquired and disseminated through the system. Kennedy and Eberhardt [11] observe that new behaviors are acquired by experience or in the form of communication from others. In the case of an ad hoc mobile intelligent system, further distinctions must be made between knowledge that must be communicated in real-time to meet mission requirement, that which can be programmed in by the user, and that which is designed in a *priori*. As will be discussed in the next part of this paper each of these sources of knowledge are subject to uncertainty.

The system architecture should be driven by the requirements, and taxonomies for our potential engineering solutions need to relate to the system architectures. For example, if the system requires persistent situational awareness of its surroundings and an ability to navigate autonomously in them, the design taxonomy needs to provide the necessary elements for sensing, signal and data processing, and simultaneous localization and mapping.

The existing ALFUS construct provides such a clear framework for autonomy. It has evolved meaningful metrics for implementation. It also inherently suggests logical extensions for characterizing collective intelligence within an intelligent system of systems context. Those extensions are:

- **Dynamic architectural complexity**—the number, diversity of intelligent behaviors, and mobility and spatial distribution of the nodes comprising the system of systems;
- **Functional complexity**—specifically the extent to which the hardware and software functions comprising intelligence (i.e., SPA^{2}CE), are centralized or distributed across the dynamic system architecture; and
- **Capability of the computing and communications infrastructure** to provide the necessary services to support and exploit those functions in real-time.

The resulting framework, if not completely developed, is reasonably comprehensive. ALFUS provides a framework for characterizing functional divisions between human and machine intelligence and for classifying autonomy in terms of the complexity of the task and the operating environment where position on the axis correlates strongly with system intelligence. The discussion provides some initial suggestions for possible ways to classify implementation—***how*** collective intelligence is a system of systems is attained.

4. CHALLENGES TO DEVELOPING A USEFUL ONTOLOGY

The challenges to developing an ontology that is sufficiently robust to provide a framework for systems of systems design and analysis fall into several broad categories: Mission uncertainty, complex interdependence of the diverse metrics; lack of accepted metrics for certain essential aspects of the system; and uncertainty associated with real-world physical limitation where mature metrics arguably exist.

Mission Uncertainty: The potential applications cited—transportation, military operations, and information infrastructure—share a common characteristic. They involve scenarios where the sets of tasks to be performed may be essentially indeterminable. To the extent that approaches to machine intelligence require iterative training or optimization, the best we can do is approach some level of confidence for the highest priority tasks (that is, those most likely to be encountered, where the cost impact associated with performance of the tasks are the highest).

Interdependence: At the system of systems level, the diverse metrics are highly interdependent, and the functions that comprise intelligent behaviors may be dynamically distributed across multiple elements of the system. Those elements exist in, and are integral features of the complex environment in which the system of systems must operate. For example, the electromagnetic transmission of active sensors and communications and the observable emanations of the physical elements of the system are all part of the complex signal environment against which those same subsystems and communications links must work. For intelligent transportation and military combat the system of systems will, in effect, constitute a significant part of its own environment.

Inadequate metrics for enabling technologies: Computing is clearly a key enabling technology. Yet to date, quantitative metrics for predicting computational performance (and particularly software performance) elude us. The best we can do is rely on benchmarks (which are, at best, valid only within a fairly narrow problem set) and on qualitative measures, not of the software itself, but of the software engineering process. This has the potential to limit our ability to predict the performance of intelligent systems in complex, computationally-intensive applications.

Reality and the laws of nature: Finally, where quantitative metrics exist (for examples: detection range, image resolution, data transfer rate, latency), our ability to predict their values accurately is limited by the laws of physics. Such limits are inherent when subsystems interact with the real-world environment—sensing, inter-system communications, and effecting external action. There are dynamic non-linear effects that the system cannot affect directly, such as: electromagnetic propagation; dynamic signal characteristics and noise; and effects of weather.

The net result of these effects can be illustrated by reference key functions of a distributed intelligent system:

Sensing, Perception and Classification. The outcomes of any binary classification process can be defined in terms of four possibilities, illustrated in Figure 2. [4] For physical sensors these are often characterized in terms of a receiver operating characteristic (ROC) curve shown in Figure 3. There is evidence of interest in using ROC analysis more broadly for evaluating classifiers for other purposes, including machine learning and data mining. [16][22] [2]

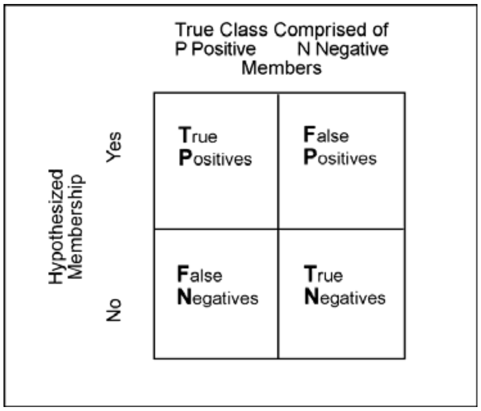
The ROC has been used for many decades as a metric for radar and radio receivers. The curve illustrates an essential idea—that the probability of a true classification can be made arbitrarily high, as long as the system (user) can tolerate a corresponding increase in false positive identifications.

A similar characteristic applies to distribution of knowledge in a mobile distributed ad hoc system. Shannon's theory of communications describes a level of uncertainty (entropy) associated with information, and the maximum theoretical information transfer rates that can be attained for a given transmission bandwidth (or bit rate) and signal to noise. Particularly in the case of mobile operations variations due to variations in path length, weather, multipath fade and interference, and signal blocking, predictions of signal-to-noise pose significant challenges.

The same statistical principles apply to retrieval of information from very large data sets. A relatively recent development is a growing interest in the use of the ROC for other applications, including for data mining and for the extraction of digital information very large data sets.

Finally, readers familiar with the literature on evolutionary computing and genetic algorithms will recognize the similarity in the general shape and behavior of the ROC curve with plots of performance of genetic algorithms as a function of the number of iterations/generations run. [3][9][14][21]

The primary significance of this discussion is to reaffirm the validity of the observation that "there is no such thing as a free lunch," and to underline the basic fallacy of an all-too-common belief in "information assurance" as something attainable in an absolute sense.



True Class Comprised of P Positive N Negative Members

Hypothesized Membership	P Positive	N Negative
Yes	**T**rue **P**ositives	**F**alse **P**ositives
No	**F**alse **N**egatives	**T**rue **N**egatives

Figure 2. Matrix illustrating possible outcomes of classification; sometimes referred to as a "confusion matrix"

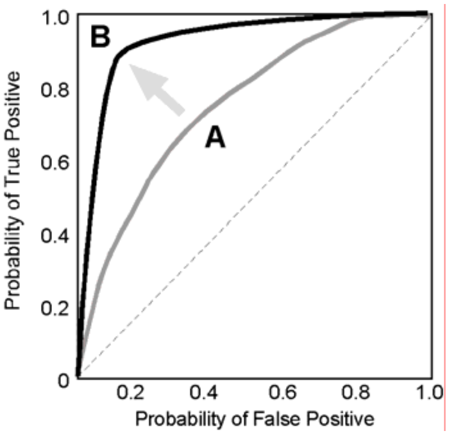


Figure 3. Receiver operating characteristic (ROC) curves. The ratio of true-to-false positives (shown as moving from A to B), but only at a cost of increased sampling or processing time.

[2] A Search of the ACM digital library for documents referencing ROC Analysis since 2005 returned 774 citations (June 16, 2009).

5. CONCLUSIONS

This paper suggests the utility of extending existing analytic/taxonomical frameworks to more explicitly address systems of systems. The specific framework suggested is intended to be illustrative, not definitive. Competing frameworks should be investigated. Once a suitable framework is established, much work will be required to bring new elements of the framework to a level of specificity comparable to ALFUS. But assuming that occurs, one can envision a new framework in which a composite measure of autonomy based on ALFUS, a measure of internal complexity, and measures for distributed computational performance form the axes of a system of systems metric.

Even given a comprehensive framework, as we begin to build more complex intelligent systems of systems, we will need to acquire knowledge and improve analytic tools and metrics. Among the more important will be:

- Better understanding of the complex interdependencies among existing metrics and system design;
- More accurate and reliable predictive for models characterizing the effects of complex dynamic non-linear physical effects on those subsystems that interact with external environments, and
- Better models and metrics for characterizing limits of information assurance based on these effects. This will be both a critical need and a major challenge.

ACKNOWLEDGEMENTS

The author would like to gratefully acknowledge the contributions of PerMIS 2009 conference reviewer, whose comments led to substantial improvements in this paper.

REFERENCES

[1] Albus, J., Bostelman, R., et al. 2006. Learning in a Hierarchical Control System: 4D/RCS in the DARPA LAGR Program, Journal of Field Robotics, December 29, 2006.

[2] Early, D., 2005. Resolving Taxonomy Challenges and Information Architecture Challenges, Presentation to the New Jersey Chapter of International Data Management Association, May 13, 2005.

[3]A.P. Engelbrecht , 2002. Computational Intelligence: An Introduction, Wiley, 2002.

[4] Fawcett, T. 2004. ROC Graphs: Notes and Practical Considerations for Researchers, HP Laboratories, dated 16 March 2004.

[5] Huang, H. 2007. Autonomy Levels for Unmanned Systems (ALFUS) Framework: Safety and Application Issues, Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, August 2007, Gaithersburg, MD.

[6] Huang, H. 2006. The Autonomy Levels For Unmanned Systems (ALFUS) Framework : Interim Results, Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, August 2006, Gaithersburg, MD.

[7] Huang, H., Messina, E., Albus, J. 2003. Autonomy Level Specification for Intelligent Autonomous Vehicles: Interim Progress Report*:* Proceedings of the 2003 Performance Metrics for Intelligent Systems Workshop, August 16-18, 2003.

[8] Jamshidi, Mo, 2005. System of Systems Engineering—a Definition, 2005 IEEE International Conference on Systems, Man, and Cybernetics, 10-12 October, 2005.

[9] Jan 't Hoen, P. Bohte, S. COllective INtelligence with Sequences of Actions, Accessed Aug 2009, on "CiteSeer," Penn State University
http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.14.6867

[10] Jones, M. T. Artificial Intelligence: A Systems Approach, 2008. Jones and Bartlett Publishers, Sudbury, Mass.

[11] Kennedy, J. Eberhart, R., and Shi, Y. 2001. Swarm Intelligence.

[12] Kennedy, J. and Eberhart, R. 1995. New optimizer using particle swarm theory in Proceedings of the 1995 6th International Symposium on Micro Machine and Human Science, pp. 39– 43.

[13] Howe, D. Dictionary.com. The Free On-line Dictionary of Computing.
http://dictionary.classic.reference.com/browse/ontology

[14] Michalewicz, Z., Dasgupta, D. (Eds) 1997. Evolutionary Algorithms in Engineering Applications, Springer Verlag.

[15] Pamadi, B., 2004. Performance, Stability, Dynamics and Control of Airplanes, American Institute of Aeronautics and Astronautics (AIAA), 2nd Edition.

[16] Prati, R., Batista, G., Monard, M, 2005. Evaluating classifiers using ROC curves, IEEE Latin American Transactions, Vol. 6, no. 2, June 2005.

[17] Reynolds, C. W. 1987. Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics, Vol. 21(4), pp. 25-34.

[18] Shannon, C. E., 1948. A Mathematical Theory of Communications, The Bell System Technical Journal, July-October 1948.

[19] Simon, H. and Kaplan, C. 1989. Foundations of Cognitive Science, from Foundations of Cognitive Science, MIT Press, ed. Michael Posner.

[20] Stock, G. 1993. Metaman: The Merging of Humans and Machines into a Global Superorganism, Simon and Schuster.

[21] Wolpert, D. H. and McReady, W.G. 1997. No free lunch threorems for optimization, IEEE Transactions of Evolutionary Computation 1.

[22] Zolghadri, M, Mansoori, E., 2007, Weighting fuzzy classification rules using receiver operating characteristics (ROC) analysis, Information Sciences , vol. 177 , no. 11, Elsevier, June.

A Decision-Theoretic Formalism for Belief-Optimal Reasoning

Kris Hauser
School of Informatics and Computing, Indiana University
Lindley Hall, Room 215
150 S. Woodlawn Ave.
Bloomington, Indiana
hauserk@cs.indiana.edu

ABSTRACT

Intelligent systems must often reason with partial or corrupted information, due to noisy sensors, limited representation capabilities, and inherent problem complexity. Gathering new information and reasoning with existing information comes at a computational or physical cost. This paper presents a formalism to model systems that solve logical reasoning problems in the presence of uncertainty and priced information. The system is modeled a decision-making agent that moves in a probabilistic belief space, where each information-gathering or computation step changes the belief state. This forms a Markov decision process (MDP), and the belief-optimal system operates according to the belief-space policy that optimizes the MDP. This formalism makes the strong assertion that belief-optimal systems solve the reasoning problem at minimal expected cost, given the background knowledge, sensing capabilities, and computational resources available to the system. Furthermore, this paper argues that belief-optimal systems are more likely to avoid overfitting to benchmarks than benchmark-optimized systems. These concepts are illustrated on a variety of toy problems as well as a path optimization problem encountered in motion planning.

1. INTRODUCTION

To complement the standard tool of benchmarking, computer scientists have theoretical tools to express algorithm performance, such as big-O notation for worst-case asymptotic complexity. The power of these tools is that they express rigorous, theoretical performance bounds over all inputs. But for complex intelligent systems that interact and reason about the world, benchmarks are often the most reliable way to measure performance. Indeed it is not clear that worst-case complexity is meaningful in the physical world. For example, the problem of navigation among movable obstacles is PSPACE-complete [20], but humans routinely navigate among movable obstacles without much difficulty. The worst-case problems that are truly hard are puzzles specifically designed to be hard. But benchmarking is not without its flaws; systems can perform well on benchmarks and poorly in the real world, and it can be difficult to design enough benchmarks to fully capture real-world performance.

This paper presents a new formalism to define rigorous, but realistic, theoretical bounds on the performance of systems that reason about and interact with an uncertain world. Specifically, given a scalar valued performance metric, this paper derives a bound on the best expected-case performance over a (typically infinite) space of problem instances Ω. The system is treated as a *decision-making agent* that operates in an "environment" drawn from Ω. The agent does not know which instance was picked, and must learn about the environment by probing it. This perspective has often been taken for systems that operate with physical uncertainty; the novelty in this work is to also treat "mental" uncertainty in the same unified framework. For example, it may know that $f(x) = 0$ but has not yet computed the value of x. Though x is mathematically defined, it is considered a random variable *unknown to the system* until the system computes it.

The formalism casts the operation of the system as a Markov decision process (MDP) (see [2] for a survey). At every point in time the system is identified with a belief state, and it has a set of available actions, such as gathering sensor readings or performing computation. Executing an action moves the belief state in a probabilistic way. To encode the performance metric, a utility function sums costs and rewards encountered by the system before it terminates. The belief-optimal system behaves according to the policy that optimizes expected utility. This paper shows that if the belief state encodes hypotheses about Ω accurately (through appropriate definitions of background knowledge), then on average a belief-optimal system performs better than any algorithm on Ω.

Furthermore, this work highlights the danger of overfitting on benchmarks when they are used to optimize program performance. A program optimized on benchmarks $\Omega' \subset \Omega$ will perform as least as well as any other program on Ω', but may perform poorly on Ω. I argue that systems optimized with decision-theoretic principles are often more robust than those optimized on benchmarks, because any set of benchmarks that represents a complex, high-dimensional problem space is impractically large. The decision-theoretic formalism approximates the problem space with statistical models, which are more likely to generalize to new problems.

PerMIS'09 September 21-23, 2009, Gaithersburg, MD, USA

2. RELATED WORK

Decision theory is a well-known tool for modeling and designing systems that interact with the real world, but it also has powerful applications to *logical problem solving.* Decision-theoretic approaches have been explored for reasoning problems in a diverse range of fields, such as numerical analysis, artificial intelligence, and robotics, where solutions are deterministically defined, and no *inherent* uncertainty exists except in the "mind" of the system. They have also been used to model bounded rationality in humans [17].

Examples of applications include analysis of Monte Carlo and quasi-Monte Carlo numerical integration and function approximation techniques [19], and in sampling strategies for stochastic numerical optimization [5, 21]. In artificial intelligence, efficient strategies have been developed for testing hypotheses represented as boolean formulas of uncertain statements [8]. Decision theoretic models have been applied to heuristic selection in heuristic search [4]. They have also shown promise in the field of robot motion planning, with speed gains of up to orders of magnitude [3, 11, 18]. My own research has applied decision-theoretic approaches to several motion planning subproblems [9], including path optimization, collision testing, configuration sampling, and contact selection strategies for robotic systems with contact.

It has also been argued that certain heuristics for randomized algorithms can be understood in a decision-theoretic sense, even if the heuristics are not themselves derived from the same principles. For example, stochastic optimization heuristics like simulated annealing, genetic algorithms, and ant colony optimization can be interpreted has having implicit probabilistic models of the function space [21]. The probabilistic roadmap (PRM) technique for robot motion planning, which connects a network of randomly sampled points in the robot's free space, can be interpreted as having implicit hypotheses about the shape of the free space [10]. This suggests that the performance of these algorithms may be improved by using better initial hypotheses, or better exploiting the information gathered during computation.

3. FORMAL MODELING OF BELIEF SPACE REASONING

This section describes how to model an intelligent reasoning system as a decision-making agent, and describes the associated POMDP formalism.

3.1 Assumptions

Consider a computer program that is given some background knowledge as input, and produces an output after executing a sequence of tests, which may involve either computational reasoning or gathering sensor readings. Assume that the sequencing of tests and termination conditions are structured by the problem logic such that the output is always correct. The outcome and/or cost of each test behaves with uncertainty, due either to stochasticity (a result of randomization or physical noise) or unpredictability (must be treated as a black box with internal workings that are too complex to be explicitly represented).

3.2 Modeling as a Belief-Space POMDP

The agent operates on a number of hypothetical logical *statements.* To a statement S, assign a belief p in $[0, 1]$, with $p = 0$ meaning S is certainly false and $p = 1$ meaning S is certainly true. An assignment of beliefs to all statements in the scope of a problem defines a *belief state*, and the set of all possible belief states is the *belief space.*

The program executes a sequence of *tests*, which are atomic operations that modify the belief on hypothetical statements. Upon observing the results of executing a test, the program moves to a new belief state. Without loss of generality, we define a test such that it determines the factuality of a statement S exactly.

The problem is considered solved when the belief state contains certain factual statements. The program may choose to terminate, which produces an output. The performance of the system is measured by a utility function that sums the negative *costs* incurred during execution, and positive *rewards* that assess output quality. Costs may include execution time and resource usage; for example, real-time constraints could be implemented by penalizing time limit violations. Rewards and costs should be weighted by the system designer so that the utility function measures overall system performance.

In full generality, a belief state on statements $\mathcal{S}$ forms a joint probability distribution $Pr(\mathcal{S}|Z)$. Here, Z represents *background knowledge* established prior to the current state. The importance of background knowledge will be discussed in Section 4.1. After executing a test T, the belief state should change to $Pr(\mathcal{S}|T \text{ succeeds}, Z)$ with probability $Pr(T \text{ succeeds}|Z)$, and to $Pr(\mathcal{S}|T \text{ fails}, Z)$ with probability $Pr(T \text{ fails}|Z)$. These changes are called the *transition dynamics.*

With these definitions, the problem solver has been cast as a belief-space Markov decision process. Because the variables that define the belief state are not directly observable, the MDP is known as a partially-observable Markov decision process (POMDP).

3.3 Illustration on Matrix Inversion

To illustrate the approach on a small example, suppose we are designing a system to compute the inverse or pseudoinverse of a matrix, where the properties of the matrix are unknown beforehand. Suppose the system has access two three methods of computing inverses: Cholesky decomposition, LU decomposition, and singular value decomposition (SVD). The challenge in this particular system is that Cholesky decomposition is much faster than LU decomposition when the matrix is symmetric positive definite (s.p.d.), but this property is unknown beforehand (in fact, attempting a Cholesky decomposition is frequently used to test if an unknown matrix is s.p.d.). Furthermore, LU decomposition is much faster than SVD when a matrix is invertible, but again, invertibility is unknown beforehand. SVD will compute a pseudoinverse or inverse for a general matrix.

Here, the belief space consists of four unknown properties of the input matrix: Square, Symmetric, Invertible, and SPD (symmetric positive-definite). The tests available to the system consist of two "probes", Is-Square and Is-Symmetric, and the three methods of computing inverses, labeled Cholesky, LU, SVD. The operations listed in the order of increasing cost are Is-Square, Is-Symmetric, Cholesky, LU, and SVD.

The optimal policy depends on the distribution of the kinds of matrices in Ω. For example, if very few matrices are invertible, then the policy of always using SVD would be

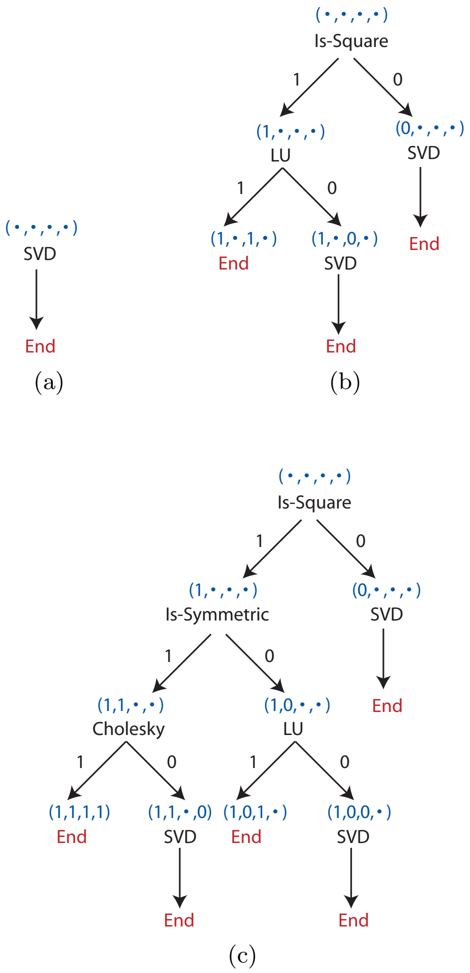


Figure 1: Three policies for the matrix inversion example which may be optimal, depending on the distribution of input matrices. Tuples $(\cdot,\cdot,\cdot,\cdot)$ indicate belief states, where elements respectively indicate that a matrix is square, symmetric, invertible, and symmetric positive-definite.

optimal (Figure 1a). If many matrices are invertible but few are symmetric positive-definite, then the policy in Figure 1b may be optimal. If all kinds of matrices are well-represented, then the optimal policy may gather as much information as possible, as illustrated in Figure 1c.

The joint distribution of Ω over the belief space (which can be represented by 16 real values), and the distribution of costs associated with each test, is sufficient to compute the optimal policy using the POMDP framework. This is not difficult for a problem of this size, but can be tremendously challenging for problems that are not much larger.

3.4 Illustration on a Household Mobile Robot

Consider another example of a mobile robot that is asked to retrieve an item in a house. It has a map of the house, and knows of N locations $L_1, \ldots, L_N$ where the item may be, some of which are inaccessible (out of reach or blocked by obstacles). The true location of the item is given by the random variable I. The robot does not know a priori which locations are inaccessible, and instead calls a planning subroutine Plan(L_i) that either returns a path from its current location R to the location L_i, or reports that it is inaccessible. If a path is available, Travel(L_i) moves the robot to the location and looks for the item. This example combines both physical uncertainty (where the item is located), as well as "mental" uncertainty (which locations are accessible).

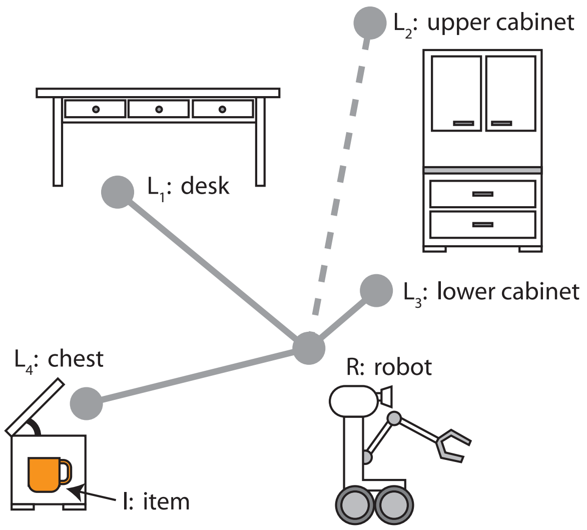


Figure 2: A household mobile robot is asked to search for a cup. The robot does not know, a priori, that the upper cabinet location is inaccessible. The robot determines accessibility by running a planning subroutine. Physical uncertainty is present in the location of the cup, and "mental" uncertainty is present in the accessibility of locations.

If we denote the accessibility of location L_i with the variable A_i, the belief state is defined over the statements $(I, R, A_1, \ldots, A_N)$. The program terminates when the item is found, or the robot knows the item is in an inaccessible location. The background knowledge will define the probability distribution on I and $A_1, \ldots, A_N$, as well as the expected costs of Plan() and Travel().

4. BELIEF-OPTIMAL POLICIES

The decision-theoretic formalism almost immediately defines the concept of a *belief-optimal policy*, and the performance bound associated with it. More precisely, let $U(\pi, Z)$ denote the expected utility of executing a policy π given background knowledge Z. Then the belief-optimal policy is

$$\pi^\star = \operatorname*{argmax}_\pi U(\pi, Z).$$

The next section shows that $U(\pi^\star, Z)$ is an upper bound to the average performance for problem instances drawn from Ω, so long as Z is properly defined.

4.1 The Role of Background Knowledge

Typically, Ω is enormous and unknown, so beliefs about Ω must be represented in the background knowledge Z. Suppose that background knowledge is defined such that, given any observed test results, the distribution of future test results can be inferred *accurately*. Then, the policy that optimizes the POMDP is the policy that achieves the lowest expected cost over Ω.

Accuracy is defined as follows. Let $\mu(X)$ denote the probability that a problem instance is drawn from a subset $X \subseteq \Omega$. Let $\Omega_{\mathcal{T}}$ be the set of instances that are consistent with a history of observed tests $\mathcal{T}$. Then the background knowledge Z is accurate if

$$Pr(T|\mathcal{T}, Z) = \frac{\mu(\Omega_{\mathcal{T}\cup\{T\}})}{\mu(\Omega_{\mathcal{T}})} \tag{1}$$

holds for any history of tests $\mathcal{T}$ and any future test T.

Appendix A shows that if background knowledge is accurate, then $U(\pi, Z)$ is equal to the average performance of π on the problem instances in Ω, and by consequence, $U(\pi^\star, Z)$ is an upper bound on performance. So in theory, the belief-optimal policy and performance bound are rigorously defined, and can be computed for small problems. But in practice, one can only hope for approximate solutions.

4.2 Representing and Computing Belief States

It is usually extremely difficult to define a problem distribution Ω and μ, much less compute an accurate representation in background knowledge. So for algorithm designers the perspective is reversed: background knowledge is chosen explicitly, and the problem distribution is defined implicitly to be consistent with the chosen background knowledge. Background knowledge can be encoded using a variety of machine learning and statistical models, e.g., logistic models, Bayesian networks, decision trees, neural networks, etc. Such models can encode prior beliefs as well as incorporate training on real problem instances. This in turn enables belief-optimal policies to generalize better to unseen instances, as argued in Section 6.

It is also impractical to compute or explicitly represent the joint distribution of the belief state because its size is exponential in the number of statements. The belief state and dynamics can be approximated by assuming statement independence, in which case the joint distribution is simply the product of distributions of individual statements in $\mathcal{S}$. More refined strategies might use assumptions of conditional independence, for example, representing variables in a sparse Bayesian network or other graphical models.

4.3 Optimizing POMDP Policies

Solving POMDPs in the general case is extremely computationally hard [15, 16]. In general there are two primary approaches to solving large POMDPs (which are not mutually exclusive): exploit problem structure, or approximate. Some problems have a structure for which exact solutions can be computed efficiently, such as several variants of the n-armed bandit problem [1, 4], and hypothesis testing of two-level AND/OR boolean-formulas [8]. POMDP approximation in large belief spaces is an area of active research, and perhaps the most promising results have come from sparse sampling and depth-limited search techniques [12, 13].

The advent of POMDP optimization techniques for large problems, on the order of thousands or millions of logical statements, will greatly accelerate the rate of development of complex intelligent systems. Intelligent systems will inevitably need to tackle computationally hard problems more frequently in the future, and belief-based optimization is an attractive, systematic approach to achieving good practical performance in the face of discouraging worst-case complexity theories.

5. A PATH SMOOTHING EXAMPLE

Here we illustrate a practical application of the decision-theoretic formalism to a path smoothing problem encountered in motion planning. Probabilistic roadmap motion planners tend to produce jerky, unnatural-looking paths due to their random exploration. A simple shortcutting method [6, 7] smoothes these paths by picking two random points A and B on a path, and testing the line segment $\overline{AB}$ for feasibility. If $\overline{AB}$ lies in free space, it replaces the portion of the path between A and B. After a handful of iterations, the largest unnecessary jerks are likely to be eliminated. However, it can take a huge number of iterations before the process converges to a smooth path. How many iterations are enough?

In a decision-theoretic formulation, the agent chooses shortcuts as tests and receives rewards that are accumulated over time. Each potential shortcut incurs a negative cost, and produces a reward only if it is successful. Although it is difficult to define background knowledge accurately, we approximate it by making some independence assumptions. The belief-optimal policy, given these assumptions ,is a greedy strategy. Experiments demonstrate that the greedy strategy converges much faster than picking random points, and has a natural termination criterion: halt when no shortcut has positive expected utility.

5.1 Problem Statement

Suppose the path $y(u)$ is parameterized with u in [0,1]. Denote the length of the path between parameters u and u' as $l(u, u')$, and denote the distance between two points q and q' in C-space as $d(q, q')$. Testing the line segment between u and u' incurs cost $c(u, u')$. If successful, $y(u)$ is replaced with the new path, and the planner receives reward $l(u, u') - d(y(u), y(u'))$ (the amount that path length is reduced). Let $p(u, u')$ denote the estimated probability of success. We assume $c(u, u') = c_s d(y(u), y(u'))$, where c_s is a constant reflecting the amount of computation time one is willing to spend for a unit decrease in path length.

This is essentially a "one-pull" variant of the well-studied n-armed bandit problem [1]. In the n-armed bandit problem, the decision maker chooses from n actions, $A_1, \ldots, A_n$. A stochastic payout z_k is awarded after choosing A_k, and payouts accumulate over time. If the distributions of each z_k are known and independent, the optimal strategy is greedy and picks the choice with the highest expected reward. This is also the case for "one-pull" bandits.

In other words, the greedy choice of u and u' maximizes $p(u, u')(l(u, u') - d(q, q')) - c(u, u')$. Dependencies between two candidate shortcuts do occur when the range of the shortcuts overlap, so the greedy strategy is not necessarily optimal, but it does perform quite well in practice.

5.2 Belief Estimation

The performance of the greedy strategy depends on the quality of the estimate $p(u, u')$. From first principles we know that the probability that a random line segment is feasible decreases with distance between its endpoints. Also, a line segment is likely to be invalid if nearby line segments are invalid. We keep a history of infeasible configurations $\mathcal{I}$ (which may be initialized with samples from PRM planning) and updated as shortcuts fail. We assign a belief to each shortcut as a function of segment length and the distance between the segment and the closest configuration in $\mathcal{I}$.

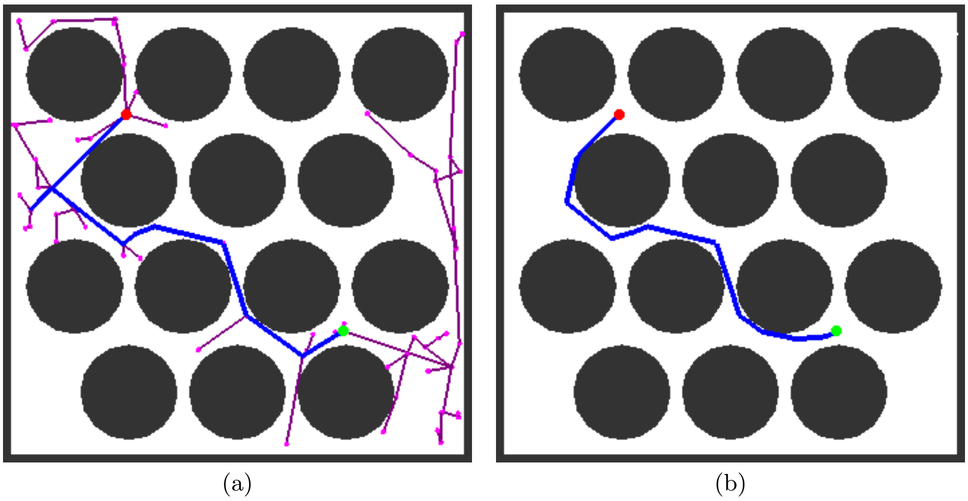


Figure 3: (a) PRM planners produce jerky paths. (b) Shortcutting produces a shorter path.

In the following experiments, we use a crude method to estimate $p(u,u')$ given $\mathcal{I}$. We tested the feasibility of 100,000 randomly sampled line segments, and built a histogram $p_0(d)$ of success rate indexed by distance d. We then weight the baseline probability $p_0(d(q,q'))$ by a function of the distance d' to the closest infeasible configuration. The experiments below used $f(d') = 1 - e^{-\alpha d'}$, where α was chosen by a small amount of tuning. A better method might estimate the joint success rate as a function of $d(q,q')$ and d'.

5.3 Experimental results

We demonstrate this for a C-space with regularly spaced circular obstacles in a unit square (Figure 3). The shortest path between two points may contain curves on the C-space obstacle boundaries (this is also true in C-spaces for robotic mechanisms with revolute joints). A good piecewise linear approximation to the shortest path requires a huge number of line segments, so shortcutting converges slowly.

To compare performance across varying start and goal configurations, we normalize reward by dividing by the straight-line distance between the starting points. We set the cost proportionality constant c_s to 0.01. Figure 4 compares the randomized strategy with a greedy, adaptive strategy. 200 random start and goal locations were sampled and connected using an RRT planner [14]. Starting at the output path, random shortcuts were performed for 1000 iterations. For the same starting path, adaptive shortcuts were performed until the strategy chose to terminate. The set of infeasible configurations $\mathcal{I}$ is initially empty.

Initially, the adaptive strategy reduces the path length quickly. Throughout execution, it achieves a given reduction in path length about two or three times faster than the randomized strategy. As more iterations are taken, shortening the path becomes increasingly harder. The adaptive strategy terminates naturally when the cost of making a shortcut exceeds the expected reward. All adaptive runs terminated by iteration 200.

6. AN ALTERNATIVE TO BENCHMARKS

6.1 An Argument Against Benchmarks

Benchmarks are a useful tool for performance optimization, but it is difficult to pick a representative benchmarking suite, and caution must be taken to avoid overfitting. In general, there will exist a sub-optimal algorithm whose benchmark performance is at least as good as the optimal algorithm. Existing techniques for avoiding overfitting include using large benchmarking sets, regularization, and cross-validation. Outside of the machine learning community, researchers rarely employ such techniques (I myself am guilty of this), and even if they are employed, the size of the benchmarking suite is rarely adequate to draw statistically significant conclusions.

One reason why it is so difficult to choose a benchmarking suite that accurately reflects real-world performance is that any set of problem instances that is statistically identical to Ω, as far as the algorithm is concerned, must span the joint distribution of Ω in belief space. Such a set is practical only if the belief space is tiny, or Ω happens to span a low dimensional subspace of belief space.

I argue that performance optimization using the decision-theoretic formalism can be more robust than benchmarks. The system's background knowledge Z can be defined to capture the belief space distribution of Ω more accurately and more broadly than a set of benchmarks.

A skeptic may raise two objections. First, background knowledge should be trained on a set of problem instances, which is essentially a set of benchmarks. Second, in practice, background knowledge can only represent the distribution of Ω approximately, and these modeling errors may reduce performance. To the first objection, I argue that background knowledge can better generalize small datasets by taking advantage of statistical and machine learning tools that are based on sound and well-developed theory (some of which were mentioned in Section 4.2). These tools will generalize far better than ad-hoc techniques that try to encode generalization into a complex algorithm. To the second objection, the empirical success of the decision-theoretic approach suggests that their behavior is relatively robust to approximation errors. An example will be shown in the following section.

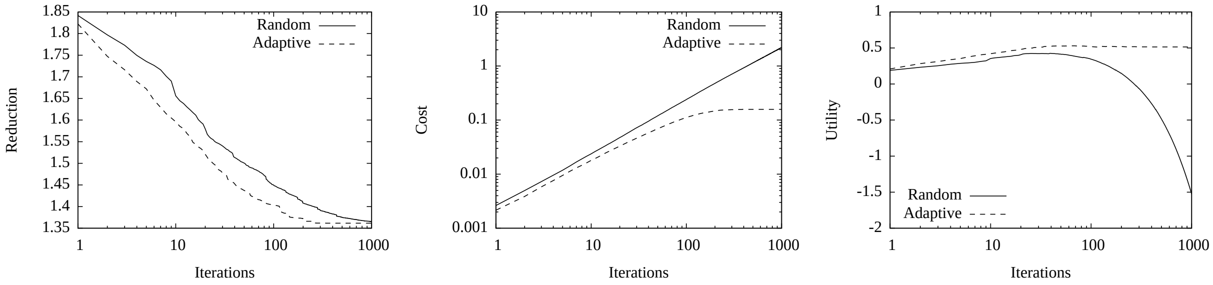


Figure 4: Results of shortcutting experiments, reporting relative path length, accumulated cost, and utility. Averaged over 200 plans with random start and goal configurations. Iteration numbers are plotted on a log scale.

6.2 Illustration on an Urn Problem

Consider a system that searches for a black ball placed in one of two urns. In each problem instance, the black ball is placed into an urn and both urns are filled with additional white balls. The system has two types of tests: A_k asks the number of balls in urn k, and U_k picks urn k and searches through it. A search continues until it finds the black ball or empties the urn completely (that is, it is not allowed to stop midway through). A problem instance can be specified as a 3-tuple (n_1, n_2, K), where n_1 and n_2 are respectively the number of balls in the 1st and 2nd urns, and K is the index of the urn containing the black ball. The cost of A_k is 1, and the expected cost of U_k is $n_k/2$ if $K = k$, and is n_k otherwise.

Suppose the benchmark problem instances are $(30, 1, 1)$, $(20, 15, 1)$, $(40, 30, 2)$, and $(35, 60, 2)$. A benchmark-optimal strategy is to test A_2, and if $n_2 \geq 30$, then to pick U_2 first. Otherwise, pick U_1 first. Call this strategy π_1. On the benchmarks, π_1 has average cost 18.5.

Suppose now that our *approximate* background knowledge consists of a minimum and maximum estimate of the number of balls in each urn, and the fraction of black balls found in urn 1. Also, let the system assume that the number of balls in an urn has a uniform distribution between the minimum and maximum values, and is independent of the number in the other urn. Given these approximations, the POMDP expected utility of π_1 is 35. Now consider a strategy π_2 that tests both A_1 and A_2, and then picks U_1 first if $n_1 < n_2$ and U_2 otherwise. The expected value of this strategy is approximately 28.3. The worst-case cost of π_2 is also better than π_1: 72 versus 81.

7. CONCLUSIONS

This paper presented a decision-theoretic formalism for optimizing and bounding the expected performance of an intelligent system that interacts and reasons about the physical world. The formalism is based on casting the system as a decision-making agent that faces uncertainty due to sensor noise as well as uncertainty due to bounded rationality. This forms a partially-overvable Markov decision process (POMDP) that can be optimized. I showed that if the beliefs of the system accurately capture the problem space characteristics, then the strategy that optimizes the POMDP also optimizes the average case performance over the problem space. This paper also argued that the decision-theoretic approach is an attractive alternative to benchmarking for performance optimization that avoids the problem of overfitting to benchmarks. These principles are illustrated on a variety of example problems. Future work should investigate the tractability and accuracy of approximation techniques for solving the large POMDPs that result from this formalism, and also to apply the formalism to systems that solve new and challenging problems.

8. ACKNOWLEDGEMENT

Many thanks to Jean-Claude Latombe for numerous helpful discussions during the early stages of this work.

APPENDIX

This appendix derives the result that if background knowledge Z is defined such that (1) holds, then $V(\pi, \Omega) = U(\pi, Z)$. It is largely a technical matter, but is included here for completeness.

To define U and V more precisely we first need some preliminary definitions. Here we will represent a state s as a history of tests $\{T_1, \ldots, T_n\}$, because the belief state and dynamics are fully determined by the background knowledge (which stays constant) and the test history. Let $\pi(s)$ denote the action taken by policy π at state s, and in a slight abuse of notation, also let it denote the result of the test $\pi(s)$. Let $R(s, a)$ be the reward minus the cost of executing action a in state s. Assume without loss of generality that Ω is defined such that given a problem instance ω, transitions are deterministic (i.e., ω is the "true" value of the world state). Also, assume terminal states are absorbing, and the performance metric is bounded.

Define the utility function $U_\pi(s)$ as the unique solution to the following system of equations over s:

$$\begin{aligned} U_\pi(s) = R(s, \pi(s)) &+ Pr(\pi(s)=0|s, Z)U_\pi(s \cup \{\pi(s)=0\}) \\ &+ Pr(\pi(s)=1|s, Z)U_\pi(s \cup \{\pi(s)=1\}). \end{aligned} \tag{2}$$

This is a slight abuse of notation; our original definition of the utility function is $U(\pi, Z) \equiv U_\pi(\{\})$.

Let the trace of running policy π on instance ω be defined as $(s_0, s_1, \ldots)$. Define the return v_0 using the recursive formula

$$v_i(\pi, \omega) = R(s_i, \pi(s_i)) + v_{i+1}(\pi, \omega).$$

Then the average performance of π is

$$V(\pi, \Omega) = \int_{\omega \in \Omega} v_0(\pi, \omega) d\mu(\omega).$$

Let's define

$$Y_\pi(s) = \int_{\omega \in \Omega_s} v_{|s|}(\omega, \pi) d\mu(\omega)$$

so that $Y_\pi(\{\}) \equiv V(\pi, \Omega)$. Then, denote $s' = s \cup \{\pi(s)=1\}$ and $s'' = s \cup \{\pi(s)=0\}$, and split Ω_s into subsets $\Omega_{s'}$ and $\Omega_{s''}$ depending on the results of $\pi(s)$. We get

$$\begin{aligned} Y_\pi(s) &= \int_{\omega \in \Omega_s} R(s, \pi(s)) + v_{|s|+1}(\omega, \pi) d\mu(\omega) \\ &= \mu(\Omega_s) R(s, \pi(s)) + \int_{\omega \in \Omega_{s'}} v_{|s|+1}(\omega, \pi) d\mu(\omega) \\ &\quad + \int_{\omega \in \Omega_{s''}} v_{|s|+1}(\omega, \pi) d\mu(\omega) \\ &= \mu(\Omega_s) R(s, \pi(s)) + Y_\pi(s') + Y_\pi(s'') \end{aligned} \tag{3}$$

Now if we let $X_\pi(s) = Y_\pi(s)/\mu(\Omega_s)$, we have

$$X_\pi(s) = R(s, \pi(s)) + \frac{\mu(\Omega_{s'})}{\mu(\Omega_s)} X_\pi(s') + \frac{\mu(\Omega_{s''})}{\mu(\Omega_s)} X_\pi(s'') \tag{4}$$

Since background knowledge is accurate, then by (1) we have

$$\begin{aligned} X_\pi(s) = R(s, \pi(s)) &+ Pr(\pi(s)=0|s, Z) X_\pi(s \cup \{\pi(s)=0\}) \\ &+ Pr(\pi(s)=1|s, Z) X_\pi(s \cup \{\pi(s)=1\}). \end{aligned} \tag{5}$$

Since X_π solves for (2), then it must be identical to U_π. Then $V(\pi, \Omega) \equiv Y_\pi(\{\}) = U_\pi(\{\}) \equiv U(\pi, Z)$ as desired.

A. REFERENCES

[1] D. Berry and B. Fristedt. *Bandit Problems.* Chapman and Hall, 1985.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[3] B. Burns and O. Brock. Single-query motion planning with utility-guided random trees. In *IEEE Int. Conf. Rob. Aut.*, 2007.

[4] V. A. Cicirello and S. F. Smith. The max k-armed bandit: A new model of exploration applied to search heuristic selection. In *AAAI*, 2005.

[5] D. Fouskakis and D. Draper. Comparing stochastic optimization methods for variable selection in binary outcome prediction, with application to health policy. *Journal of the American Statistical Association*, 03(484):1367–1381, 2008.

[6] R. Geraerts and M. Overmars. Clearance based path optimization for motion planning. In *IEEE Int. Conf. Rob. Aut.*, New Orleans, LA, 2004.

[7] R. Geraerts and M. H. Overmars. Creating high-quality paths for motion planning. *Intl. J. of Rob. Res.*, 26(8):845–863, 2007.

[8] R. Greiner, R. Hayward, M. Jankowska, and M. Molloy. Finding optimal satisficing strategies for and-or trees. *Artificial Intelligence*, 170:19–58, 2006.

[9] K. Hauser. *Motion Planning for Legged and Humanoid Robots.* PhD thesis, Stanford University, 2008.

[10] D. Hsu, J. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Int. J. Rob. Res.*, 25(7):627–643, 2006.

[11] D. Hsu, G. Sánchez-Ante, and Z. Sun. Hybrid prm sampling with a cost-sensitive adaptive strategy. In *IEEE Int. Conf. Rob. Aut.*, pages 3885–3891, Barcelona, Spain, 2005.

[12] M. Kearns, Y. Mansour, and A. Y. Ng. Approximate planning in large pomdps via reusable trajectories. In *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.

[13] H. Kurniawati, D. Hsu, , and W. Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*, 2008.

[14] S. M. LaValle and J. J. Kuffner, Jr. Rapidly-exploring random trees: progress and prospects. In *WAFR*, 2000.

[15] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

[16] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov chain decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[17] A. Rubinstein. *Modeling Bounded Rationality.* MIT Press, 1998.

[18] G. Sánchez and J.-C. Latombe. On delaying collision checking in PRM planning: Application to multi-robot coordination. *Int. J. of Rob. Res.*, 21(1):5–26, 2002.

[19] J. F. Traub, H. W. G. W., and Wasilkowski. *Information-Based Complexity.* Academic Press, New York, 1988.

[20] G. Wilfong. Motion planning in the presence of movable obstacles. In *Fourth Annual Symposium on Computational Geometry*, pages 279 – 288, Urbana-Champaign, IL, 1988.

[21] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo. Model-based search for combinatorial optimization: A critical survey. *Annals of Operations Research*, 131(1–4):373–395, October 2004.

Evaluation of Automatically Generated Reactive Planning Logic for Unmanned Surface Vehicles

M. Schwartz
Energetics Technology Center
La Plata, MD, USA
mschwartz@etcmd.org

P. Svec
University of Maryland College Park
Department of Mechanical Engineering
petrsvec@umd.edu

A. Thakur
University of Maryland College Park
Department of Mechanical Engineering
athakur@umd.edu

S. K. Gupta
University of Maryland College Park
Department of Mechanical Engineering and Institute for System Research
skgupta@umd.edu

ABSTRACT

Unmanned Surface Vehicles (USVs) often need to utilize high speed reactive planning to carry out certain mission tasks. Development of a robust reactive planning logic is a challenging task. We have been exploring the use of virtual environments and machine learning to automatically synthesize a reactive planning logic to block the advancement of an intruder boat toward a valuable target. An important component of our work is to evaluate the performance of the automatically generated planning logic. We have used a virtual environment based game to compare the efficiency of an automatically discovered decision tree representing a planning logic for blocking to the behavior exhibited by the human operators. During our testing we used four volunteers to play against each other and against the computer. In human against human testing, the four players took turns playing the role of the USV and the intruder. In computer against human tests the four players played the role of the intruder while computer played the role of the USV defending a target. The efficiency of the logic was measured in terms of the time delay applied on the intruder by the USV as the USV carried out blocking maneuvers to protect a target. Our preliminary results show that a genetic programming based framework is capable of generating decision trees expressing useful reactive blocking logic.

Categories and Subject Descriptors

I.2.1 [Applications and Expert Systems] *Games*

I.2.2 [Automatic Programming] *Program synthesis*

I.2.6 [Learning] *Knowledge acquisition, Parameter learning*

I.2.8 [Problem Solving, Control Methods, and Search] *Control theory*

I.2.9 [Robotics] *Autonomous vehicles, Kinematics and dynamics*

I.3.5 [Computational Geometry and Object Modeling] *Physically based modeling*

I.3.7 [Three-Dimensional Graphics and Realism] *Virtual reality, Color, shading, shadowing, and texture*

General Terms

Dynamics Simulation, Dynamics Meta-model, Evaluation, Evolutionary Computing, Genetic Programming

Keywords

Autonomy, co-evolution, reactive planning logic, unmanned surface vehicle

1. INTRODUCTION

A major issue in the development of increased autonomy for robotic vehicles such as unmanned surface vehicles (USVs) is the time and expense of developing the software necessary to handle a large variety of missions and all the variations in the encountered environments. This is a truly challenging task and requires writing hundreds of thousands of lines of code by human programmers.

We have developed a new approach for developing planning software that operates autonomous USVs. This new approach takes advantage of the significant progress that has been made in virtual environments and machine learning. The basic idea behind our approach is as follows. The USV explores the virtual environment by randomly trying different moves. USV moves are simulated in the virtual environment and evaluated based on their ability to make progress toward the mission goal. If a successful action is identified as a part of the random exploration, then this action will be integrated into the logic driving the USV. We anticipate that there may be portions of the mission, where trial and error alone will not be adequate to discover the right decision rule. In such cases, two additional approaches are utilized to make progress in acquiring the right logic. The first approach involves seeding the system with the logic employed by humans to solve a challenging task. The second approach is to restrict the action space based on some type of feasibility criteria. This paper mainly

PerMIS '09, September 21-23, 2009, Gaithersburg, Washington, D.C., USA.

focuses on a reactive planning logic used for blocking the advancement of an intruder boat towards a valuable target.

An overview of our overall approach is shown in Figure 2. The first major component of our approach is development of a physics-based meta-model. High fidelity simulation of USV is time consuming and cannot be used for discovering decision rules or trees used in planning. We have developed a meta-model by conducting off-line simulations of the USV in the sea. This simulation accounts for wave and USV interactions. The meta-model provides information about turning radius, steady state velocity, and acceleration as a function of rudder angle and throttle position. Section 4 provides more details on this component.

We have developed a mission planning system whose main part is an evolutionary module for evolving planning decision trees. We used this system to automatically generate decision trees expressing blocking logic for the USV. This means that instead of automatically generating a program composed of low-level controller actions (steer left/right, go straight), we generate a program represented as a decision tree that consists of high-level controllers as building blocks (encircle intruder, go in front of the intruder, etc.) together with conditionals and other program constructs. The biggest challenge we faced was the efficient generation of various test-cases (intruder's attacking logic) for the USV blocking logic. To generate more general USV blocking logic, we employed the co-evolution approach. To make this approach feasible (time spent on the computation), (a) we had to seed a portion of both the initial populations for the USV and intruder by human-crafted initial set of blocking logic and (b) to evaluate the USV's blocking logic against a subset of the best intruders from the previous generations and vice-versa. Section 5 provides more details on this.

We have developed a virtual environment based game that allows human players to play against each other or against the computer. In the game, the player controlling the intruder boat must collide with a protected target, while the player controlling the USV must block and delay the intruder as long as possible. In addition to offering basic capabilities, the game provides collision detection and basic physics to the objects in the scene. The game logic is responsible for the rules of the game, game logging and replay, boat behaviors, and scoring. The game can be played on two computers over a network. Section 6 provides more details on this game.

Deployment of autonomous USVs in critical missions requires that the performance of the autonomous system matches with that of a remote controlled vehicle. Therefore, we have started an effort to assess the performance of automatically generated blocking logic compared to blocking maneuvers exhibited by human operators. We have used the virtual environment based game to compare the automatically discovered decision trees representing the blocking logic to the strategies used by human players. During our testing we gathered four volunteers to play against each other and against the computer. In human against human testing, the four players took turns playing the role of the USV and the intruder. In computer against human evaluation tests, the four players played the role of the intruder while computer played the role of the USV. The efficiency of the logic was measured in terms of the time delay applied on the intruder by the USV. Section 7 provides more details on the evaluation methodology.

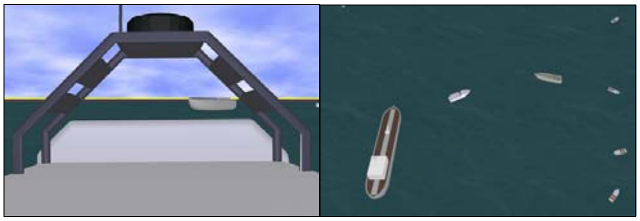
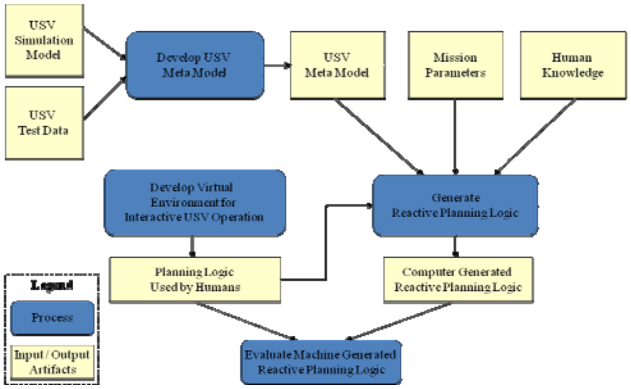


Figure 1. Virtual environment for simulation


USV Simulation Model
USV Test Data
Develop USV Meta Model
USV Meta Model
Mission Parameters
Human Knowledge
Develop Virtual Environment for Interactive USV Operation
Generate Reactive Planning Logic
Legend
Process
Input / Output Artifacts
Planning Logic Used by Humans
Computer Generated Reactive Planning Logic
Evaluate Machine Generated Reactive Planning Logic

2. Figure 2. Overview of the overall approachRelated Work

Technological advances in unmanned surface vehicles have enabled unmanned boats to be involved in certain missions which could be potentially dangerous for humans. The use of USVs is being proposed for a myriad of tasks like surveying, marine research, mine sweeping, etc. Currently, the best known USVs are semiautonomous. This means that the way-points programmed into these semiautonomous USVs are initially determined by human navigators. The built-in navigation planners of these USVs employ deliberative and reactive obstacle avoidance (OA) modules to ensure safe movement between the way points. Some also compute new way points in response to fault conditions. The Space and Naval Warfare Systems Center, San Diego has developed an autonomous navigation and OA architecture for USVs that supports both deliberative and reactive OA [1]. The deliberative OA contains a quick path planner for diverting the route of a USV away from threats of stationary and moving obstacles in the far-field. In their further work [2], they extended the deliberative part of the navigation planner to generate paths consistent with the rules of the road during all stages of the planning. For a detailed description of the current state of USV autonomy, see [3].

Genetic programming (GP) [4, 5] as one of the robust evolutionary techniques has been used for automatically generating high-level controllers or planning logic in different domains. To be able to discover a general planning logic in a competitive setting, the co-evolutionary process is needed. This process automatically generates challenging test cases for the current individual being evolved. Co-evolution is thus an alternative to standard evolutionary methods and is based on the "Arms race" assumption [6]. According to this assumption, newer individuals are expected to perform better than their ancestors. Competitive fitness and co-evolution were first explored in the context of the Iterated Prisoner's Dilemma in [7]. Koza further discussed GP based co-evolution for a simple discrete planning logic game in [5]. Reynolds discussed how to use competitive co-evolution for evolving strategies for players in the pursuit and evasion game positioned in a continuous geometric environment [8]. More recently, a game planning logic was evolved using competitive co-evolution for an ant taking part in the Ant Wars contest [9]. The task for the ant was to collect food in a toroidal grid environment in the presence of a competing ant. The co-evolved planning logic is human-competitive in a sense that it was able to beat other human-programmed planning logic. Similarly, RoboCode [10, 11] is an another example of using simulated co-evolution for generating human-competitive game planning logic for competing tanks in a grid based environment. Multi-population competitive co-evolution is used for developing more general high-level controllers in car racing domain [12]. The work described in [13] shows evolutionary architecture for evolving team tactics for a combative 2D gaming environment using GP.

Measuring the quality of task outcomes from different robotic domains (mapping and localization, obstacle avoidance, search and rescue, etc.) is necessary in order to be able to compare different techniques or algorithms. This can be done first in a simulator to prevent any possible damage to robots, followed by rigorous testing in a real environment. It is important for the benchmark itself to be accurate. This encourages other researchers to replicate it on the same problem so that they are able to accurately compare their own control or navigation strategies to existing ones.

The benchmarking work in robotics is often performed through competitions. These are mainly student competitions like National Robotics Challenge [14], FIRST Robotics Competition (Lego League, Tech Challenge) [15], or Eurobot [16]. There are also technically more challenging competitions geared towards academic communities. For example, the world's largest robot competition RoboGames [17] or RoboCup soccer [18]. There are also robotic competitions appealing to much broader communities such as aerial robotic competitions (IARC [19]), ground robotic competitions (DARPA Urban Challenge [20]), or underwater robotic competitions (Autonomous Underwater Vehicle Competition sponsored by AUVSI and the U.S. Office of Naval Research). Others can be readily found on robotics related websites.

In the field of UAVs, reference [21] describes an approach for evaluating algorithmic and human performance for UAV-based surveillance missions. Two main parts include an evaluation test-bed consisting of 243 scenarios uniformly covering most of all possible missions and a decision-theoretic framework for measuring the performance of a surveillance method in a given mission.

3. Reactive Planning Logic Executor Architecture

The complexity of interactions of a mobile robotic system suggests structured (non-monolithic) high-level controller architecture. The unmanned boats must behave based on the effect of several independent threads of reasoning. This is implied by the highly parallel nature of events and processes in the real world. The high-level controller architecture can meet this requirement if it is modular, and when the modules can act simultaneously in a coordinated cooperation.

The navigation system resides inside the USV and consists of perceptual, reasoning / planning, localization, and behavioral components. The USV's planning logic executor itself (see Figure 3) is purely reactive which means that it interprets and triggers reactive planning logic in a strict timely fashion. It computes only one action in every discrete moment based on the current state so that it is able to cope with highly dynamic and unpredictable environments.

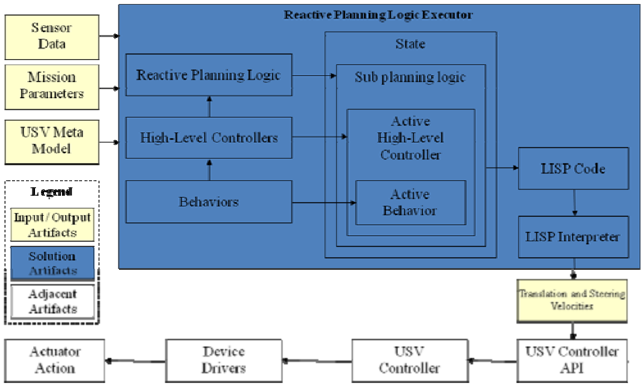


Figure 3. Reactive planning logic executor architecture

The reactive planning logic is a stored human-readable structure producing different motor actions (translation and steering velocities) in different situations. Its representation is a decision tree consisting of different high-level action controllers, conditional rules, standard boolean values and operators, conditional variables, program blocks, and velocity commands.

The structure of a high-level controller is based on the behavior-based subsumption architecture [22]. This architecture decomposes a complicated high-level intelligent behavior (a high-level controller in our case) into a set of more solvable and simple behaviors (steer left / right, go straight, arrive) organized into layers. These primitive behaviors are finite state machines acting in response to sensor inputs and producing motor action outputs. Multiple behaviors of the high-level controller can be activated simultaneously producing different conflicting motor commands. This means that a certain amount of coordination is needed. Due to its robustness, we have chosen a priority-based arbitration mechanism, picking the motor action output of the behavior with the highest priority (e. g. obstacle avoidance) as the overall motor action output of the high-level controller. In this high-level controller structure, the behavior in the bottom layer has the highest priority (for example obstacle avoidance) while the behavior in the top layer represents the most abstract functionality.

The high-level controllers, which are the main building blocks of a planning logic, can be parameterized. The parameters of a high-level controller define its underlying property. For instance, for the GO-INTRUDER-FRONT controller, the parameter defines the USV's relative goal position (in polar coordinates) in front of an intruder. This effectively allows the USV to cover all feasible positions, as defined by its planning logic around the intruder.

The inputs into the reactive planning logic executor are sensor data, description of mission parameters, and the USV meta model itself. The outputs are translation and steering velocities for a low-level USV controller that are directly translated into motor commands for device drivers of a particular actuator. At any particular moment, the logic executor can be in only one state. As figure 3 shows, the planning logic interpreter takes the currently used planning logic as an input and produces action outputs. Based on the current sensor data readings, only one high-level controller inside the planning logic is activated. This high-level controller consists of multiple primitive behaviors and decides which one is used to produce the ultimate action output.

4. Development of Physics-based Meta Model

The system consists of a 6 degree of freedom USV simulation model, which is used for computing the dynamics of the USV under any given sea-state. The full fledged USV simulation is computationally expensive, so we created simplified meta-models in the form of lookup tables by performing exhaustive computations using the USV simulation model for each given sea-state and USV model. The dynamic meta-models enable faster computation of fitness values in the evolution of the planning logic. We employed 6 degree of freedom nonlinear USSV dynamics model reported by Krishnamoorthy et al. as given by equation 1 [22].

$$\left.\begin{array}{l} M_H\dot{v} + C_H(v)v + D_H(v)v + g(p) = \overline{F_E} + \overline{F_P} \\ \dot{p} = J_P(v) \end{array}\right\} \qquad (1)$$

where,

$M_H = M_{RB} + M_A$: M_{RB} is the (6X6) inertia matrix and M_A is the (6X6) added mass matrix,

$C_H = C_{RB} + C_A$: C_{RB} is the (6X6) Coriolis and Centripetal matrix and C_A is the (6X6) hydrodynamic Coriolis and Centripetal matrix,

D_H : (6X6) diagonal damping matrix,

$g(p)$: (6X1) restoring force vector,

F_E : (6X1) environment force vector,

F_P : (6X1) actuator force vector,

J_P : (6X6) Jacobian matrix

v : (6X1) velocity vector relative to inertial frame and expressed in body-fixed frame

p : (6X1) vector representing position of USV relative to as well as expressed in the inertial frame of reference

We computed the added mass matrix M_A using strip theory as explained by Fossen [23]. Unlike the approach suggested by Krishnamurthy et al. to estimate the restoring force vector using an approximated simplified formula we computed the instantaneous restoring forces and moments $g(p)$ by actually intersecting the USV geometry with the wave at the given location of the USV resulting into a more accurate estimation of restoring force and moments. To compute the environment force we only considered the wave forces and ignored the effects of wind $(F_E = F_W)$. To compute the wave force we used the equation 2.

$$F_W = \begin{bmatrix} \left[-\rho \iint_{S_B} \left(\frac{\partial \phi}{\partial t} + 0.5 \nabla \phi . \nabla \phi \right) \hat{n} dS \right]_{3\times 1} \\ \left[-\rho \iint_{S_B} \left(\frac{\partial \phi}{\partial t} + 0.5 \nabla \phi . \nabla \phi \right) \vec{r} \times \hat{n} dS \right]_{3\times 1} \end{bmatrix} \qquad (2)$$

where,

ϕ : Velocity potential

S_B : Instantaneous wet surface area of USV

ρ : Density of water

$\hat{n}$: Normal surface vector

$\vec{r}$: Position vector of points on the surface in body –fixed frame of reference

The simulation technique described above is computationally expensive because at every time step it performs the geometric computation of the wet area of USV and the surface integration.

To use the simulator in reactive planning we employed a look up table based meta-model. The lookup table meta-model is pre-populated by running the simulator under various sea-states. The meta-model is then used in subsequent computations as described in the following sections.

The structure for the meta-model that we employed is shown in Figure 4.

```
class MetaModel {
  SeaState seastate;
  double AvgMaximumVelocity;
  double StddevMaximumVelocity;
  double AvgPositionalError;
  double StddevPositionalError;
  double AvgMaximumAcceleration;
  double StddevMaximumAcceleration;
  double AvgTurningRadius;
  double StddevTurningRadius;
};
```

Figure 4. Structure of the meta-model

5. Automated Generation of Blocking Decision Tree

Our solution to the problem of reactive planning was the use of a computer simulated evolution based on the Darwinian principles of survival and reproduction of the fittest. Using this phenotype evolutionary process, we were able to evolve the actual decision tree representing a planning logic. The specific evolutionary method we used is the strongly-typed genetic programming (GP) [4, 5]. This is a robust stochastic optimization method that searches a large space of candidate hypotheses (programs) while looking for the one with the best performance (fitness value). During the evolution itself, a population of USV decision trees is being stochastically transformed into a new population with a better average fitness value using standard evolutionary operators like crossover and mutation.

We were particularly interested in automatically discovering a blocking logic for slowing down the movement of the intruder toward the protected object. This blocking logic is defined in the context of a simple mission. During this mission, the USV must protect an oil tanker by patrolling around it while avoiding collisions with friendly boats and scanning the environment for a possible intruder. The environment around the oil tanker is divided into danger and buffer zones (see Figure 5). Once the intruder enters the buffer zone, the USV approaches the intruder boat and circles it for surveillance purposes. If the intruder enters the danger zone, the USV does its best to block the intruder, slowing the intruder's progress toward the protected object.

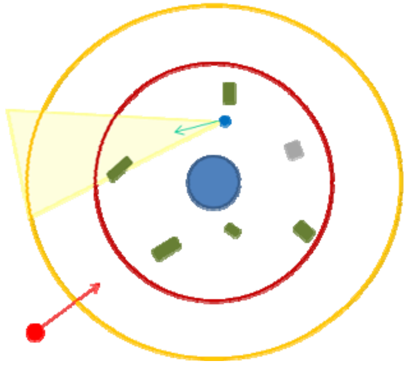


Figure 5. Simple mission: USV protects area containing objects of interest

Instead of generating a decision tree composed of low-level controller actions (e.g. steer left / right, go straight), we utilized our evolutionary framework to automatically generate a decision tree consisting of high-level parameterized controllers as building blocks (see Table 1). Other components of this decision tree are conditional rules (IF-THEN-ELSE), standard boolean values and operators (TRUE, FALSE, AND, OR, NOT), conditional variables (see Table 2), program blocks (SEQUENCE), and velocity commands (SET-VELOCITY, SLOW/NORMAL/HIGH-VELOCITY, STOP).

The final decision tree expressing the blocking logic had to be general enough to cope with a broad variety of enemies. The key was to let the intruder develop its own attacking logic thereby competing with the USV's blocking logic. To generate a more general USV logic, we employed a simulated competitive co-evolutionary process [6] using which the USV's and intruder's logic was being improved simultaneously. During this process, the improved intelligent intruder was taking advantage of inefficiencies in the blocking logic used by the USV. It was gradually developing its own set of rules exploiting the weak points of the blocking logic.

The competitive co-evolution is a form of evolution in which the fitness function of an individual is completely dependent on other individuals. This means that fitness evaluation requires interaction between multiple individuals. We used two populations representing two different sets of individuals (USV and intruder decision trees) as they are made by different program primitives.

During the evolutionary process, a newly created USV individual is evaluated against a particular intruder in three different test scenarios. In each scenario, the intruder had a different initial orientation and distance from the target, and the USV always started from an initial position close to the target. The fitness function is defined as the squared distance between the intruder and the target over all time steps. This squared distance is normalized due to the different initial distances of the intruder from the target in the test scenarios. The fitness function is defined as

$$F = \frac{1}{T}\sum_{i=1}^{T}\left[\frac{d_i}{d_0}\right]^2 \qquad (3)$$

where T is the total number of time steps, d_i is the distance of the intruder from the target at time step *i*, and d_0 is the initial distance of the intruder from the target in a particular test case. When evolving the blocking logic for the USV we are trying to maximize its fitness value and vice-versa when evolving the logic for the intruder.

Due to the stochastic nature of the simulated evolution and various co-evolutionary dynamics [6], not all co-evolutionary runs are successful. To make the co-evolutionary run feasible in terms of probability of successfully evolving a solution and time spent on the computation, we had to (a) seed a portion of both of the initial populations (for the USV and intruder) by an initial set of simple human-crafted planning logic and (b) evaluate the USV blocking logic against an archive of previously encountered intruder champions from previous generations and vice-versa.

The evaluation process of an individual is based on an archive of previously encountered champions from the other population. Using this archive, the individual being evaluated is pitted against a randomly sampled set of best adversaries from previous generations. The final fitness value of the individual is then computed as an average of all fitness values resulting from each single competition.

The initially seeded individuals provide a baseline from which the co-evolution can start without having to spend much valuable time on evolving the basic functionality from scratch. This baseline also serves as a basic evaluation standard in further generations for penalizing suboptimal novel solutions. This means that a newly created individual is evaluated not only against the best adversaries from the archive but also against the baseline individuals. The initial values of all high-level controller parameters inside the initial set of randomly generated decision trees are randomly generated. They are further mutated in the course of the evolution process, possibly improving the functionality of their parental high-level controllers.

In order to pick the most successful blocking logic from a particular co-evolutionary run, we developed a tool analogous to the Master Tournament matrix as defined in [23]. We also needed to analyze whether a given co-evolutionary run produced logic which could cope with all previously encountered adversaries. This is due to the fact that the co-evolution is prone to various complex dynamics, such as overspecialization, which are often difficult to analyze. This tool allowed us to detect failures during the co-evolutionary search, identify similar phenotypes related to similar efficiency, identify early convergence to a particular type of logic, and identify possible breakdowns in the "Arms race" due to overspecialization.

Table 1. High-level controllers of reactive planning logic for USV and intruder

GO-INTRUDER/USV: FRONT, FRONT-LEFT/RIGHT, LEFT, RIGHT, BACK, BACK-LEFT/RIGHT
GO-STRAIGHT
TURN-LEFT/RIGHT
ENCIRCLE-INTRUDER/USV INTRUDER-SEEK-PROTECTED-OBJECT

Table 2. Conditional variables of reactive planning logic for USV and intruder

INTRUDER/USV: ON-THE-LEFT/FRONT-LEFT/FRONT-RIGHT/RIGHT, IN-FRONT, AT-THE-BACK/BACK-LEFT/BACK-RIGHT
USV/INTRUDER: PROTECTED-OBJECT-ON-THE-LEFT/RIGHT,
INTRUDER/USV: CLOSE

6. Virtual Environment-Based Game

We have developed a virtual-environment-based game to evaluate the automatically generated blocking logic represented as a decision tree as described in Section 5. The game software consists of two parts: a virtual environment and game logic itself. The virtual environment offers a realistic 3D immersive world by implementing physics based scene, incorporating rigid body dynamics, waves, and dynamic obstacles. The virtual environment also handles user input. User input consists of keyboard strokes, mouse clicks and movements, or a Microsoft XBox controller interface. The XBox controller offers a very intuitive and familiar user interface, especially for the experienced gamers. It is important to make the user interface as intuitive and as simple as possible so that the performance of the human operators, when compared to the performance of the computer reactive planning logic, is not biased by a poor user interface. The XBox interface allows the human player to not only control a boat, but also control the translation and rotation of the view and switch between different vantage points.

All the boats in the environment are represented with realistic 3D models created using a CAD tool. The ocean is rendered using a dynamically generated triangulated mesh that is linked to a height map. The same height map is used to calculate bobbing motion for all the boats in the environment. Up to a certain distance from the view point, the triangles which make up the ocean surface are constantly updated and redrawn based on the height map queries. The dynamic ocean mesh was broken up into a grid of tiles, where each tile represents an independent object. As the user's view moves and rotates, the virtual environment uses an efficient method to check which tiles are in view and which tiles are no longer within the view angle. Tiles which are not within the view angle are made invisible to prevent the system from rendering unnecessary triangles. The dynamic ocean implementation also supports multiple levels of detail (LOD). Ocean tiles with high LOD level have more triangles than those with a low LOD level. The detail level increases for the tiles that are closer to the view and decreases for those that are far from it. When the system signals an ocean tile to change its LOD level, the triangulated mesh for a specific tile is regenerated.

The game logic is responsible for the rules of the game, game logging and replay, boat behaviors, and scoring. Players must navigate the boats around various obstacles to perform their respective missions. Once models of all the boats are loaded into the scene, a physics-based dynamics model is used to govern the boat behavior. In human versus human mode the game is played on two computers over a network using User Datagram Protocol (UDP) and a client-server architecture. The game also supports the concept of a limited visual range. Each player is only able to see objects which are within a certain configurable radius of the boat they are driving.

The game supports collision detection for all objects in the environment so that objects are not allowed to pass through each other. Some basic boat physics has been implemented to deal with boat collisions. When two boats collide, they deflect each other depending on the collision point for each boat, each boat's velocity, mass, and each boat's direction of travel. The deflection involves both translation and rotation. Upon collision, each boat receives a certain amount of damage. Too much damage results in the sinking of the boat. The computer driven boat is equipped with visibility sensors used for avoiding obstacles.

The game also supports logging so that it can be played back from different perspectives for visual analysis and inspection. The logging system records the movements of all objects during the game.

The boats serving as obstacles have randomly generated structured motion. During the initialization of the environment, a randomly generated sequence of actions, as well as their duration, is built for each boat.

7. Evaluation

The virtual environment based game that we developed allows human players to play against each other or against the computer. In the game, the player controlling the intruder boat must reach a protected object as quickly as possible, while the player controlling the USV must block and delay the intruder for as long time as possible. This game allows us to compare the efficiency of the automatically discovered decision tree expressing the USV's blocking logic to the behavior exhibited by human players.

7.1 Evaluation Protocol

For our testing, we gathered four volunteers to play against each other and against the computer. In human to human testing, the four players took turns playing the role of the USV and the intruder. Each of the four players played nine games resulting in a total of 36 games played during the testing. In computer to human benchmarking tests the four players took turns playing the role of the intruder while computer played the role of the USV. In this second round of gaming, each volunteer played 3 times resulting in a total of 12 games played.

The efficiency of the logic was measured in terms of the time delay applied on the intruder by the USV as the USV carried out blocking logic to protect a target. The maximum time for the game was set to five minutes. The speed of the USV was set to be 25 per cent higher than the speed of the intruder.

7.2 Preliminary Results and Discussion

The results of our evaluation tests depicted in the graph in Figure 6 show that the human operators performed better in terms of delaying the intruder than the computer generated decision trees. The blocking logic currently being generated by the computer using many simulations is highly reactive in nature. In other words, the computer generates simple, short term plans in response to the intruder's actions. No long term planning or reasoning is performed by the computer. Human operators, on the other hand, are capable of perceiving long term outcomes such as congested regions predicted based on the current direction of travel and the speed of obstacles.

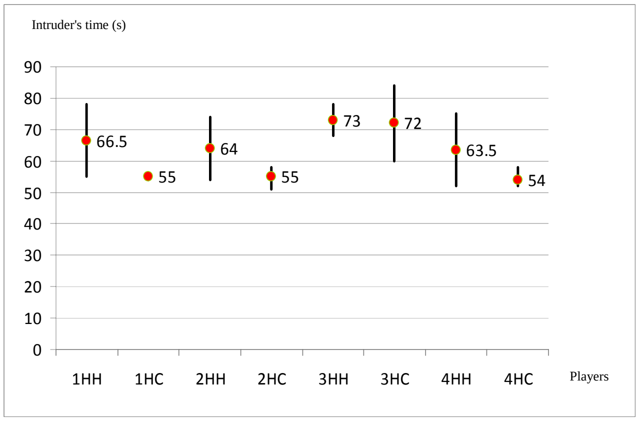


Figure 6. Evaluation results

The graph in Figure 6 shows the results of human to human and human to computer competitions. Each pair of vertical bars in the graph represents the scores for a particular human player, while that player controlled the intruder boat. The score represents the number of seconds it took for the human operated intruder to reach and collide with a protected target. A high number means that the intruder performed poorly and the USV performed well. The first bar in each pair (XHH) represents the scores for that player after playing against other human operated USVs. The second bar in each pair (XHC) shows the scores for this player after playing against the computer operated USV. For example, the first 1HH case shows the average time (66.5 s) player 1 took to get to the protected object when playing against USVs driven by other players. This number is computed as an average of three games played by intruder 1 against USV 2, USV 3, and USV 4. Each game played by intruder 1 against each USV consisted of three rounds. The second 1HC case shows how much time on average (55 s) intruder number 1 took to get to the protected object when playing against a USV driven by a computer. The bar associated with each number shows maximum and minimum values. These results show that human players consistently outperformed the computer in blocking another human player.

To get a fair assessment of computer performance versus human performance, the time values being compared must be normalized by 46 seconds baseline. This baseline represents the amount of time needed to reach the target if the intruder is completely unobstructed. Any additional time above this baseline thus represents the effective delay time of the intruder when being blocked by the USV.

During the testing, we noticed that the computer and the human players had different strengths and weaknesses. The computer was able to precisely and swiftly direct the USV into advantageous blocking positions around the intruder. This is mainly due to rapid reaction of the USV to sudden changes in intruder's velocity, position, and orientation. The computer was also more efficient at taking the shortest path to a waypoint, while the human players tended to weave side to side. This is due to the underlying precision of the computer's control system compared to the relatively imprecise control of the human players. Humans on the other hand had the ability to perform better long term prediction. For example, humans were better able to observe a set of moving obstacles and predict natural barriers for the intruder as a result of high congestion and traffic. A human could then stay closer to less congested routes to the protected target. Since the current type of the planning logic is not able to perform long term planning, this capability was not available to the computer. Human players were also able to spot repetition in computer's behavior and take advantage of it. For example, once some of the human players noticed that the computer tends to over shoot a little in the process of blocking, they would often suddenly slow down and turn toward the stern of the blocking USV, passing the USV from behind.

We noticed a number of blocking logic emerging during the human to human testing. In the human to human testing, the best blocking logic for the USV was to block the intruder far from the target. By doing this, the USV found more possibilities for blocking the intruder and had more time to recover in case of a mistake. It was also beneficial for a player controlling the USV to stay close to the protected object and wait for the intruder. Once the intruder approached the base, the USV would start to actively block it. In order for this to work, however, the human player controlling the USV had to maintain a certain distance from the protected object and be skillful at controlling the boat. Otherwise, the intruder could take advantage of the USV's close proximity to the base and lack of adequate space for maneuvering. The blocking logic, in this case, aimed to predict the future position of the intruder and use that prediction to make sure that USV was constantly between the intruder and the protected base. The intruder's solution in this case was to drive at high speed around the base, waiting for the USV to make a mistake and leave an opening. Figure 7 shows an example from a game played by a human intruder against a computer-driven USV during which the USV utilizes a blocking maneuver in front of the intruder.

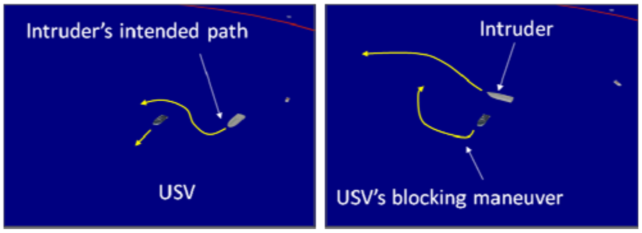


Figure 7. Example of a game played by a human intruder against a computer driven USV

8. CONCLUSIONS AND FUTURE WORK

We have developed a mission planning system for automatically generating a reactive planning logic for USVs. The planning logic is represented as a decision tree which consists of high-level controllers as building blocks, conditionals and other program constructs. We used our strongly-typed GP-based evolutionary framework for automatic generation of planning logic for blocking the advancement of an intruder boat toward a given target. We employed a simulated competitive co-evolutionary process to improve the general capability of this blocking logic by pitting the autonomous USV against a set of different, simultaneously evolved enemies.

An important part of our work was to fairly assess the performance of the blocking logic. For this reason, we developed a virtual environment based game to compare the efficiency of the computer generated planning decision tree to planning skills of human operators.

The results of our tests show that human players driving the USVs consistently outperformed the computer driven USV in blocking another human driven intruder. From purely reactive point of view, the performance of humans and machine were quite similar. However, the human players demonstrated better long term planning and were better at discovering weaknesses in the computer planning logic. We are planning to combine deliberative planning with predictive capabilities in the next round of testing.

We recently started testing physical radio controlled boats so that we can evaluate the generated planning logic in a real environment. Testing in a real environment adds some additional challenges due to the differences with the virtual environment. We are planning to improve the virtual environment based boat simulations (high and low-level controllers, environment dynamics, sensors, etc.) inside the mission planning system in order to generate a planning logic reflecting the real world dynamics.

9. ACKNOWLEDGMENTS

This research has been supported by Office of Naval Research and NSF Grants OCI-0636164 and CMMI-0727380. Opinions expressed in this paper are those of the authors and do not necessarily reflect opinions of the sponsors.

10. REFERENCES

1. Larson, J., M. Bruch, and J. Ebken. *Autonomous Navigation and Obstacle Avoidance for Unmanned Surface Vehicles*. in *Proceedings of SPIE Unmanned Ground Vehicle Technology VIII*. 2006.
2. Larson, J., et al., *Advances in Autonomous Obstacle Avoidance for Unmanned Surface Vehicles.* AUVSI Unmanned Systems North America, 2007.
3. Cornfield, S. and J. Young, *Unmanned Surface Vehicles - Game Changing Technology for Naval Operations*, in *Advances in Unmanned Marine Vehicles*, G.N. Roberts and R. Sutton, Editors. 2009, The Institution of Electrical Engineers: Stevenage, United Kingdom.
4. Koza, J.R., et al., *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. 2003: Kluwer Academic Publishers.
5. Koza, J., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. 1992: MIT press.
6. Miconi, T., *The Road to Everywhere: Evolution, Complexity and Progress in Natural and Artificial Systems*. 2007, School of Computer Science, The University of Birmingham. p. 197.
7. Axelrod, R., *The Evolution of Strategies in the Iterated Prisoner's Dilemma.* Genetic algorithms and simulated annealing, 1987: p. 32-41.
8. Reynolds, C. *Competition, Coevolution and the Game of Tag*. 1994.
9. Jaskowski, W., K. Krawiec, and B. Wieloch, *Winning Ant Wars: Evolving a Human-Competitive Game Strategy Using Fitnessless Selection.* Lecture Notes in Computer Science, 2008. **4971**: p. 13.
10. Shichel, Y., E. Ziserman, and M. Sipper. *GP-Robocode: Using Genetic Programming to Evolve Robocode Players*: Springer.
11. Sipper, M., et al., *Designing an Evolutionary Strategizing Machine for Game Playing and Beyond.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2007. **37**(4): p. 583-593.
12. Togelius, J., P. Burrow, and S. Lucas. *Multi-Population Competitive Co-evolution of Car Racing Controllers.*
13. Doherty, D. and C. O'Riordan. *Evolving Agent-Based Team Tactics for Combative Computer Games.* 2006.
14. *National Robotics Challenge*. 2009 [cited 2009 May]; Available from: http://www.nationalroboticschallenge.org.
15. *US FIRST*. 2009 [cited 2009 May]; Available from: http://www.usfirst.org.
16. *Eurobot*. 2009 [cited 2009 May]; Available from: http://www.eurobot.org.
17. *Robogames*. 2009 [cited 2009 May]; Available from: http://www.robogames.net.
18. *RoboCup*. 2009 [cited 2009 May]; Available from: http://www.ai.rug.nl/robocupathome.
19. *International Aerial Robotics Competition*. 2009 [cited 2009 May]; Available from: http://iarc.angel-strike.com.
20. *DARPA Urban Challenge*. 2009 [cited 2009 May]; Available from: http://www.darpa.mil/grandchallenge.
21. Freed, M., R. Harris, and M. Shafto. *Measuring Autonomous UAV Surveillance Performance*. in *Proceedings of PerMIS '04*. 2004. Washington, D.C.
22. Brooks, R.A., *Intelligence Without Representation.* Artificial Intelligence, 1991. **47**: p. 139-159.
23. Nolfi, S. and D. Floreano, *Coevolving Predator and Prey Robots: Do "Arms Races" Arise in Artificial Evolution?* Artificial Life, 1998. **4**(4): p. 311-335.

Fork Lift Awareness

Mark E. Austin
Occupational Safety and Health Administration
Baltimore Washington Area Office
1099 Winterson Road, Suite 140
Baltimore, MD 21090
mark.austin@dol.gov

ABSTRACT

In this paper, we discuss the Occupational Safety and Health Administrations' statistics on fork lift accidents and injuries caused by these accidents. Fork lift benefits and the operating environments where these manned machines work are considered. Methods used to reduce fork lift accidents as well as reported cases of lost time at work due to fork lift accidents will be discussed, along with the percentage of lift truck accident causes.

Categories and Subject Descriptors

B.0 [HARDWARE/GENERAL]: Fork Trucks;

K.m [MISCELLANEOUS]

General Terms

Human Factors, Performance

Keywords

Fork lift, accidents, injuries, lost work time, OSHA, pedestrians

1. PRESENTATION

Fork Lift Accidents

- OSHA estimates that there are 110,000 accidents each year.
- Approximately 31,600 employees suffer some type of injury.
- Losses affect employees through physical and mental suffering.

Benefits of Fork lifts

- Assist in the movement of materials
- Reduce employee injuries

Fork Lift operating environments include:

- Pedestrians
- Blind spots
- Indoors/Outdoors
- Narrow aisles
- Building columns
- Operate 24 hours per day
- Turning radius

Fork Lift and Pedestrians

- Pedestrians contribute to accidents
- Pedestrians do no understand stopping distances
- Pedestrians tried to "beat" a lift truck

Methods used to reduce Fork Lift Accidents

- Training of drivers
- Maintenance of equipment
- Areas of operation

Fork Lift Accidents

- Losses affect employers
    - damage to equipment
    - loss productivity

A breakdown of the 1,158,870 Report Lost Time Cases in 2007:

- 11,040 Involved Fork Trucks
- 5,730 Involved Transportation and material moving
- 1,630 Involved Production Worker
- 1,360 Involved Office Or Administrative Workers

Source: Powered industrial truck accidents report through Bureau of Labor Statistics 2007

Type Body Parts Affected:

Lower Extremities	6,040
– Foot, Toe	3,290
– Knee	400
Trunk	1,780
Back	1,090
Shoulder	280

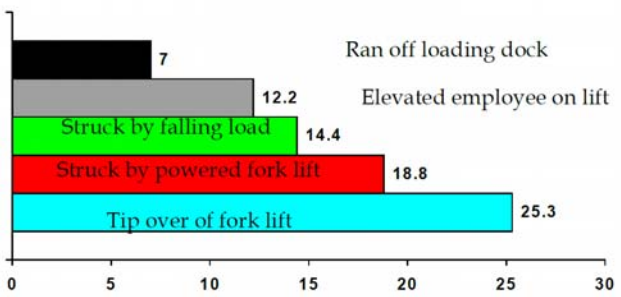
Event or Exposure

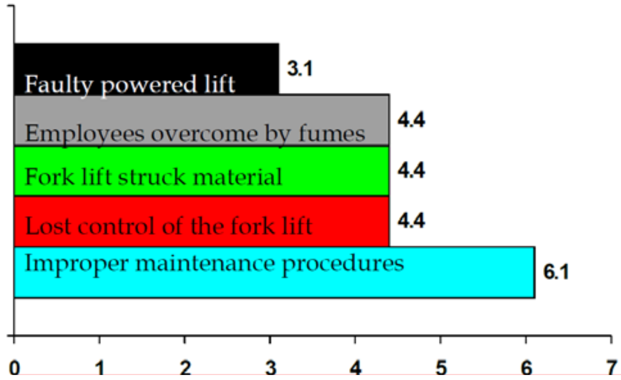
Contact with Object, Equipment	4,540
– Caught in object, equipment	1,630

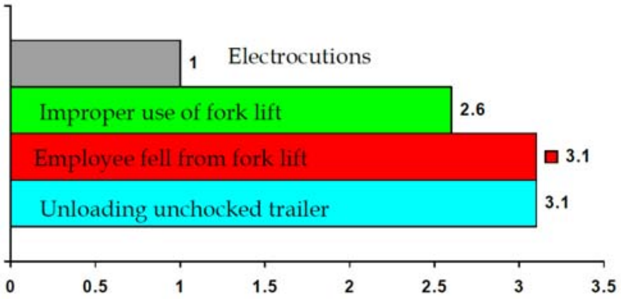
PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
ACM 978-1-60558-747-9/09/09

–	Struck against object	1,600
–	Struck by object	1,000

In percent what Causes Lift Truck Accidents?







Where AGV’s and Forklifts Roam:
Preserving Operational Safety in a Shared Workspace

Richard H. Ungerbuehler
Co-Founder and Chief Engineer

Sky-Trax Incorporated
98 Quigley Blvd.
New Castle, DE 19720
USA
(302) 395-9540

RHU@Sky-Trax

Operational safety can be assured in facilities where industrial utility vehicles are used to transport goods only when safe practices are clearly established and carefully monitored. For example, forklift trucks, pallet jacks, buggies and carts can safely share operational space if safety is designed into the operation and vehicle operators follow safety rules and remain alert.

In facilities where autonomous vehicles are used, a different set of safety requirements is needed. Autonomous vehicle control systems must assure that inter-vehicular collisions are prevented, and the vehicles must be equipped with safety devices, such as laser bumpers, to prevent collisions with people or equipment.

Shared Operational Space

When considering the design of a new storage facility, designers may choose between implementing a manual operation using driven vehicles such as forklifts, or a fully automated operation that employs autonomous vehicles or an automated storage and retrieval system. By making a choice of one or the other, managers commit to a non-reversible decision to invest capital in a facility with predestined operational efficiency.

In the event that a company wishes to utilize driven and autonomous vehicles in a shared space, two basic safety strategies exist: (1) separate by subdividing and sharing space, (2) separate by sharing time. Facilities can be subdivided into multiple areas; perhaps one where humans work, another where driven vehicles are permitted, and another where automated vehicles operate. This strategy results in a high degree of safety, but often complicates operations and significantly limits efficiency. Another example would be to separate driven vehicles from autonomous vehicles by utilizing forklifts during one time period and autonomous vehicles during another period. As an example, imagine a warehouse that allows powered vehicles to operate during the day shift, while precluding pedestrians from entering the facility. Then, on the night shift, no vehicles are operated, while workers have full access to the facility. This plan implements a time-shared facility.

An alternative would be the space- shared facility, where workers, forklifts, and autonomous vehicles operating during both shifts, but impervious barriers separate the three. Neither arrangement proves very practical in the industrial setting. A combination of time and space separation is needed, and that’s where modern technology steps in.

Collision Avoidance: The Time/Space Problem

Traditionally, collision avoidance strategies that were designed to separate driven vehicles, autonomous vehicles, and people, have relied on facility design, safety procedures, and driver training and compliance. While these strategies will always be elements of workplace safety programs, collision statistics clearly indicate that training, signage, and floor markings for traffic control are not enough to assure a safe environment. Avoiding collisions between powered vehicles, or between pedestrians and vehicles, is based on a simple principal; all pedestrians and all vehicles must be kept separated in time and space.

The key to understanding the safety versus efficiency dilemma is to understand that time separation and space separation, which have always been effective, are inefficient, while simultaneous, real-time monitoring and control can improve both safety and efficiency.

Safety Makes Good Business Sense

Leading businesses know that safety is not incompatible with efficiency – instead, it can improve efficiency and enhance productivity. These companies use safety as a competitive advantage. Since the majority of serious accidents involve stability incidents and vehicular collisions with pedestrians, the new safety systems solution will integrate technology addressing both problems with intelligent speed control, vehicle tracking, and pedestrian tracking that can provide both improved safety and increased productivity.

With the possibility of improving efficiency, a strong business case can be made for investing in safety systems and technology to eliminate hazards and prevent accidents.

Collision Avoidance: A Technological Approach

Technological solutions are now becoming available to allow safe operations in facilities where mixed manual and automated equipment is used; for example, in a factory or warehouse where driven and autonomous vehicles co-exist.

Sensor technology today can provide the ability to detect and track the location and proximity of vehicles and pedestrians in industrial facilities. Further, sensors can work in localized areas, over large areas, or throughout entire facilities. Whether tracking pedestrians or trucks, the best safety technologies will have the following capabilities:

- Location determination accurate to within a meter or less;
- Velocity determination;
- Determination of orientation or direction of travel;
- Ability to identify individual vehicles.

Technology for Detecting Pedestrians

To systematically improve safety, safety engineers recommend the following strategy for hazard control: 1) Remove, 2) Guard & 3) Warn. Strategy 1 strives to remove the hazard by eliminating the danger by designing it out of the system or environment. For example, closing a blind door that opens into a warehouse can eliminate the possibility of a pedestrian walking into forklift traffic.

If the hazard cannot be eliminated, Strategy 2 is to guard the hazard by installing safety apparatus to prevent exposure to the danger. Even if pedestrians and drivers are distracted, guards can protect both from harm. For example, automatic barrier guards can be installed to prevent fork trucks from falling off a vacant receiving dock. A good example is the Rite-Hite RHH dock leveler with the Safe-T-Lip™ barrier[1], which prevents forklifts from running off an open dock and can stop a 10,000 lb. forklift traveling at up to 4 mph.

Finally, if the hazard cannot be eliminated, or exposure to the danger prevented, Strategy 3 is to warn workers of the hazard. Ideally, specific alerts should be communicated only to those directly involved in the hazard situation and only where and when a danger actually exists. For example, warning lights can be installed at blind corners to warn of oncoming forklifts with a system like the Wickham Fork-Alert™ product[2].

Safety system designers now have new technologies to consider for hazard control, particularly for detecting collision and speeding hazards.

Two types of pedestrians can be found in industrial settings; employees who work in an area with vehicular traffic, such as a hand truck operator or an order picker, and visitors, who are less likely to understand, remember, and comply with safety requirements. Safety system designers must address the risk and establish effective measures for both classes of pedestrians.

The Accident Research Centre at Monash University (Victoria, Australia), a leader in evaluating technology for preventing forklift-pedestrian accidents, has developed a system employing a simple RF-tag placed in safety vests worn by warehouse workers[3]. An RF receiver was installed on each truck alerting drivers to the presence of any workers within the detection radius of the receiver. The researchers found this wearable RF tag prototype to be a low cost solution that they recommend to be used along with other safety measures.

A product from ProxAlert[4] takes the Monash University prototype concept to the next step. ProxAlert places an RF transceiver on each vehicle. A similar battery-powered portable transceiver is clipped onto any pedestrian entering the warehouse. As illustrated; the transceiver range creates a virtual protection zone around the vehicle or person. When the zones intersect, the transceivers energize a warning signal for both the pedestrian and the vehicle operator. This approach is a viable solution for workers and pedestrians.

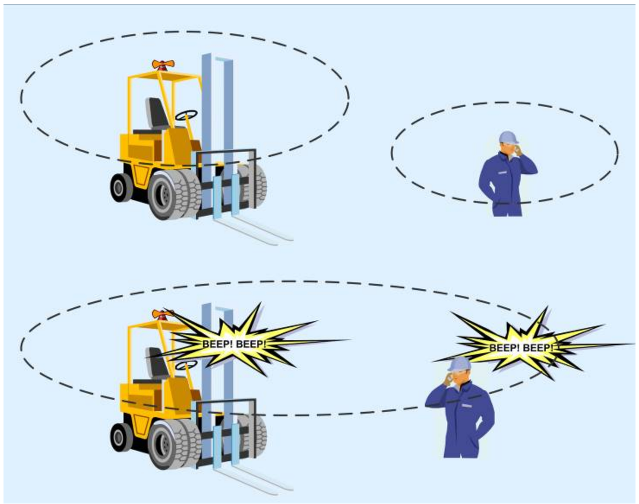


Figure 1. Proximity Detection and Alert

Tracking Pedestrians with Machine Vision

Over the last 10 years, image processing research has made great progress in developing methods for detecting, identifying, and tracking people in video images. Driven largely by the need for smart surveillance and security systems, the technology has moved beyond military uses and is now used in commercial applications. Brickstream (Atlanta, GA) has marketed a pedestrian tracking system since 2002 that tracks and analyzes the movement of customers in commercial buildings[5]. Processing images from overhead cameras, the system determines the number of customers entering a store and the exact paths taken by customers shopping in the store. In retail and banking applications, the technology is used to track queues of customers and to signal when more check out lanes need to be opened.

While this technology has not yet been applied to collision-avoidance systems, it can be expected in the near future. Because industrial spaces are less diverse and more orderly than public areas like streets and stores, the application of machine vision for pedestrian tracking in warehouses should be very feasible. Location accuracy is likely to be less than a meter, and the pedestrian's direction of movement and speed will also be provided. These capabilities are needed for detecting pedestrian hazards in areas where forklifts operate.

Technology for Detecting Vehicles

A wide spectrum of technologies can be used to detect industrial utility vehicles. These can be functionally categorized as follows:

- Presence Detection
- Presence & Distance Detection
- Presence & Identification
- Location and Tracking

Presence Detection

Presence detection sensors indicate that a vehicle is within the detection distance or zone of the sensor. In most cases, there is some ability to configure or engineer the detection distance. Inductive or capacitive proximity sensors and photoelectric sensors, all of which are familiar to automation engineers, fall into this category.

Proximity sensors detect the presence of a large metal mass like a truck within their detection range – usually less than 1-2 meters. This short detection range makes this type of sensor most applicable for detection at "chokepoints" such as dock doors. Photo detector sensors are also used for this purpose. Wickham Fork-Alert ™ and Alert Safety Products offer safety products for the warehouse based on this technology. Fork-Alert employs an invisible infrared light beacon mounted on the top of the vehicle. An infrared receiver can detect the infrared light up to 25 meters away and trigger warning lights or audible alarms for pedestrians and other drivers. Alert Safety Products combines the light source and the detector in a single unit that is mounted on a wall or post. Strips of reflective tape are applied to both sides of forklifts so the vehicles can be detected when traveling by the sensor.

Microwave sensors work similarly and can shape the detection zone to match an area of interest. For example, products from Door-Man[6] and Alert Safety Products[7] offer warehouse intersection warning products using microwave sensors. Four sensors and a warning light are hung above an intersection with microwave sensors aimed in all four directions. A vehicle approaching the intersection is detected and triggers the appropriate warning light.

Presence and Distance Detection

The next class of sensors not only detects a target but can accurately measure the distance from the sensor to the object. The principal technologies here are ultrasonic range sensors and laser time-of-flight sensors. Ultrasonic sensors emit high-frequency sound waves which are too high for the human ear to hear. When waves are reflected back from a solid object, the sensor can determine the distance from a few centimeters up to 10 meters.

Laser systems can measure distances with higher accuracy and longer ranges. Found typically in high-end safety systems on automated guided vehicles, these sensors measure distances very accurately with time-of-flight calculations on the reflected laser light. A commercial safety laser scanner from SICK GmbH[8] can be programmed for different scanning areas and distances and configured to have both warning and emergency stop thresholds.

Presence and Identification

RFID technology has received extensive press for inventory tracking applications in warehouses. Typical applications use passive RFID with inexpensive tags that can be read (detected and identified) by an RFID reader, but the read distance is small - usually less than a meter. Longer read distances of up to tens of meters are possible with active RFID systems. These systems detect and identify a tagged entity within the proximity of the RFID reader. This capability has been employed widely for security and access control applications.

Location and Tracking Systems

Systems that can accurately track industrial vehicles will have great impact on creating the next generation of safety systems for warehouse operators. This new technology is known as real time location systems (RTLS). Radio frequency RTLS and optical RTLS systems are available today.

Radio frequency RTLS tracks vehicles that carry an RFID tag. The identifying tag can be read simultaneously by multiple RF receivers in the detection region. Using one of several different sensing algorithms, a high speed computer applies triangulation techniques and computes a location estimate for any tag that is read by three or more sensors. Overall accuracy in industrial buildings currently is approximately 2 to 10 meters. Leading suppliers of RF RTLS technology include AeroScout[9], WhereNet[10], and Ekahau[11].

The latest technology for tracking vehicles in warehouses is machine vision for optical RTLS. Machine vision (image processing) has been used widely in industrial automation for high speed package sortation, automated product inspection, and robotic guidance for the past 20 years. Sky-Trax Inc. has adapted this technology to provide accurate and reliable tracking of industrial vehicles inside buildings. With the Sky-Trax Indoor Position Sensing™ (IPS) technology[12], vehicles are tracked in real time to accuracy of 5-20 cm. Important to many safety applications, IPS systems determine the instantaneous speed and orientation (heading or direction of travel) of each tracked vehicle.

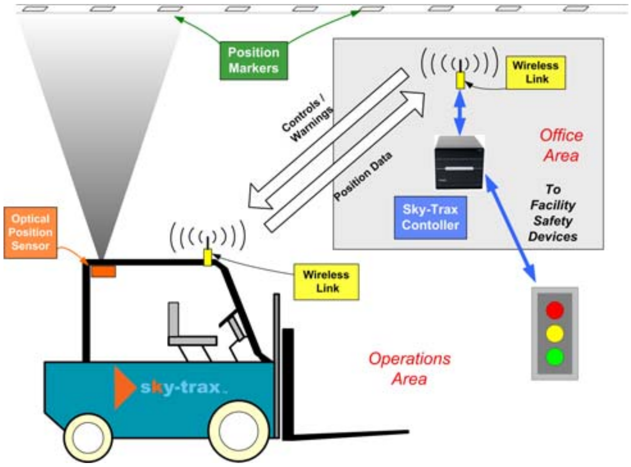


Figure 2. Indoor Position Sensing system

IPS (Figure 2) employs a small imaging sensor mounted on each vehicle that views upward toward the ceiling, where an array of encoded position markers is visible. The imaging sensor includes image processing intelligence to capture and analyze pictures of the ceiling several times per second. From ceiling scene analysis, IPS calculates X and Y position as well as angular direction of the vehicle. Velocity data is calculated from location changes noted from frame to frame. Position data are transmitted wirelessly to a computer which collects data on the location and status of all vehicles in real time.

Speed Control for Safety

Speed is usually a contributory factor to both collisions and stability-induced incidents (tip-over) which together represent well over half of all serious accidents. Monash University researchers report that 75% of side tip-over's occur when a forklift is empty, leading them to conclude that these incidents are due more to speeding than other causes. Systems to control speed will be a significant way that technology will improve safety. The best solutions will do this without impacting productivity.

Like automobiles, forklifts cannot safely stop on a dime; and panic stops create additional hazards with loss of handling capability and unstable loads. Stopping distance is a function of speed, mass, driver reaction time, driving surface conditions, and braking system performance. Speed and reaction time are the key variables that can be controlled by the driver, and are the two areas where technology can help. Technology can provide advance warning of hazards (earlier reaction time) and can directly limit speed to assure adequate stopping distance based on location, load, vehicle type, and known hazards.

Safety system analysis begins with understanding safe stopping distances. If an empty forklift truck moving at 10 mph requires 40 feet for a safe stopping distance, the driver needs to allow at least 40 feet to react once a pedestrian hazard is recognized. If this is not practical, for instance at blind intersections, speeds need to be reduced in order to allow for proper stopping distance.

Speed limits are established by rule and drivers are expected to recognize and obey the appropriate speed limits. As on the highways, these rules are often violated and difficult to enforce. Many forklifts do not have a speedometer, and speed limits are not always posted. Complicating the situation, the safe speed changes as load mass changes, and driving surface conditions vary at different locations in the facility. Unfortunately, drivers sometimes experience more pressure to be efficient (drive fast) than to be safe.

Some believe that installing speed limiters on trucks is the solution to control speeding; however, this recreates the safety versus efficiency dilemma. While reduced speeds are necessary in some areas and conditions, speed reductions are an unnecessary restraint on productivity in other areas and circumstances. Speed control must balance productivity with safety to permit a vehicle to travel at the fastest safe speed for the specific location and conditions. Allowable speed must vary as the vehicle moves from location to location and as conditions change. This can be accomplished with technology that monitors the conditions, location, direction, and speed of the vehicle and of all the other vehicles and pedestrians in the area.

The Intelligent Safety System

The Intelligent Safety System (ISS) will include a direct means for alerting drivers and pedestrians when hazards exist and a direct means of automatically limiting speed. ISS will utilize data collected from on-board sensors and facility monitoring systems to:

- Accurately track the location, direction, and speed of all vehicles;
- Accurately track the location and movement of all pedestrians;
- Know the status of each vehicle (driver ID, load, current task, impact events, etc.)

Given an abundance of real-time data, ISS intelligence will predict collision hazards and initiate action to warn or eliminate hazards. ISS will have communication and control links with drivers for hazard alerting, with trucks for automatic speed limiting, and with facility safety systems for intersection control and other intelligent warning systems.

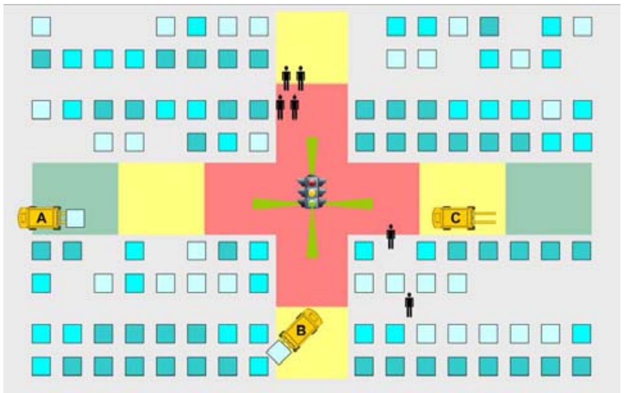


Figure 3. A Basic Intelligent Safety System (ISS)

Figure 3 illustrates a simplistic ISS. Safety zones are defined and configured in ISS software; for example, safe, caution, and danger zones are designated for intersections with the green, yellow, and red shading in the illustration. As truck A approaches the intersection from a safe zone ISS, which controls the intersection traffic lights, gives a green light to Truck A and to pedestrians, even though Truck B and Truck C are in the intersection's caution zones. With knowledge of the exact location, direction of travel, and speed of vehicles and pedestrians, ISS determines that Truck C is moving away from the intersection and presents no danger to vehicles A or B, or to the pedestrians. Likewise, Truck B's orientation and speed indicate that it is putting away a load and not entering the intersection.

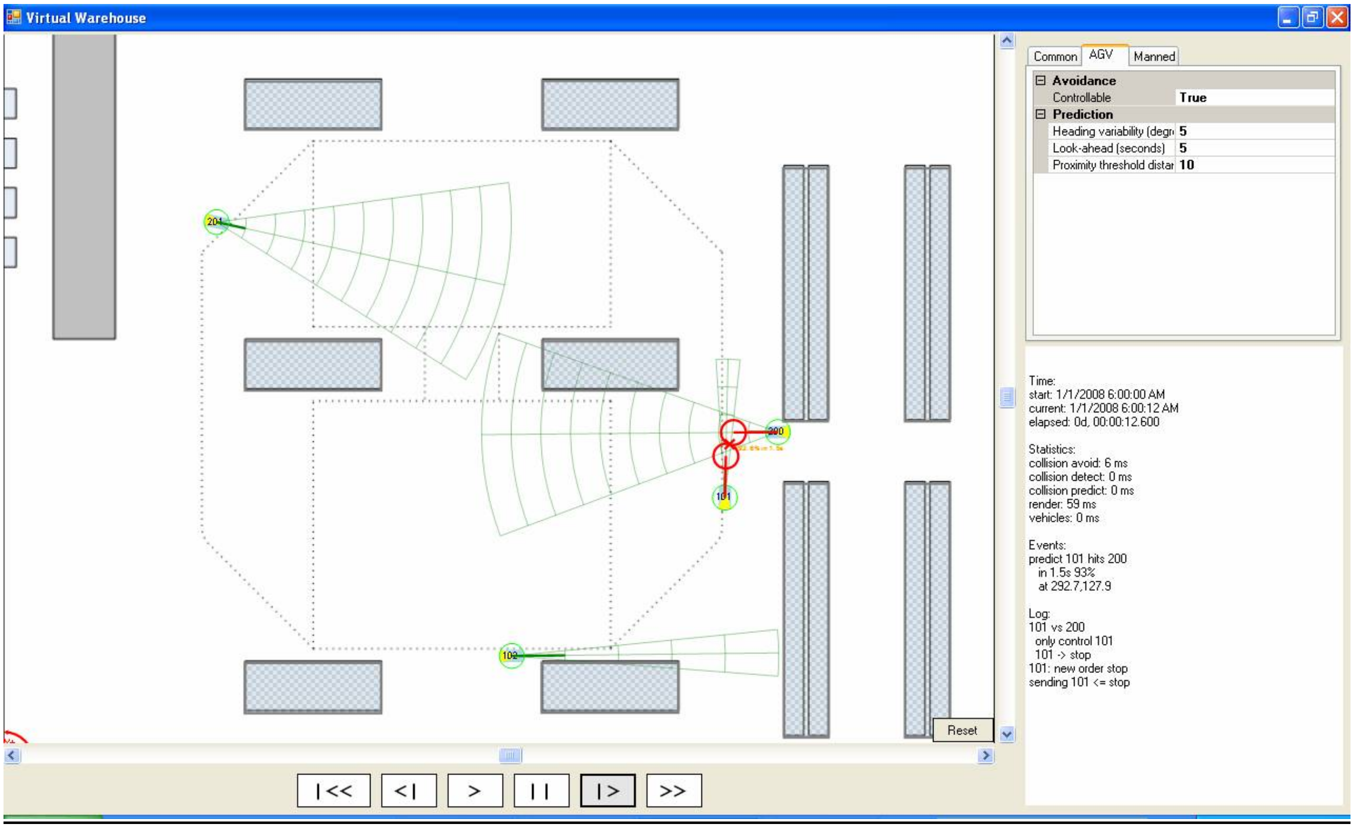
An unintelligent intersection safety signal system would illuminate caution or stop signals for Truck A based on the proximity of Truck B and Truck C. Importantly, ISS preserves productivity - Truck A will not be slowed - while establishing a safer workplace.

Figure 4 illustrates an example of collision prediction based on a more sophisticated ISS. Four utility vehicles are tracked second-by-second by IPS while ISS makes instantaneous calculations of collision potential. Vehicle location, speed and heading are fed into the ISS "engine", which uses the operating parameters for each vehicle and the facility layout to predict collision probabilities. When warranted, control or warning signals are transmitted to an AGV to slow or stop the vehicle, or to a forklift operator for driver action.

Safety Wins

Perhaps highway safety will someday become as safe as the ISS environment. Technology will continue to offer the ability to reduce or eliminate vehicle stability accidents, multiple vehicle collisions, and vehicle/pedestrian collisions, using new sensors and intelligent, automated safety solutions. As this progress is increasingly recognized by industry leaders, regulatory agencies, and safety researchers, it is expected that industrial vehicle manufacturers will incorporate the new safety technology into their products, and that market-leading companies will take advantage of the new capabilities.

Everyone is in favor of increased safety, especially when it enhances productivity and the bottom line. Technology cannot replace the basics of safety – strong management commitment, good operations design, training, and accountability. But when safety is introduced into a safety conscious culture, technology will provide the tools for transforming safety into a competitive advantage.



⊟ **Avoidance**	
Controllable	**True**
⊟ **Prediction**	
Heading variability (degr	**5**
Look-ahead (seconds)	**5**
Proximity threshold distar	**10**

```
Time:
start: 1/1/2008 6:00:00 AM
current: 1/1/2008 6:00:12 AM
elapsed: 0d, 00:00:12.600

Statistics:
collision avoid: 6 ms
collision detect: 0 ms
collision predict: 0 ms
render: 59 ms
vehicles: 0 ms

Events:
predict 101 hits 200
  in 1.5s 93%
  at 292.7,127.9

Log:
101 vs 200
 only control 101
 101 -> stop
101: new order stop
sending 101 <= stop
```

Figure 4. A Sophisticated Intelligent Safety System (ISS) for Inter-Vehicular Collision Prediction

[1] Rite-Hite (Milwaukee, WI) RHH dock leveler with the Safe-T-Lip™ barrier; www.ritehite.com

[2] Wickham (Victoria, Australia) Fork-Alert™; www.wickhamplastics.com

[3] Industrial Forklift Trucks – Dynamic Stability and the Design of Safe Logistics, 2003; Accident Research Centre at Monash University, Australia.

[4] ProxAlert (Phoenix, AZ); www.proxalert.com

[5] Brickstream (Atlanta, GA); www.brickstream.com

[6] Door-Man (Auburn Hills, MI); www.door-man.com

[7] Alert Safety Products (Cincinnati, OH); www.alertsafetyproducts.com

[8] SICK GmbH (Waldkirch, Germany); www.sick.com

[9] AeroScout (Redwood City, CA); www.aeroscout.com

[10] WhereNet (Oakland, CA); http://zes.zebra.com

[11] Ekahau (Reston, VA); www.ekahau.com

[12] Sky-Trax Inc. (New Castle, DE); www.sky-trax.com

---- < *End* > ----

Contact

For more information or discussion on this topic, contact Richard Ungerbuehler by email at RHU@sky-trax.com, or visit www.sky-trax.com.

Credits

Some content was originally published by Larry Mahan in an article for Industrial Utility Vehicle magazine. The author wishes to thank Mr. Mahan and the staff of Sky-Trax for their help in preparing this paper and the accompanying slide presentation.

Abstract

Everyone wants to work in a safe environment where no hazards exist, close calls never happen, and personal injuries and property damage simply do not occur. As companies strive to maintain safe operations by establishing safe practices, and using accepted safety controls and equipment as necessary for their industry, new technology is helping to bring a new level of safety to facilities where driven and automated vehicles operate.

Author's Bio

Richard H. Ungerbuehler is a founding partner and Chief Engineer of Sky-Trax Inc. Rich enjoyed a twenty five year career with DuPont where he was involved with the development of specialized product inspection instrumentation and automation systems. He led the internationally recognized DuPont DART team, which coordinated data collection applications throughout the company. Rich represented DuPont on international bar code standards organizations, and participated in the authoring of ANSI MH10.8 – the American National Standard for Bar Code Shipping Labels. He co-founded Sky-Trax with Larry Mahan and David Emanuel in 2004 and serves today as Chief Engineer. Rich received a Bachelor of Science degree in Electronic Engineering Technology from Spring Garden College and he holds various meritorious awards and certifications from the United States Army Signal Corps.

Performance Measurements Towards Improved Manufacturing Vehicle Safety

Roger Bostelman
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
(301) 975-3426
roger.bostelman@nist.gov

Will Shackleford
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
(301) 975-4286
shackle@nist.gov

ABSTRACT

In this paper, we describe the current 2D (two dimensional) sensor used for industrial vehicles and ideal sensor configurations for mounting 3D imagers on manufacturing vehicles in an attempt to make them safer. In a search for the ideal sensor configuration, three experiments were performed using an advanced 3D imager and a color camera. The experiments are intended to be useful to the standards community and manned and unmanned forklift and automated guided vehicle industries. The imager that was used was a 3D Flash LIDAR (Light Detection and Ranging) camera with 7.5 m range and rapid detection. It was selected because it shows promise for use on forklifts and other industrial vehicles. Experiments included: 1) detection of standard sized obstacles, 2) detection of obstacles with highly reflective surfaces within detection range, and 3) detection of forklift tines above the floor. We briefly describe these experiments and reference their detailed reports.

Categories and Subject Descriptors

I2.10 [**Vision and Scene Understanding**]: 3D/stereo scene analysis

General Terms

Performance, Design, Experimentation, Standardization

Keywords

3D Flash LIDAR, Forklifts, Powered Industrial Trucks, Automated Guided Vehicles (AGV), ANSI/ITSDF B56.5

1. INTRODUCTION

The National Institute of Standards and Technology (NIST), Intelligent Systems Division (ISD) has been performing measurements to be used as background information for advancing standards and for the manned and unmanned vehicle and sensors industries in an attempt to make forklifts and other vehicles safer. The Occupational Safety and Health Administration states: [1] “Each year, tens of thousands of injuries related to powered industrial trucks (PIT), or forklifts, occur in US workplaces. Most incidents also involve property damage, including damage to overhead sprinklers, racking, pipes, walls, and machinery. Unfortunately, most employee injuries and property damage can be attributed to lack of safe operating procedures, lack of safety-rule enforcement, and insufficient or inadequate training.” The statement suggests the need for improving driver’s knowledge, although safer vehicles can also help. Obstacle detection sensing that completely surrounds the vehicle could augment the driver’s or autonomous vehicle’s environmental awareness. Driver alerts and/or autonomous slow or stop vehicle operations are then possible based on this sensor information and, therefore, could provide safer vehicles.

NIST ISD has been working for several years with the Industrial Truck Standards Development Foundation (ITSDF) which manages “ITSDF B56.5 Safety Standard for Guided Industrial Vehicles and Automated Functions Of Manned Industrial Vehicles” [2] as approved by the American National Standards Institute (ANSI). NIST’s involvement with the B56.5 standard includes performance measurements of advanced non-contact sensors for automated guided vehicles (AGVs), and has led to proposed changes to the standard. AGVs are typically programmed to follow prescribed paths but still need sensors to detect obstacles such as closed doors, equipment, personnel or material left temporarily in the vehicles’ paths. Currently, they rely heavily on 2D line scanners, while some are equipped with a physical bumper as the final backup to stop the vehicle. The 2D line scanners work well with ground-based vertical obstacles but it takes many sensors to completely protect against overhanging obstacles and even then they do not scan the full volume of space through which the vehicle travels. Figure 1 shows how 2D line scanning sensors are typically oriented on vehicles to aid detection of overhanging obstacles. The lower triangular detection region near the vehicle would be undetected at vehicle startup. The longer upper non-detect area is never detected and the sensors themselves may be struck by overhanging obstacles, like a crane hook. Side and top sensors in the figure could themselves become obstacles if mounted as shown. The red dotted lines depict sensor scan-lines. From the side view, it is clear that 2D sensors of this type may not detect obstacles that are not within the sensor scan-line or have already passed through the scan-line. As shown, the scanner can miss an overhanging obstacle completely when directly in front of

PerMIS'09, September 21-23, 2009, Gaithersburg, MD, USA.
ACM 978-1-60558-747-9/09/09

the sensor housing, or if the obstacle is detected, it may be too late to stop or slow the vehicle.

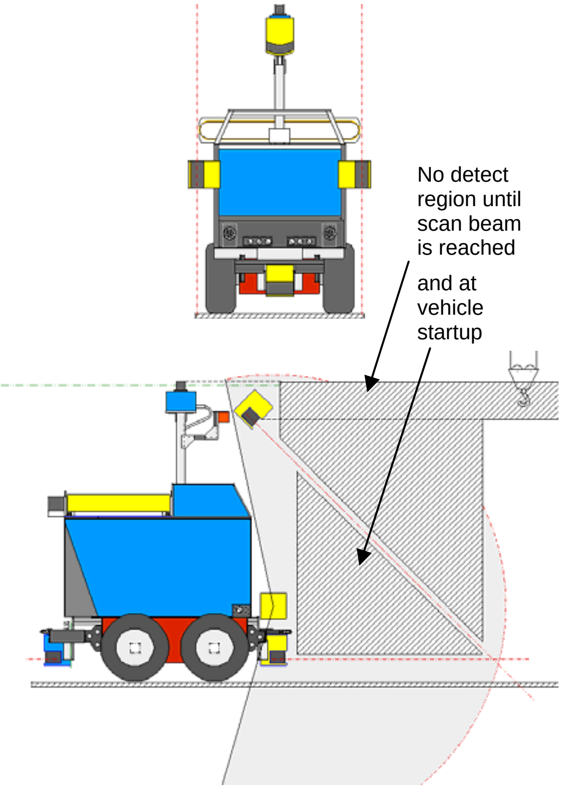


Figure 1 – Front (top) and side (bottom) views of typical 2D line scanning LIDAR sensors mounting configuration on AGV's and the areas they detect. The upper-right portion of the bottom view shows a crane hook that would never be detected.

3D Flash LIDAR technology has led to a relatively new class of range imaging sensors with the potential to scan 3D volumes faster than the 2D scanning systems. Capabilities of this type of sensor could dramatically change the way sensors are used on manufacturing vehicles. This concept will be further explained in Section 2.

To evaluate this class of sensors, a consortium of AGV vendors was formed that took preliminary data with several flash range imaging systems and selected one for further development and investigation. The one selected by the AGV consortium is the sensor used for this work. The data collection system was integrated with a NIST-developed vehicle control system, the Mobility Open Architecture Simulation and Tools (MOAST) framework. [3] This allowed the system to collect data while the vehicle was driving autonomously.

Three experiments were completed providing background data towards illustrating the usefulness of advanced 3D Flash LIDAR cameras on forklifts and PITs. Experiments included: 1) detection of obstacles specified in the ITDSF/ANSI standard, 2) detection of obstacles while highly reflective surfaces are also within the camera's field of view, and 3) detection of forklift tines above the floor. Each of these experiments is briefly explained in Section 3.

2. IDEAL 3D VEHICLE SENSING CONCEPT

Ideally, based on proposed B56.5 standard changes, the volume that completely surrounds the manned or unmanned vehicle should be sensed to ensure a safe manufacturing environment. Ideal 3D vehicle sensing volume concepts are depicted in the graphics shown in Figures 2 and 3. Figure 2 shows top and front views of the ideal 3D sensing volumes for an AGV in green completely surrounding the vehicle, extending beyond the vehicle to include a safe stopping distance. Figure 3 depicts the concept of using multiple 3D imaging sensors to measure the volume surrounding an AGV (Figure 3, top) and a forklift (Figure 3, bottom). Each orange triangle represents a sensor's field of view (FOV). The bottom graphic shows a forklift carrying a double-height load and a movable arm (green) that carries a 3D imager to look over, and in front of, the load. Also, especially for high reach forklifts, a sensor is needed on the mast top to sense overhead obstacles such as ceilings.

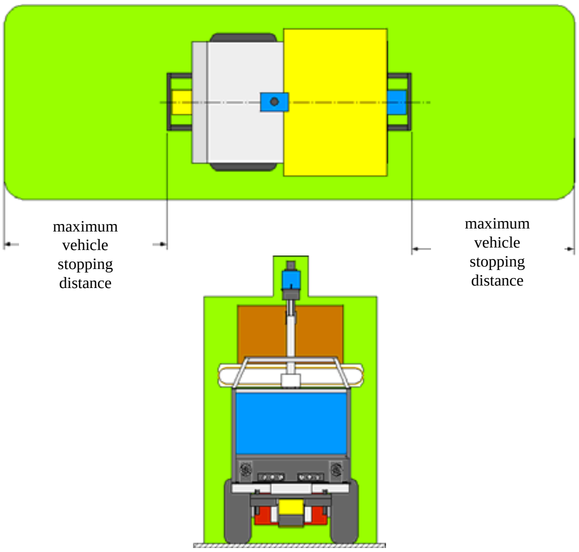

maximum vehicle stopping distance
maximum vehicle stopping distance

Figure 2 – Top view (top) and front view (bottom) of the ideal 3D sensing volume for AGVs.

3. 3D IMAGING EXPERIMENTS

As a preliminary to the implementation of the 3D sensing scenarios posed in the previous section, performance measurements are required of advanced 3D imagers. 3D Flash LIDAR, a time-of-flight range measurement sensor, is still fairly new to the vehicle industry and requires further experimentation in real or simulated manufacturing environments to ensure safe vehicle operations. The sensor used for these experiments measures range to 7.5 m for each of its (176 x 144) pixels with an internal modulating frequency of 20 MHz. The distance of an object is measured by determining the phase-shift between a continuously modulated sine wave that is emitted and the one that is received after having been reflected by the measured scene. The Flash LIDAR sensor that was tested emits a short pulse of light at 870 nm into the environment and senses returned illumination within its 0.26 rad x 0.22 rad (47.5º x 39.6º) field of view (FOV). The following subsections briefly discuss 3D Flash LIDAR experiments performed at NIST for use as safety sensors on manufacturing vehicles.

3.1 Detection of Standard Obstacles

NIST has recently performed measurements with results [4] to be used as background information towards changes to the ITSDF B56.5 Safety Standard with regard to non-contact sensors detecting standard test pieces. The B56.5 standard defines safety requirements relating to the elements of design, operation, and maintenance of powered, not mechanically restrained, unmanned automatic guided industrial vehicles and automated functions of manned industrial vehicles.

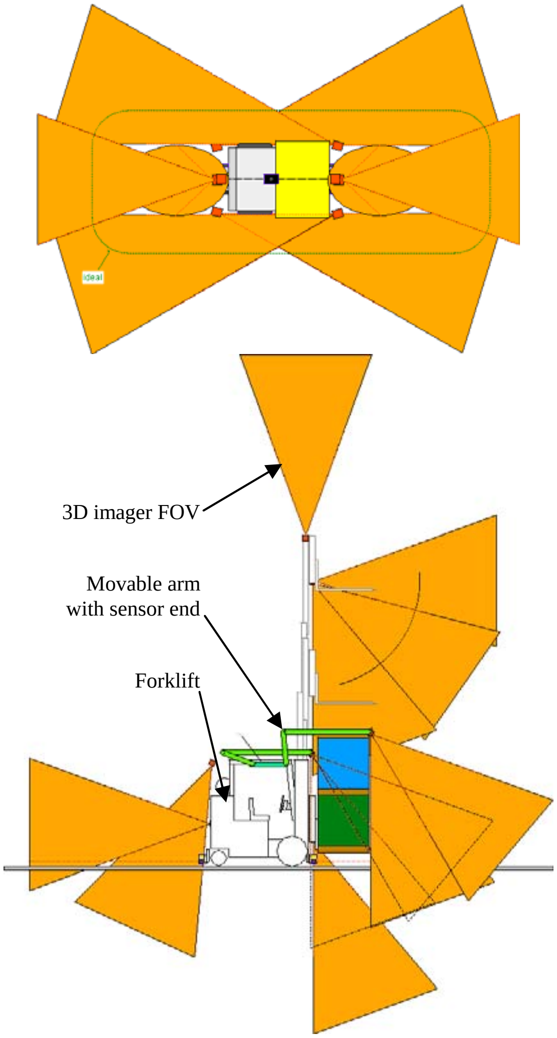
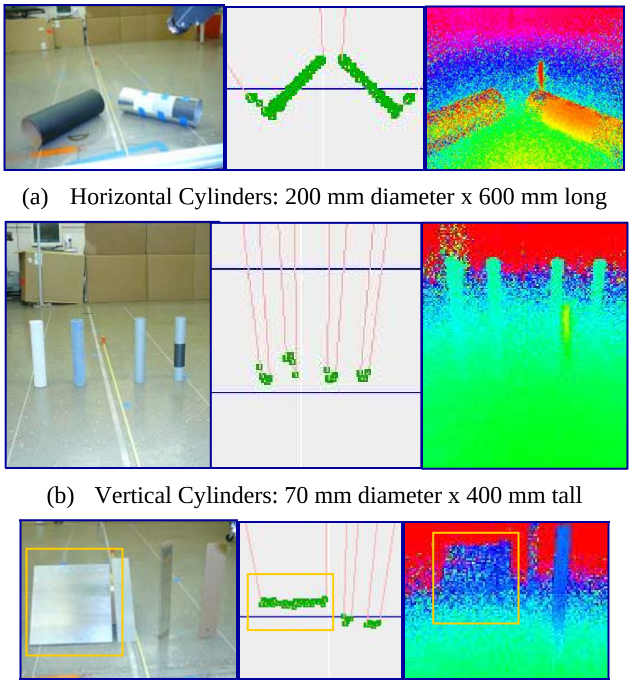


Figure 3 – (top) Top view of an AGV with multiple 3D imaging sensors surrounding the vehicle and measuring to and beyond the ideal sensing region, (bottom) side view of a forklift with multiple 3D imagers showing sensing volumes required to reach the ideal sensing capability. The forklift is shown carrying a double-height palletized load and the green arm carries a 3D imager to look over, and in front of, the load.

Optical and acoustic sensors were tested in these experiments on B56.5 standard test piece sizes, as well as a large flat metal plate, cinder block, and other test pieces and test piece coverings. Over 120 data sets from 21 different tests using a variety of test piece configurations, coverings, layouts, and sensors (sonar, color camera, 2D scanning LADAR, and 3D Flash LIDAR) were collected in a NIST laboratory. For this paper, our focus is mainly on the optical, 3D Flash LIDAR sensor experiments and results.

The ITSDF B56.5-2005 Safety Standard section on non-contact sensing devices states that if the sensor is used as the primary emergency device, the sensor shall be fail-safe in its operation and mounting and shall stop the vehicle travel prior to contact between the vehicle structure and the object detected. Test pieces are to be detected in the main direction of travel and are to be: a 600 mm cylinder with a 200 mm diameter lying at any angle to, and anywhere on, the path of the vehicle and a second, 400 mm cylinder with a 70 mm diameter set vertically anywhere fully within the path of the vehicle. The test pieces described in the standard are of specific size, originally based on the British EN1525 standard. [5] Because the standard is based on contact sensors, however, there are currently no restrictions on test piece coverings. Requirements for covering test pieces are necessary because non-contact sensors may react differently to various materials to be detected. A sensor may or may not detect a particular material and a failure to detect could cause a safety hazard. An example might be that a person wearing dark clothes may not be detected by some optical sensors. Also, only cylindrical test pieces are listed in the standard and perhaps provide better performance than flat test pieces might when positioned at specific angles with respect to the sensors. The experiments were designed to evaluate these additional problems, with the goal of suggesting new language to add to the standard. The experimental setup for each test included positioning the test piece (see Figure 4) at approximately 1 m, 2 m, 3 m, and 4 m distances away from the sensors as data was collected.



(a) Horizontal Cylinders: 200 mm diameter x 600 mm long

(b) Vertical Cylinders: 70 mm diameter x 400 mm tall

(c) Suggested Flat Plate: 500 mm x 500 mm

Figure 4 – Standard and suggested test pieces for the B56.5 standard measured by a (left) color camera, (middle) 2D scanning LADAR, and (right) 3D Flash LIDAR. The suggested flat plate images are marked with a rectangle showing the 500 mm square suggested test piece.

Various coverings over the test pieces, including cotton cloth, paint, known density color patches, and clear glass were used to evaluate how well the sensors could detect the pieces under different conditions. The coverings used were representative of different colored clothing and of manufactured or other industrial materials that may be near vehicles. Figure 4 shows the three test pieces suggested for the B56.5 standard and a snapshot of data collected by the 2D scanning LADAR and the 3D Flash LIDAR.

Figure 4a shows two cylinders, the left one is painted with flat black paint and the right one is partially covered with three reflectance paper patches (6 % (density of 1.22D-black), 50 % (density of 0.30D-gray) and white). Figure 4b shows in the color camera image the right most cylinder covered with the 6% density (black) patch. Figure 4c shows a suggested flat plate test piece to be added to the B56.5 standard.

Overall results from using the 3D Flash LIDAR sensor included the following:

- The sensors used in the tests show a noticeable difference between highly reflective versus relatively low reflective targets.
- In a horizontal cylinder test, two cylinders placed side by side were difficult to detect at 2 m range and undetected beyond 2 m with the flat black painted cardboard cylinder being much more difficult to detect than the metal cylinder. The cylinder appears to blend in with the floor (see Figure 5 range data). The cylinders are detected only when they are in front of a background obstacle or wall.
- In a flat plate detection test, the 1 m, 0º (perpendicular to the sensor) test produced poor results. The obstacles at this distance and angle were difficult to discover by the researcher in the range image, although the obstacles were detected in the intensity image. However, at 1m and tilted at a 45º angle with either the horizontal or vertical axis, the plate was detected in the range and intensity data. Beyond 1m, all flat plates were detected except when the plates were covered with reflective foil. There were no problems detecting painted or unpainted cinder blocks.
- In the flat plate glass test, the glass was never detected because the sensor saw through the glass. The frame holding the glass was detected.

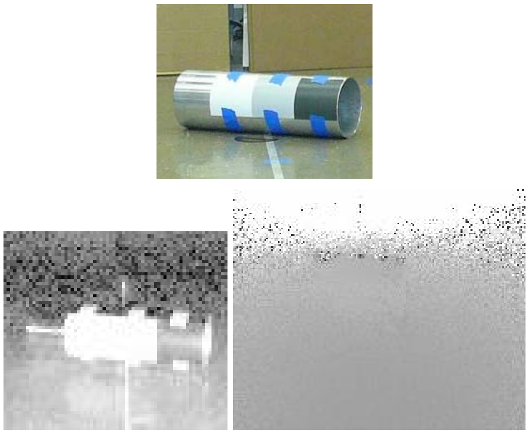


Figure 5 – (top) Close-up of the test piece covered with reflectance patches. (bottom) 3D Flash LIDAR intensity (left) and range (right) data from the horizontal cylinder.

3.2 Effects of Highly Reflective or Emissive Surfaces

Experiments were conducted to better understand how 3D Flash LIDAR reacts to highly reflective objects in their fields of view. Such objects are typical of AGV laser positioning system reflectors. Highly reflective surfaces may cause distortions in the data which could affect how the vehicle sensors perceive their surroundings, potentially causing them to miss obstacles in the vehicle path. Here we briefly explain the experiment and results. Full details can be reviewed in [6]. Two experiments were completed: 1) highly reflective object test and 2) sensor passing by a light source.

The 3D Flash LIDAR was fixed to the front edge of a small table on wheels at a height of approximately 1 m above the floor. We consulted with an AGV manufacturer to establish the typical size and mounting height at which AGV positioning reflectors (cylinders) were typically mounted. We set up a 0.75 m x 0.1 m diameter reflector so that the sensor beam hits the center of the reflector at 2.5 m above the floor, as well as at 2 m (called 0 height), 1.5 m (called -0.5 m height) and 1 m (called -1 m height) above the floor and at ranges of 7 m to 3.5 m from the 3D sensor. The bottom of the reflector cylinder was placed on these surface heights. Figure 6 (top) shows a top view drawing of the experimental layout. Figure 6 (middle) shows the experimental setup showing several obstacles in the sensor’s FOV, including a reflector brightly illuminated by the camera’s flash, and the data capture computer laptop (lower right). Figure 6 (bottom) shows the data captured from the scene using a 3D Flash LIDAR. The yellow arrows show the chairs in the left and right in the photo and range data. Note that the side view (bottom-right) of the data is skewed (i.e., vertical surfaces appear angled back) as a result of the highly reflective surface from the reflector.

Results show that when the 3D Flash LIDAR was mounted low so as to not receive returns from highly reflective surfaces, the received data was not distorted. When the reflector was detected within the scene, the image was distorted in that region. Masking out upper rows of the sensor’s light emitting diodes helped to remove some distortion of the scene for high mounted reflectors. Further, we determined that two options are possible to alleviate the problem: (a) algorithm A, to adjust the threshold of the 3D imaging sensor and/or (b) algorithm B to remove the high intensity measurements – this process is done off-line or post-processed. For algorithm A, the 3D Flash LIDAR can be adjusted to remove high intensity data directly from the received camera data. We added a simple software slide “adjuster” tool for simplicity. This can be run as a constant image adjuster in real time.

Thresholding is performed within the camera so the host computer is not burdened by this extra task. For the second option, we developed an algorithm built into our display tool that finds and counts reflectors in the scene and masks out a variable-size region around the reflector based on the area of high intensity returns. Both the threshold and the masking algorithms can run in real-time. Additionally, algorithm B can be run as image post-processing because the thresholding is carried out in the sensor. Neither algorithm corrects the distorted data. They only detect the region of the image where distortion is likely and data should not be used. Whether this is useful depends on factors outside the sensor processing system. For a particular application it may be acceptable for the AGV to run slowly enough while near a reflector to rely only on a physical bumper or other safety sensor instead of a 3D flash imager. Figure 7 shows color camera (left) and 3D Flash LIDAR intensity and range data (right) of (a) no reflector in the scene, (b) a reflector laying on the chair, (c) a reflector in the chair where the threshold algorithm A has removed it from the data, (d) a reflector in the chair and the masking algorithm B has removed the region of high reflectivity.

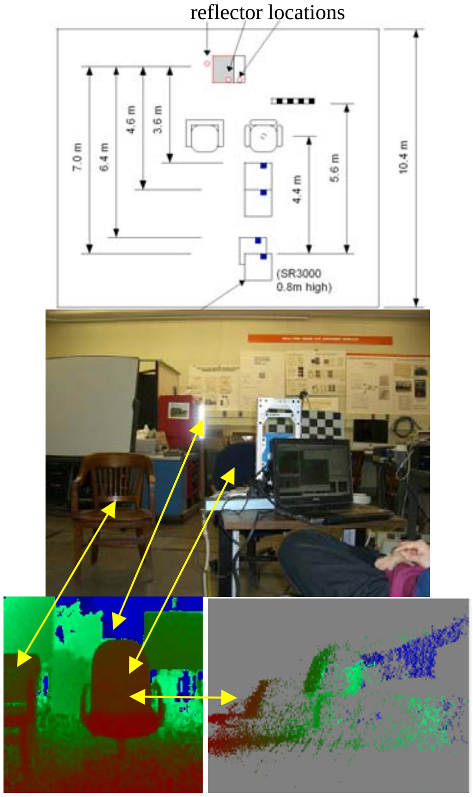
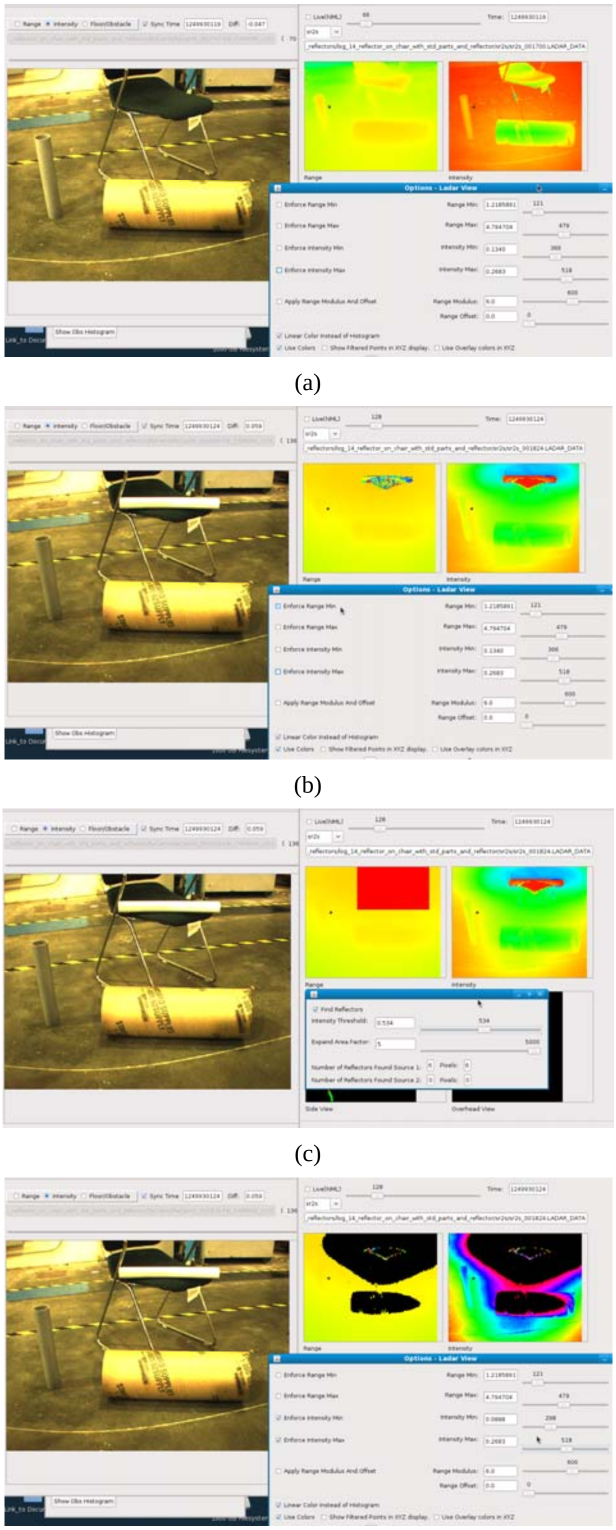


Figure 6: (top) Graphic showing the top view of the experimental setup of the highly reflective surface test; (middle) photo of the experimental set-up showing several obstacles in sensor view, a reflector illuminated by the camera's flash, and the data capture computer laptop (lower right in the photo); (bottom-left) captured data showing the front view and (bottom-right) side view of the scene using a 3D Flash LIDAR sensor. The yellow arrows point to the same chairs and reflector in the photo and data.

An engineering solution to the reflector problem would be to mount the 3D sensor so that it is less likely to see the reflectors as shown in Figure 8. The floor could be flagged so that the sensor does not detect it as an obstacle, given the known sensor height. Unfortunately, this would leave the AGV laser positioning system's sensor above the field of view of the obstacle detection sensor, and thus unprotected by the sensor.



(a)

(b)

(c)

(d)

Figure 7: Color camera (left) and 3D Flash LIDAR intensity and range data (right) of (a) no reflector in the scene, (b) a reflector lying on the chair, (c) a reflector on the chair where the threshold algorithm A has removed the highly reflective region from the data, (d) a reflector in the chair where the masking algorithm B has removed the region of high reflectivity.

The results of the reflectance experiments were:

- Using sensor software drivers programmed to automatically threshold out highly reflective objects could improve bad range data issues.
- Masking the upper sensor LED's removed some image distortion. A better solution is to use a non-reflective surface just above the camera lens to block the upper LED's. And perhaps even better is not to use the data by masking it using software.
- Mounting the 3D sensor specifically to detect obstacles below absolute positioning reflector heights could eliminate or greatly reduce position sensor reflector interference with the 3D sensor.

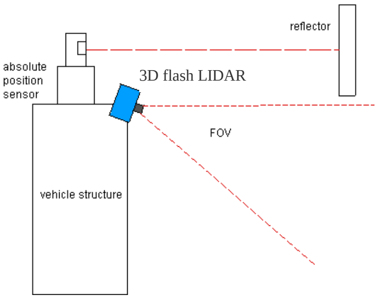


Figure 8: Possible mounting scheme for the 3D Flash LIDAR sensor so as to not detect absolute positioning system reflectors.

For a second part of this experiment, we used a second 3D Flash LIDAR with a similar light source and pointed it directly at the first LIDAR. This experiment provided information about how well the sensor would function when a passing vehicle had similar sensors onboard. The results were that the sensor demonstrated no visually detectable change in range measurements when an LED array light source from a similar sensor passed by. This is probably due to the extreme unlikelihood that one sensor would send out its illumination flash at the same time as the other sensor was in receive mode.

3.3 Detection of Forklift Tines

As suggested by an AGV manufacturer, NIST recently measured forklift tines using the 3D Flash LIDAR. Full details of this experiment, including time results, can be reviewed in [7]. The issue is that forklift tines and other obstacles can overhang the path of automated guided vehicles or other forklifts and go undetected when using only a 2D line scanning LADAR mounted to the vehicle so that the scan line is just above and parallel to the floor. We overlaid the 3D Flash LIDAR data on an image from a color camera to provide a clear view of the tines or other obstacles detected. All measurements were taken dynamically while moving the sensor towards the forklift tines.

The 3D Flash LIDAR sensor and a color camera were mounted together with the camera lens just behind the flash sensor (see Figure 9. The camera FOV is slightly larger than that of the 3D Flash LIDAR. The two sensors were angled so that the 3D Flash LIDAR sensor detected the floor at a maximum distance of 6 m in front of the vehicle. This setting allowed a known sensor-to-floor distance to be used in the data processing algorithm, eliminated detection of the highly-reflective objects above the FOV, and eliminated detection of the cluttered background.

The forklift tines were set at heights of 0.25 m and 0.5 m. No preparation of the tines (e.g., paint, sand, etc.) was done. The cart was pushed towards the forklift tines at approximately 0.09 m/sec during most data captures and for one experiment the cart was pushed at 0.53 m/s. Figure 10 shows the range, intensity, and overlaid obstacle detection on a color image when viewing the forklift tines from the front and side.

After reviewing the results from the unprepared tines and floor experiments, the researchers decided to conduct three additional tests, including: paint the tines with fluorescent paint, cover the floor and combine the painted tines with the covered floor to see what improvements, if any, would result. For each of the additional tests, the above experimental procedure was repeated.

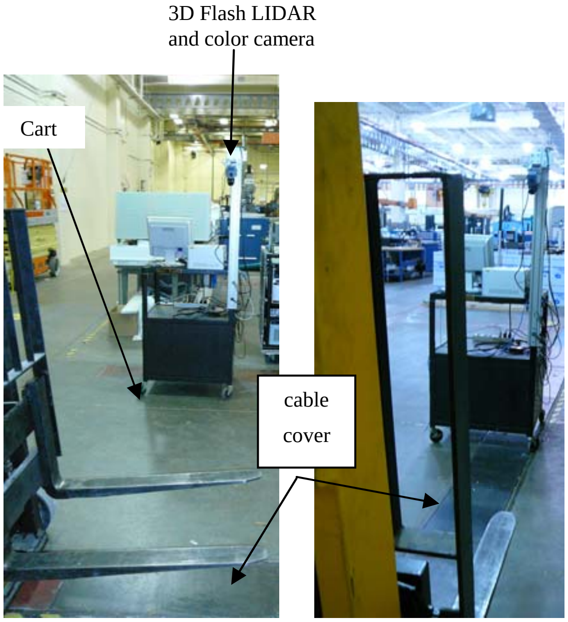


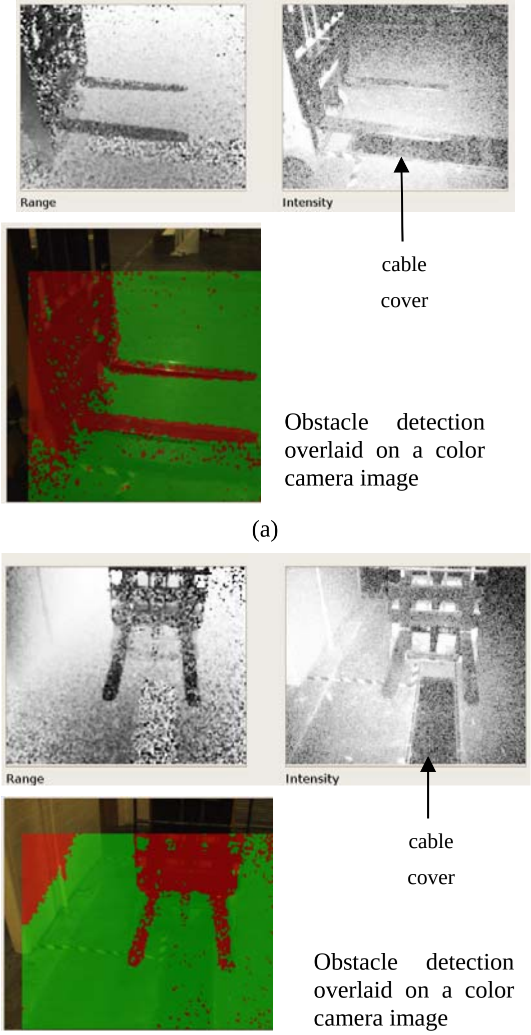
Figure 9: Experimental setup of a cart with 3D Flash LIDAR and color camera sensors (background) and raised forklift tines (foreground). The forklift tines were measured from the side (left) and from the front (right). The metal cable cover appears similar to the fork tines.

Painted Tines: Only the sides of the tines were painted. The reason being: 1) this surface is the smallest, yet still visible to the sensor, and 2) this surface is the least likely to have paint removed when the tines are in use (i.e., wear against pallets is minimal for this surface). Only slight paint overspray covered the top surface of the tines.

Covered Floor: Another experiment included covering the floor with either white poster boards or with gray paint. This created a bright, uniform surface that was less detectable than the unprepared floor.

Combined Painted Tines with Covered Floor: A third experiment included both painted tines and covered floor. Figure 11 shows collected data from the 3D imager overlaid onto a photo of painted tines over painted floor where the floor in the foreground remains unpainted. As shown in the figure, there is little difference in this case between the painted and unpainted floors creating noise displayed as obstacles. However, the forklift and its tines were clearly detected as obstacles.

Each data set (video) was reviewed and a human interpretation that the tines were detected required that a significant number of 3D Flash LIDAR pixels be clustered on the forklift tines. The percentage of time and the distances from the sensor at which the necessary pixel clustering on the tines appeared was noted. Table 1 shows the percentage detection distance moving the sensor from its start position to the tines, at 1 m and at 0.5 m for both front and side views of the tines.


Range
Intensity
cable
cover
Obstacle detection overlaid on a color camera image
Range
Intensity
cable
cover
Obstacle detection overlaid on a color camera image

(a)

(b)

Figure 10: Data from the 3D Flash LIDAR showing range and intensity and obstacle detection overlaid on a color camera image of forklift tines as viewed from the (a) side and (b) front.

A key issue is that the 3D Flash LIDAR sensor processing program uses a height threshold to remove pixels beneath the forklift tines. Without this threshold, data from the floor and the tines may not appear different, and the tines would not be detected even by a human observer. Pixels with heights below 4 cm were removed, leading to two false negative results (Side 1, 4 cm and Front, 4 cm). It may be preferable to develop an adaptive filtering algorithm that would allow the threshold to be lowered.

In some data sets, we saw 'bleeding' of obstacle detect data between, behind and in front of the tines. Figure 12 shows the tines being detected in the intensity and range images and also shows 'bleeding' of data perhaps from the left wall onto the floor in the bottom image. The painted tines joined with the floor covered with white poster boards where another unexpected phenomenon was detected as the front and rear tines were combined as if they were one large obstacle. Since the tines were clearly detected as shown in the intensity and range images, during our evaluation of fork tine detection, we determined that this phenomenon did not change our forklift tines detection results. Therefore, our results show that during this test the 3D Flash LIDAR did detect the front tine.

Table 1: Percentage of Successful Forklift Tines Detection

	Detect Percentage		
tine view (1 =from right, 2 = from left), tine height above the floor	*at the full distance*	*along the last 1 m*	*along the last 0.5 m*
Side 1, 4 cm	0 %	0 %	0 %
Side 2, 4 cm	17 %	57 %	100 %
Front, 4 cm	0 %	0 %	0 %
Side 1, 8 cm	50 %	100 %	100 %
Side 2, 8 cm	50 %	100 %	100 %
Front, 8 cm	27 %	85 %	100 %

A snapshot of data from the additional tests is shown in Figure 11 showing painted and unpainted floors and painted tines.

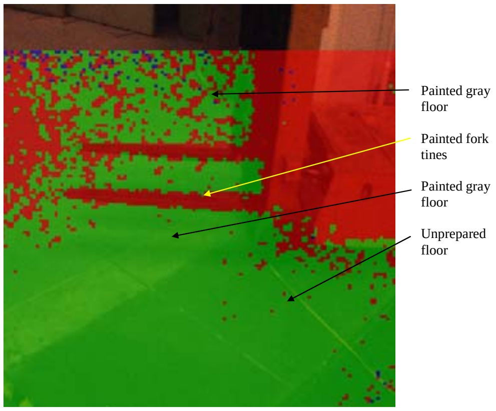


Figure 11: 3D Flash LIDAR sensor data overlaid onto a color camera image of painted forklift tines above a painted floor and beside unpainted floors.

We also showed that a 3D imager can be adjusted so as not to flag obstacles outside of a chosen area. This is useful for when the sensor is attached to a vehicle and the vehicle is driving along a narrow path and/or approaches a turn and the wall or obstacle in front of the vehicle prior to the turn is detected as an obstacle and in turn, stops or slows the vehicle.

Figure 13 shows blue areas on the right and left sides of the vehicle path that have been excluded from processing. Although the forklift tines appear in the image to be beyond the right edge (threshold) between light and dark colors, they are not. Some of the tines are shown as blue (grayscale black) and some are red (grayscale gray). The exclusion regions can be set for any side or range from the sensor and can be varied for complex paths and volumes if needed. Slow or stop regions can be set simultaneously in the same manner.

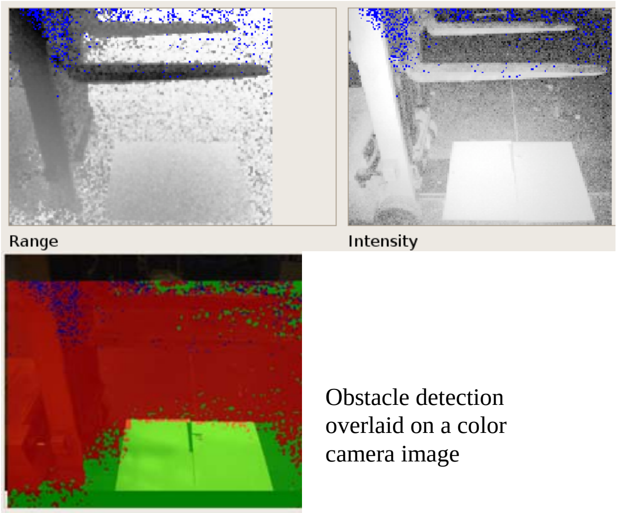


Figure 12 – Range, intensity (top) and obstacle detection overlaid on a color camera image (bottom) of forklift tines. "Bleeding" detect data phenomenon is shown in the bottom image of detected forklift tines above a uniform poster-board floor covering.

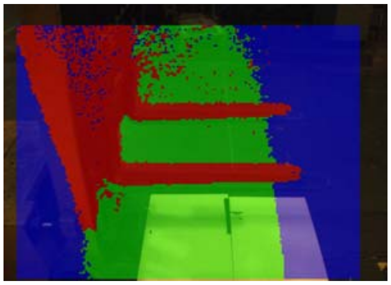


Figure 13 – Data from the 3D imager overlaid onto a color camera image after regions have been excluded from processing (blue areas) using an algorithm that flags when obstacles are outside the vehicle's path.

Results of the tine detection experiments, as detailed in [5], were as follows:

- Within 1 m from the sensor to the tines provides much more robust tine detection than longer ranges.
- The combination of sensors close to the tines and a high threshold height above the floor provides excellent tine detection.
- Higher tines are detected more often than lower tines due to the use of the height threshold.
- When the tines sides were painted with reflective yellow paint and the floor was covered with white poster board, the sensor performed very well.
- Slightly lower performance results were found when the floor was painted with light gray paint.
- Other color floor paints may provide similar results. However, high contrast between the tine and floor paint colors is expected to provide the best results.

4. CONCLUSIONS

In this paper, current and proposed ideal non-contact sensor configurations for manufacturing AGVs and forklift vehicles were presented. These concepts showed the need for performance measurements of advanced 3D imagers. NIST conducted experiments using a 3D Flash LIDAR sensor and a color camera on standard sized test pieces, coated and uncoated with materials and standard colors. The resulting measurements were used as background information to recommend changes to the ITSDF B56.5 standard with regard to non-contact sensors detecting standard test pieces. Before these experiments, only two cylindrical test pieces were considered in this standard. Experimental results determined that a flat test piece should be added. We determined that the 3D sensor used is not a viable safety sensor for vehicles since obstacles near highly reflective surfaces, returned skewed data, 'bleeding data' occurred, and very 'noisy' data was returned when viewing non-uniform floor surfaces. Suggestions for sensor improvements, as we determined from our experiments, may be to support the sensor with robust data processing algorithms that detect highly reflective surfaces, and turn off or block sensor LED's. We found that the light source from another 3D imager had little effect on the 3D sensor data. The percentage of tine detection shown in Table 1 provides a measure of the frequency with which the tines were detected. It shows that higher percentages of detection occur as range decreases. The snapshots and the percentage of detected tines data show that the 3D imager is not robust enough to detect black forklift tines 100 % of the time. However, the detection improves when the tines were painted with fluorescent paint and the floor was painted.

5. REFERENCES

[1] Occupational Safety and Health Administration website, http://www.osha.gov/SLTC/poweredindustrialtrucks/

[2] Industrial Truck Standards Development Foundation, (2005). ITSDF B56.5 Safety Standard for Guided Industrial Vehicles and Automated Functions of Manned Industrial Vehicles, http://www.itsdf.org.

[3] Stephen Balakirsky, Fred Proctor, Chris Scrapper, Tom Kramer. An Integrated Control and Simulation Environment for Mobile Robot Software Development. 8-3-2008. New York. Proceedings of the ASME Computers and Information in Engineering Conference. 8-3-2008.

[4] Roger Bostelman, Will Shackleford, Time of Flight Sensors Experiments Towards Vehicle Safety Standard Advancements, submitted to the Computer Vision and Image Understanding special issue on Time of Flight Sensors, April 2009.

[5] British Standard Safety of Industrial Trucks (1998). Driverless Trucks and their Systems. Technical Report BS EN 1525.

[6] Roger Bostelman, Will Shackleford, Test Report on Highly Reflective Objects Near the SR3000 Sensor, NIST Internal Report to Consortium CRADA Partners, February 2008.

[7] Roger Bostelman, Will Shackleford, Test Report of Performance Measurements of a 3D Imager and Color Camera Viewing Forklift Tines, NIST Internal Report to Consortium CRADA Partners, September 24, 2008.

White Paper

Towards Improved Forklift Safety

Roger Bostelman, Manager
Mobility and Manipulation Project
Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
Department of Commerce
Gaithersburg, MD 20899

October 16, 2009

1 Introduction

There are over 1 million forklifts in operation in the United States with an estimated 2 million operators (6 million including part time operators) [Chugh] and nearly 2 000 automated guided vehicles (AGVs) in use in the US. Forklifts are a necessary piece of material handling equipment for many industries. If used properly, they can reduce employee injuries. Unfortunately, they can also pose some safety risks to drivers, pedestrians, and other equipment and goods. This White Paper summarizes presentations and discussions from the PerMIS 2009 Special Session on "Performance Measurements to Improve Forklift Safety." Papers presented during this special session are listed in the references section.

Attendees of this special session included:

Attendee	Organization
Roger Bostelman,	NIST
Mark Austin	OHSA – Baltimore/Washington Office
Benny Forsman	Danaher Motion/Kollmorgen
Richard Ungerbuehler	SkyTrax, Inc.
Mike Shneier	NIST
Will Shackleford	NIST
David McCartney	US Army Aberdeen Test Center
Luke Fletcher	Massachusetts Institute of Technology
Garrett Place	IFM Efector
Steve Ruth	IFM Efector
Tim Meyers	Toyota Material Handling

This paper is structured to first summarize information from special session papers presenting statistics and issues which define the forklift safety challenge. Following are remedies presented and discussed during the session that can improve forklift safety. Last are discussions and recommendations to further improve forklift safety from the final discussion period of the special session. Some excerpts are copied directly from the papers and presentations from this session.

2 Forklift Safety Statistics and Issues

- OSHA estimates that there are 110000 accidents each year.
- $135000000 immediate costs are incurred due to forklift accidents
- Each year, an additional 94750 injuries related to forklift accidents are reported
- Approximately every 3 days, someone in the US is killed in a forklift related accident
- Approximately 31600 employees suffer some type of injury.
- Losses affect employees through physical and mental suffering.
- Almost 80 % of forklift accidents involve a pedestrian

- 18.8 % of forklift accidents occur when a forklift strikes a pedestrian
- One in six of all workplace fatalities in this country are forklift related
- According to OSHA, approximately 70 % of all accidents reported could have been avoided with proper safety

(some of these statistics are courtesy Bircher America, Inc.)

Forklift operating environments include: pedestrians, blind spots, both indoor and outdoor use, narrow aisles, building columns, 24 hour per day operations, and can include tight turning radii. Pedestrians contribute to accidents since they sometimes don't understand forklift stopping distances and try to "beat" forklifts. Many incidents involve limited driver field of view (FOV) issues where driver controls are mostly designed to drive facing the forks. This forces drivers to see through bars, chains and cables and at times causing their FOV to be completely blocked in the travel direction. Drivers are usually forced to sit facing towards the load, yet look backwards to drive. Researchers report that 75% of side tip-over's occur when a forklift is empty, leading them to conclude that these incidents are due more to speeding than other causes. Losses that affect employers due to forklift accidents include damage to equipment and loss of productivity. Most lost work time reported in 2007 was due to fork truck accidents totaling over 11,040 which is: nearly two times higher than cases involving transportation and material moving, nearly 7 times more than production worker involvement, and over 8 times higher than office or administrative worker incidents.

3 Current Remedies to Improve Forklift Safety

Methods used to reduce forklift accidents include: driver training, safety procedures, equipment maintenance, restricted/designated areas of operation, and facility design. While these strategies will always be elements of workplace safety programs, collision statistics clearly indicate that training, signage, and floor markings for traffic control are not enough to assure a safe environment. Real-time monitoring and control can improve both safety and efficiency.

There are a number of safety systems being researched or in use today. These safety systems are briefly mentioned here and are discussed in more details in [Ungerbeuhler]. Automatic barrier guards can be installed to prevent fork trucks from falling off a vacant receiving dock. These systems prevent forklifts from running off an open dock and can stop a 4500 kg (10000 lb) forklift traveling at up to 0.8 m/s (4 mph). Warning lights can be installed at blind corners to warn of oncoming forklifts. Safety system designers now have new technologies to consider for hazard control, particularly for detecting collision and speeding hazards. For pedestrian detection, a prototype system employs a simple radio frequency (RF)-tag placed in safety vests worn by warehouse workers. An RF receiver was installed on each truck alerting drivers to the presence of any workers within the detection radius of the receiver. The researchers found this wearable RF tag prototype to be a low cost solution that they recommend be used along with other safety measures. One company places a prototype RF transceiver on each vehicle. A similar battery-powered portable transceiver is clipped onto any pedestrian entering the warehouse. The transceiver creates a virtual protection zone around the vehicle or person. When the zones intersect, the transceivers energize a warning signal for both the pedestrian and the vehicle operator. This approach is a viable solution for workers and pedestrians.

Driven largely by the need for smart surveillance and security systems, image processing technology for detecting, identifying, and tracking people in video images is now used in commercial applications. One pedestrian tracking system analyzes the movement of customers in commercial buildings. Processing images from overhead cameras, the system determines the number of customers entering a store and the exact paths taken by customers shopping in the store. In retail and banking applications, the technology is used to track queues of customers and to signal when more check-out lanes need to be opened. While this technology has not yet been applied to collision-avoidance systems, it can be expected in the near future.

Systems based on presence detection sensors indicate that a vehicle is within the detection distance or zone of the sensor. In most cases, there is some ability to configure or engineer the detection distance. Inductive or capacitive proximity sensors and photoelectric sensors, all of which are familiar to automation engineers, fall into this category. An invisible, infrared light beacon mounted on the top of the vehicle is detected by a receiver up to 25 m away and can trigger warning lights or audible alarms for pedestrians and other drivers. Microwave sensors work similarly and can shape the detection zone to match an area of interest. Some companies offer warehouse intersection warning products using microwave sensors. Four sensors and a warning light are hung above an intersection with microwave sensors aimed in all four directions. A vehicle approaching the intersection is detected and triggers the appropriate warning light.

Further complication exists when both automated guided vehicles (AGVs) and forklifts operate in the same space. One company provides accurate and reliable tracking of forklifts, AGVs and other industrial vehicles inside buildings in real time to an accuracy of 5 cm to 20 cm using onboard vehicle vision to view 2D barcodes mounted to the facility ceiling. Important to many safety applications, indoor position systems determine the instantaneous speed and orientation (heading or direction of travel) of each tracked vehicle.

Several sensors, logistical aspects and tasks are needed to bridge between manned and driverless vehicles as shown in Figure 1.

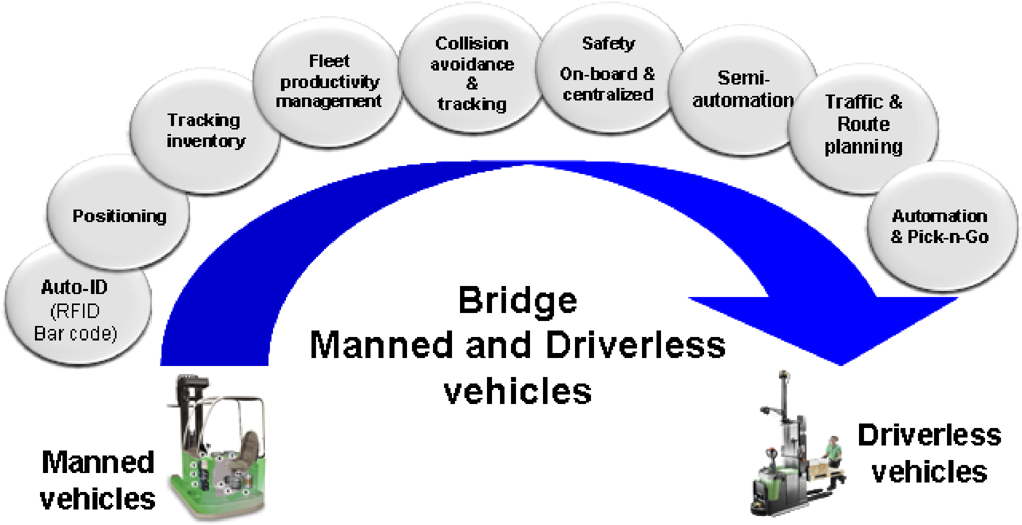


Figure 1 – Drawing showing sensors, logistical aspects and tasks needed to bridge between manned and driverless vehicles

In facilities where autonomous vehicles are used, a different set of safety requirements exists. Autonomous vehicle control systems must assure that inter-vehicular collisions are prevented, and the vehicles must be equipped with safety devices to prevent collisions with people or equipment. The current ANSI/ITSDF B56.5 standard is being improved to include noncontact safety sensors that detect standard-sized objects with specific reflectivity in the path of automated and manned industrial vehicles with automated functions. Two dimensional (2D) laser distance and ranging (LADAR) sensors are currently being used on some forklifts to assist driver field of view and on many AGVs to detect obstacles in the vehicles' paths. 2D LADAR measures range to obstacles along a plane. These sensors work well but are limited by their 2D measurement capabilities. Three dimensional (3D) imaging is needed for viewing overhanging obstacles in the vehicle path. 3D light detection and ranging (LIDAR) sensors are an upcoming sensor technology being studied and proposed for use on both forklifts and AGVs. Stereo vision is now in use on some AGVs to provide 3D viewing.

4 Discussions and Recommendations to Further Improve Forklift Safety

Discussions among the session attendees addressed manned forklifts and AGVs, as well as pedestrians near vehicles, where all three can occupy the same material handling environment. This section provides a summarized transcript of the discussion portion of the session called: "Recommendations Towards Next Generation Forklifts to be Safe" followed by group recommendations. Also listed are two additional recommendations supplied after the group discussion occurred.

Group Discussion

The group discussion was spoken, recorded by a secretary and later summarized without regard to quoting individual participants. It was captured without attribution to encourage expression of opinions. NIST expresses no opinions within the following summarized transcript:

Every facility is dramatically different, but the same types of safety steps can still be taken. There is worry because of cost that the forklift industry will be forced to install scanners on forklifts. There are things that can be done today for using the intelligence of the onboard forklift controls more than how they're currently being used. These things are not being done today because customers are not asking for them. The reason is because customers want their forklift drivers to be able to quickly operate forklifts without costing users additional money or training. Small progressive steps towards a safe forklift solution are suggested rather than a leap forward solution.

The forklift industry is similar to the automotive industry, where the element not completely being controlled is the people around the vehicle. For example, how would a driver know where there is a pedestrian in a distribution facility when their view is blocked unless spotters are used? Also, how would the driver know what are pedestrian intentions in a facility? Some sensors to track people are very expensive. Should everyone wear a sensor like an RFID tag? If so, what happens when that person forgets their tag and then whose fault is it if there's an accident? OSHA says it is the forklift driver's fault. This points to the need for additional safety measures, such as removing pedestrians from the forklift environment or adding safety sensors or better driver FOV sensors to the forklift.

Industry comparison of AGVs versus forklifts, when considering their relative industry sizes, points to AGVs as being safer. An AGV may be too expensive to implement in a factory versus a forklift although there is a need point of affordable innovations. There is a need for the ability to track both pedestrians and vehicles. The challenge with the AGV market is the cost and the safety. 2D LADAR scanners are a great product but very costly to implement to view overhanging obstacles and to completely improve the drivers FOV. The issue is cost versus safety.

Some companies are doing crossover from forklifts to AGVs. Others are converting manned industrial trucks to automated vehicles and light trucks. Long term goals are ideal but where is the balance for cost and safety? With the high cost of forklift accidents per year being $135M, there is a need to find a balance. Toyota's focus is on training to help with overcoming the safety issues associated with automated facilities by training everyone from the administrative person to the forklift drivers.

Vehicle tracking systems are effective for forklift safety, although customers are more interested in the cost versus the safety. So, there is a crossover of taking jobs versus a safe, efficient facility where ultimately safe, efficient systems are more cost effective in the long run. Productivity and efficiency are the driving forces. Companies are not trying to lay off people or get rid of forklift drivers but produce more goods. Freight transport and storage are all cost driven. Companies recognize a safety need, but no one wants to pay for it.

For automated forklifts that follow workers down aisles for manual order picking, several commands are introduced into the system so the order picker can command the robot and the robot will remain safe. However, these commands are more for the order picker than the robot.

Recommendations

The following summarizes the recommendations for improvements to increase forklift safety arising from the discussion and presented papers.

1 Follow the OSHA checklist; enforce the requirement that all drivers wear seatbelts.

2 Ergonomics of vehicles are currently difficult so change the driver's seat so that the driver is not required to turn his/her head backwards to see in the direction of travel when the forklift is carrying a load

3 In noisy environments, add rear backup lighting. Currently drivers rely on their hearing to know when a pedestrian is in the way. Therefore, there is a need for something to replace acoustics. A suggestion would be to use a laser beam that projects 15 m in front of the vehicle through the intersections to tell pedestrians where the forklift is intending to go.

4 Adding sensors and cameras to forklifts to improve the driver's FOV are suggested and being tested at NIST. See Figures 2 and 3.

5 Because there are nearly 1 million forklifts in use today in just the US, there needs to be safety equipment that retrofits to existing forklifts, as well as being designed into new forklifts.

6 There is a need for the ability to track both pedestrians and forklifts and provide the information to the driver and/or to the pedestrians.

7 Systems are needed to control forklift speed to prevent tip over. This must be done this without impacting productivity. Technology is needed that can provide advance warning of hazards (earlier reaction time) and can directly limit forklift speed to assure adequate stopping distance based on location, load, vehicle type, and known hazards.

8 Automatic load weight display is needed for the driver, similar to the speedometer in a vehicle, that would continuously show load weight and changes in % of vehicle lifting capacity as the vehicle moves, lifts, etc. *(post session input from Ted Jurca, Integrated Visual Data Technology, Inc.)*

9 Possible forklift improvements may be *(post session suggestions by Rusty Smith, McCall Handling)*:

- Driver pin-code entry into a keypad or use a card scanner mounted to each forklift to allow that driver to operate the forklift with "black box" (similar to aircraft black boxes) information on who last operated the forklift. Potential uses of this improvement may be to:
    - Recall which operator was running the forklift after an incident occurs,
    - Allow drivers who caused prior incidents to control the forklift at limited speeds and/or carry limited loads.
- Load sensors in the seat to shutdown and ensure a forklift "park" condition when the operator leaves the seat.

Figure 2 shows an experiment performed by NIST using several 3D LIDAR imagers near the edge of a loading dock to detect both positive and negative obstacles. Figure 3 shows a color camera mounted on an extendable boom on a forklift to increase driver field of view of B56.5 standard sized obstacles when blocked by loads, bars, and chains.

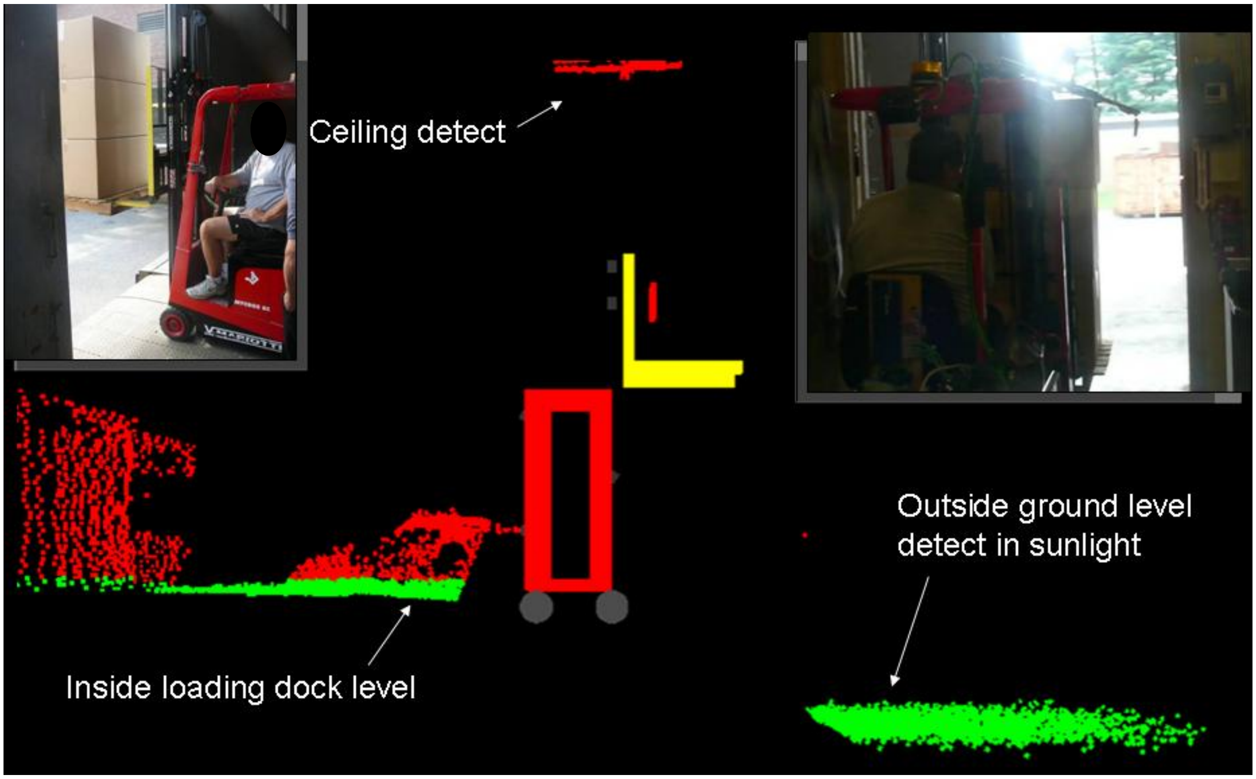


Figure 2 – Data showing detection of both positive and negative obstacles using 3D LIDAR mounted to a forklift while at the edge of a loading dock. The red points are obstacles detected and the green points are detected ground.

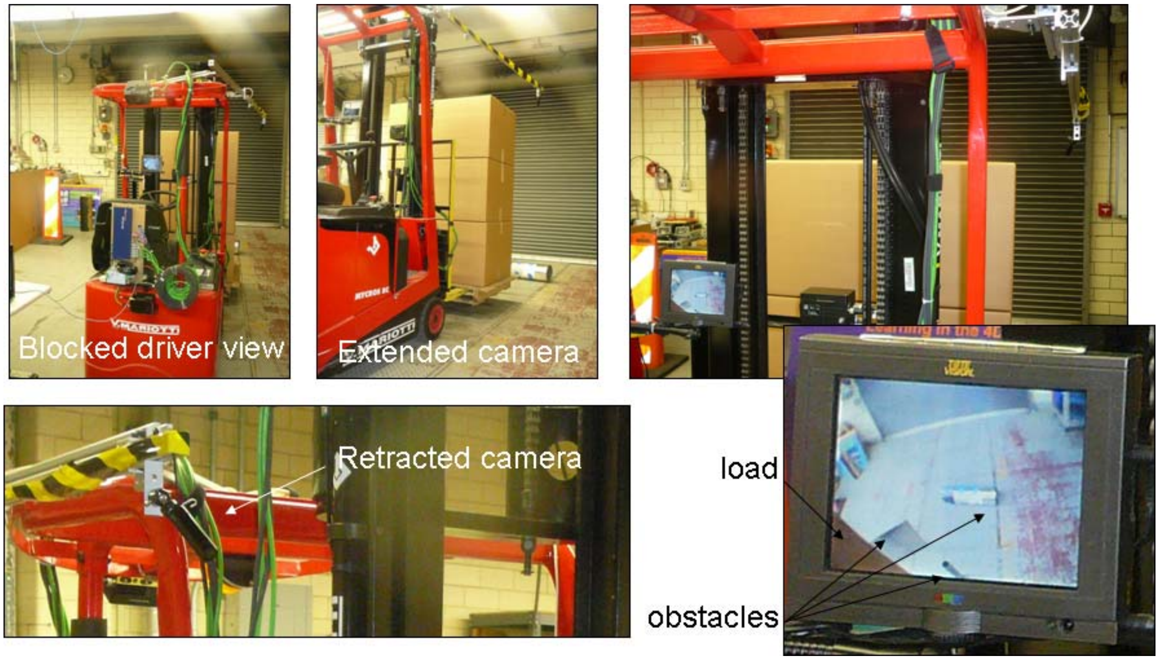


Figure 3 – Color camera mounted on an extendable boom to a forklift to increase driver field of view of B56.5 standard sized obstacles when blocked by loads, bars, and chains. Bottom right shows an onboard monitor displaying camera detected obstacles in front of the forklift load and blocked by the drivers field of view.

5 References

[Chugh], Kevin, "A PC Based Virtual Reality Simulation for Forklift Safety Training," SBIR Phase II, http://www.cdc.gov/od/science/phresearch/pi_abstract2.htm, 2009.

Special session papers presented included the following (in order of their presentation):
[Austin], Mark, "Fork Lift Awareness," Mark Austin, Occupational Safety and Health Administration

[Forsman], Benny "AGV Forklifts - Current and Future Safety Systems,", Danaher Motion, Kollmorgen Corp.

[Ungerbuehler], Richard H. "Where AGV's and Forklifts Roam: Preserving Operational Safety in a Shared Workspace," Sky-Trax Incorporated

[Bostelman], Roger; Shackleford, Will, "Performance Measurements Towards Improved Manufacturing Vehicle Safety," NIST Intelligent Systems Division