U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

NUREG/CR-6895, Vol. 3
ORNL/TM-2007/188, Vol. 3

Technical Review of On-Line Monitoring Techniques for Performance Assessment

Volume 3: Limiting Case Studies

Office of Nuclear Regulatory Research

U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment

NUREG/CR-6895, Vol. 3
ORNL/TM-2007/188, Vol. 3

Technical Review of On-Line Monitoring Techniques for Performance Assessment

Volume 3: Limiting Case Studies

Manuscript Completed: October 2007
Date Published: August 2008

Prepared by
J.W. Hines[1], J. Garvey[1], D.R. Garvey[1], and R. Seibert[1]

[1]The University of Tennessee-Knoxville
Department of Nuclear Engineering
Knoxville, TN 37966-2210

Oak Ridge National Laboratory
Managed by UT-Battelle LLC
Oak Ridge, TN 37831-6156

S.A. Arndt, NRC Project Manager

NRC Job Code N6080

Office of Nuclear Regulatory Research

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at http://www.nrc.gov/reading-rm.html. Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   Mail Stop SSOP
   Washington, DC 20402–0001
   Internet: bookstore.gpo.gov
   Telephone: 202-512-1800
   Fax: 202-512-2250
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:
Address: Office of the Chief Information Officer, Reproduction and Distribution
Services Section
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
E-mail: DISTRIBUTION@nrc.gov
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address http://www.nrc.gov/reading-rm/doc-collections/nuregs are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—
The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—
American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org

Legally binding regulatory requirements are stated only in laws; NRC regulations; licenses, including technical specifications; or orders, not in NUREG-series publications. The views expressed in contractor-prepared publications in this series are not necessarily those of the NRC.

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG–XXXX) or agency contractors (NUREG/CR–XXXX), (2) proceedings of conferences (NUREG/CP–XXXX), (3) reports resulting from international agreements (NUREG/IA–XXXX), (4) brochures (NUREG/BR–XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG–0750).

212–642–4900

Page intentionally blank

ABSTRACT

Traditionally, the calibration of safety-critical nuclear instrumentation has been performed at each refueling cycle. However, many nuclear plants have expressed a desire to move toward condition-directed rather than time-directed calibration. This condition-directed calibration is accomplished through the use of on-line monitoring (OLM).

With a sound OLM system in place, nuclear plants may be able to extend the required calibration interval. Only recently have nuclear plants come to the point where they are ready to apply for license amendments to extend their calibration frequency. To help support the regulatory review of these amendments, researchers from The University of Tennessee's Nuclear Engineering department were contracted to draft a NUREG/CR series. The goal of this entire NUREG/CR series is to provide guidance to the regulatory review process of OLM. The volumes present the technical background of OLM and explain most of the theory behind the OLM concept. Additionally, they discuss and analyze specific issues regarding the application of OLM in a nuclear power plant. The first volume in this series, NUREG/CR-6895, entitled *Technical Review of On-line Monitoring Techniques for Performance Assessment, Volume 1: State-of-the-Art*, was completed in March 2005 and published in January 2006. Volume 1 offers a general overview of current sensor calibration monitoring technologies and their uncertainty analysis, a review of the supporting information necessary for assessing these techniques, and a cross-reference between the literature and the requirements listed in the SER. *Volume 2: Theoretical Issues* presents an in-depth theoretical study and independent review of the most widely used OLM techniques It includes a presentation of the theory and further explanation of the assumptions inherent in the empirical models and the uncertainty quantification techniques.

This third and final volume summarizes seven case studies investigating the effects of model development and assumptions on model performance. Two case studies concern the effect of not meeting model assumptions: evaluating query data outside the training region and training with faulty data. Recommendations are given for identifying and correcting the problems caused by not meeting these important data assumptions. The third and fourth case studies investigate the effects of high noise levels on model performance and compare different methods of data denoising, respectively. The remaining three case studies examine different features of model development by comparing vector selection methods, different numbers of memory vectors, and robust distance measures. Methodologies to determine the appropriate model development parameters for each of these cases are outlined. Finally, a section is included that highlights the special considerations needed for redundant-sensor model architectures. Although this study is not an exhaustive review of the many issues in OLM system development, it provides a base set of considerations that must be accounted for and a method for testing these considerations with other model architectures

Paperwork Reduction Act Statement

This NUREG does not contain information collection requirements and, therefore, is not subject to the requirements of the Paperwork Reduction Act of 1995 (44 U.S.C. 3501 et seq.).

Public Protection Notification

The NRC may not conduct or sponsor and a person is not required to respond to a request for information or an information collection requirements unless the requesting document displays a currently valid OMB control number

Page intentionally blank

FOREWORD

For the past two decades, the nuclear industry has attempted to move toward condition-based maintenance philosophies, using new technologies developed to ascertain the condition of plant equipment during operation. Consequently, in November 1995, the U.S. Nuclear Regulatory Commission (NRC) published NUREG/CR-6343, *On-Line Testing of Calibration of Process Instrumentation Channels in Nuclear Power Plants*, which summarized the state-of-the-art in the area of OLM. In that report, the NRC staff concluded that it is possible to monitor the calibration drift of field sensors and associated signal electronics, and determine performance of instrument channels in a nonintrusive way. Then, in 1998, the Electric Power Research Institute (EPRI) submitted Topical Report (TR) 104965, *On-Line Monitoring of Instrument Channel Performance*, for NRC review and approval. That report demonstrated a nonintrusive method for monitoring the performance of instrument channels and extending calibration intervals required by technical specifications (TS). The NRC subsequently issued a related safety evaluation report (SER), dated July 24, 2000, in which the staff concluded that the generic concept of OLM is acceptable for use in tracking instrument performance as discussed in EPRI TR-104965. However, the staff also listed 14 requirements that must be addressed in plant-specific license amendments if the NRC is to relax the TS-required calibration frequency for safety-related instrumentation. The SER neither reviewed nor endorsed either of the two methods addressed in the topical report.

This contractor-prepared NUREG-series report is the third volume of a three-volume set, which will provide an overview of current technologies being applied in the United States to monitor sensor calibration. Volume 1, published in January 2006, provided a general overview of current sensor calibration monitoring technologies and their uncertainty analysis, a review of the supporting information needed to assess those techniques, and a cross-reference between the literature and the requirements listed in the staff's SER. To augment that overview, the second volume provides an in-depth theoretical study and independent review of the most widely used online sensor calibration monitoring techniques, including the underlying theory and an evaluation of the inherent assumptions. The techniques were selected because they were considered by the EPRI OLM working group, were applied in the EPRI OLM Implementation Project, or are currently available as commercial products. This third volume by contrast, Volume 3, provides case studies that apply modeling and uncertainty analysis techniques to a wide variety of plant data sets to consider the effects of modeling assumptions and limitations.

The NRC staff anticipates that readers will use this reference to quickly locate the technical information required to assess the methods presented in plant-specific applications. This report is intended to provide the technical details that are necessary to conduct an accurate evaluation of each technique. This report should not be construed to imply that the NRC endorses any of the methods or technologies described herein; that is, a licensee would need to provide a complete description and justification for any proposed OLM approach.

Jennifer Uhle, Director
Division of Engineering
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission

Page intentionally blank

CONTENTS

Page intentionally blank

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

For the past two decades, the nuclear industry has attempted to move toward a condition-based maintenance philosophy using new technologies developed to monitor the condition of plant equipment during operation. Specifically, techniques have been developed to monitor the condition of sensors and their associated instrument loops while the plant is operating. Historically, process instrumentation channels have been manually calibrated at each refueling outage. This strategy is not optimal because sensor conditions are only checked periodically; therefore, faulty sensors can continue to operate for periods up to the calibration frequency. In addition, periodic maintenance strategies result in the unnecessary calibration of instruments that are operating correctly, which can lead to premature equipment aging, damaged equipment, unnecessary plant downtime, and improper calibration under non-service conditions. In fact, recent studies have shown that less than 5% of process instrumentation being manually calibrated requires any correction at all. Therefore, plant operators are interested in finding ways to monitor sensor performance during operation and manually calibrate only the sensors that require correction.

In 1995, the Nuclear Regulatory Commission (NRC) published a report on the state-of-the-art in the area of on-line monitoring. This report, which was prepared by the Analysis and Measurement Services Corporation as NUREG/CR-6343, was entitled *On-Line Testing of Calibration of Process Instrumentation Channels in Nuclear Power Plants*. This report concluded that it is possible to monitor calibration drift of field sensors and associated signal electronics to determine the performance of the instrument channels in a nonintrusive way.

In 1998, the Electric Power Research Institute (EPRI) submitted Topical Report (TR)-104965, *On-Line Monitoring of Instrument Channel Performance* for NRC review and approval. This report demonstrated a nonintrusive method for monitoring the performance of instrument channels and extending calibration intervals required by technical specifications (TS). The calibration extension method requires an underlying algorithm to estimate the process parameter. In the topical report, two such algorithms were described. The NRC issued a safety evaluation report (SER) on TR-104965, dated July 24, 2000, which concluded that the generic concept of on-line monitoring (OLM) for tracking instrument performance as discussed in the topical report is acceptable. However, it also listed 14 requirements that must be addressed by plant specific license amendments if the TS-required calibration frequency of safety-related instrumentation is to be relaxed.

When evaluating the proposed methods during development of the SER, the NRC staff did not review the two algorithms presented in the topical report and did not limit the application to those two methods. However, during a license review, it is necessary that the technical details of each particular technique be understood to determine whether the technique meets the specified functional requirements. Thus, this series of reports is meant to provide a basis for these necessary technical details so that an accurate evaluation of each technique and its application can be made. This report should not be construed as the NRC's endorsing any of the described methods or technologies. A licensee would need to have a complete description and justification for any approach proposed for OLM.

This three-part NUREG/CR series provides an overview of the current OLM technologies. Volume 1 provides a general overview of the technologies currently being implemented for sensor calibration monitoring and presents the techniques used to quantify the uncertainty inherent in the empirical process variable predictions. It also provides a survey of the relevant information and a cross-reference between the relevant information and the 14 requirements. It is expected that readers will use this reference to quickly locate the technical information required to assess the methods presented in plant-specific license amendments. Volume 2 of the report presents an in-depth theoretical study and independent review of the most widely used on-line sensor calibration monitoring (OLM) techniques. It includes a presentation of the theory and a listing and evaluation of the assumptions inherent in the

methods. This third and final volume further explores some of the modeling assumptions identified in Volume 2. This volume reports the results of a study that applied an auto-associative kernel regression (AAKR) model architecture to actual nuclear plant data and determined the effect on an OLM system when some of the inherent modeling assumptions are no longer met. The assumptions examined include instances when OLM data that was either being used to develop the model, test, or apply the model to determine a sensor's status is in some way faulted. The data issues include data that contain outliers, or some form of "stuck" or erroneous data, and also data that is no longer represented by the training data. The effect of noisy sensors and the impact that the denoising routine has on OLM uncertainty are investigated. The study also evaluates factors from the model that could influence OLM predictions, such as implementing different memory vector selection methods, including too few or too many memory vectors, and using different distance calculation methods. Finally, the study considers what, if any, additional factors must be taken into account when employing redundant sensor model architecture.

ABBREVIATIONS

AAKR	auto-associative kernel regression
AAMSET	auto-associative multivariate state estimation technique
AANN	auto-associative neural network
EPRI	Electric Power Research Institute
EULM	error uncertainty limit monitoring
GA	genetic algorithm
GCC	generalized consistency check
ICMP	Instrumentation and Calibration Monitoring Program
ICV	instrument calibration verification
IIR	infinite impulse response
MIMO	multiple-input, multiple-output
MISO	multiple-input, single-output
MSE	mean squared error
NRC	Nuclear Regulatory Commission
NUREG/CR	NUREG Prepared by a Contractor
OLM	on-line monitoring
pdf	probability distribution function
PSA	parity space algorithm
PEANO	process evaluation and analysis by neural operators
RPS	reactor protection system
SER	Safety Evaluation Report
SISO	single-input, single-output
SPRT	sequential probability ration test
TR	topical report
TS	technical specifications

Page intentionally blank

1. INTRODUCTION

1.1 Background

Traditional approaches to instrument calibration at nuclear power plants are expensive in terms of both labor and money. These calibrations require that the instrument be taken out of service and falsely loaded to simulate actual in-service stimuli. This can lead to damaged equipment and incorrect calibrations due to the adjustments being made under nonservice conditions. While proper adjustment is vital to maintaining proper plant operation, a less invasive technique is desirable. For this reason, there has been a major push in the nuclear industry to move toward condition-directed rather than time-directed calibration. For the past 20 years, several nuclear utilities, along with the Electric Power Research Institute (EPRI), have investigated methods to monitor the calibration of safety-critical process instruments. The techniques being investigated are given the blanket term of "on-line monitoring" (OLM) because they monitor the calibration status of sensors while the plant (and sensors) are operating. The purpose of OLM for sensor calibration is to identify drifting or faulted channels. In empirical model-based OLM, data collected from plant sensors are sent to software, which uses an empirical model to obtain an independent estimate of the fault-free process parameter value. This estimate is compared to the measured value to monitor instrument drift or other faults. With this technology, continuous or near-continuous sensor surveillance is possible. Thus, it is possible for manual sensor calibration to be performed based on the sensor's performance, rather than the sensors simply being recalibrated on a frequency-based schedule.

In 2000, the U.S. Nuclear Regulatory Commission (NRC) issued a safety evaluation report (SER) (NRC 2000) on an EPRI-submitted Topical Report (TR) 104965, "On-Line Monitoring of Instrument Channel Performance" (EPRI 2000). This SER concluded that the generic concept of OLM for tracking instrument performance is acceptable as discussed in the topical report. However, the SER also listed 14 requirements that must be addressed by plant specific license amendments if the Technical Specification (TS)-required calibration frequency of safety-related instrumentation is to be relaxed. Current OLM techniques propose relaxing the frequency of instrument calibrations required by the U.S. nuclear power plant TS to allow sensors to be calibrated based on their calibration condition, rather than every safety critical sensor being calibrated at each outage. With OLM, it is proposed that at least one redundant sensor will still be calibrated at each scheduled fuel outage. For n redundant sensors, all sensors will be calibrated at least once every n outages. Regardless of the number of redundant sensors in a group, a sensor cannot go more than 8 years without being recalibrated. If an instrument is found to exceed the drift limits, it must be calibrated, regardless of when it is scheduled for calibration, and its calibration cannot replace the scheduled calibration.

To implement this new OLM calibration strategy, plants must file for a license amendment. In this amendment, plants must detail how OLM is accomplished and provide a technical basis for the technique they are planning to employ. The empirical model, which generally contains data from all or many of the sensors in a process, is one of the most critical elements of an OLM system, but its limitations are also generally the least understood.

The two previous volumes in this series have discussed the general theory behind some of the commonly employed OLM models. The quantification of model uncertainty, one of the most difficult requirements of an OLM system, was also discussed in these earlier reports. Volumes 1 and 2 addressed some of the modeling and data assumptions (such as coverage of the training vectors, effect of model architecture, etc.). This third volume explores those issues through case studies.

In Volume 3, an OLM model is applied to special cases where the basic modeling assumptions may no longer be met. The model performance under these limiting cases is then evaluated. This report attempts to gauge how the model handles various real-life scenarios when the underlying assumptions

cannot be validated. This volume offers recommendations as to how such situations should be dealt with as well as what safeguards or additional measures need to be in place to help identify or avoid these situations. This volume also recounts some of the "lessons learned" from the experiences of developing and implementing OLM models. These lessons attempt to solve or minimize the effects of common problems encountered when implementing or training a new OLM model. Descriptions of these "lessons learned" are embedded in many of the limiting case studies.

The following chapters describe the limiting cases and also evaluate the performance of the OLM model developed with nuclear plant data and subjected to atypical conditions or situations. An auto-associative kernel regression (AAKR) model was used in this study. This model was chosen because it is readily available and has an architecture similar to many commercially available OLM models, which are proprietary and could not be used in this study. A description of this model as well as two of the other currently used commercial OLM models, namely auto-associative multivariate state estimation technique (AAMSET) used by Smart Signal Inc. and auto-associative neural network (AANN) used in the Process Evaluation and Analysis by Neural Operators (PEANO) developed by the Halden Reactor Project, can be found in Volume 2 of this series, along with an explanation of basic OLM modeling theory. These models are by no means the only models applicable for OLM. However, they are considered the current state of the art in OLM technology.

All of the aforementioned models are considered nonredundant, meaning that they are able to monitor sensors measuring correlated but not necessarily identical process parameters. In contrast, redundant sensors measure the same process variable at nearly the same location in the process. Models classified as redundant, such as EPRI's Instrument and Calibration Monitoring Program (ICMP), require that all inputs come from truly redundant sensors. For instance, the simplest redundant model is just a direct average of the redundant sensor values. Even models that are classified as nonredundant can, and in fact often do, have only a group of redundant sensors as their inputs. However, these models are still called nonredundant simply because they are not limited to having only redundant sensor inputs. Because nuclear plants require that all measurements of safety critical processes come from at least two redundant sensors and because, in practice, most redundant sensor sets contain at least three or four sensors, redundant-sensor models are particularly applicable to OLM. However, this type of model comes with its own advantages and disadvantages. Although this report primarily focuses on the non-redundant AAKR model, it also briefly describes some of the issues that arise from using redundant sensor models.

1.2 Organization of the Document

The same AAKR modeling technique and data sets were used to conduct the studies reported in this volume, with the exception of the study that examines redundant-sensor modeling. For this reason, the following sections in this chapter describe AAKR modeling, the metrics used to gauge its performance, and the data sets used for the limiting cases. Before discussing the limiting cases, Chapter 2 provides an overview of the methods used for model development and analysis. Chapter 3 then presents the "baseline" results of the AAKR model for the datasets. These results are those from "ideal" AAKR models that have been properly developed and applied to the datasets without any of the common complications or problems that can occur during OLM (most of which are described in the limiting case studies). The baseline results are presented so that accurate comparisons can be made between them and the limiting case results, so that the effects of each limiting case can be analyzed and better understood.

Each of the limiting cases has a dedicated chapter. In these chapters, the corresponding limiting case is described in detail, the results of applying this case to the models are presented and discussed, and recommendations for preventing the case and correcting or mitigating its effects are provided. Chapter 4 presents the results of models trained with data containing sensor faults or outliers. Chapter 5 focuses on models attempting to monitor sensors operating outside of the training region. Chapter 6 considers the impact that noise in the data has on the OLM process. Chapter 7 explores the impact of the denoising

method on OLM uncertainty. Chapter 8 investigates the different vector selection methods that can be used in OLM modeling. Chapter 9 analyzes models developed using too few or too many training vectors. Chapter 10 examines the effect of a model's distance calculation method. Chapter 11 discusses some of the additional considerations that must be taken when using redundant models. Chapter 12 provides a summary of the research results and reiterates the conclusions that can be drawn from the studies. Finally, the appendixes provide supplemental tables and figures showing additional results from the limiting case studies that were too expansive to remain in the body of the document.

1.3 Model Description

Fig. 1-1 is a simple block diagram of a basic instrument calibration monitoring system. In this figure a vector of sensor measurements (**x**), which may be corrupted by noise, drift, or fault, is input to a prediction model, which calculates corrected, error-free estimates of the sensors (**x**'). The estimates are compared to the measured values forming differences called residuals (**r**). A decision logic module determines if the residuals are statistically different from zero and establishes the health or status (**s**) of each sensor. This module may also use predictive uncertainty values and drift limits to determine the condition of the instrument channel.

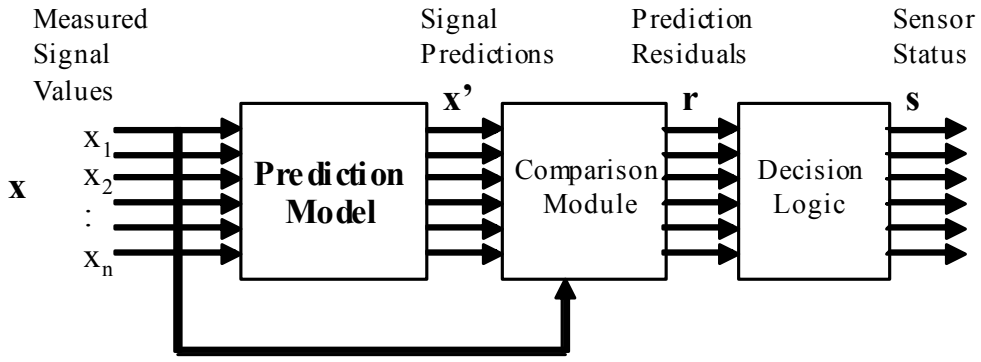


Fig. 1-1. Instrument calibration monitoring system diagram.

The prediction models are empirical models, which, unlike first-principle or physical models, are based only on data and are used to predict, not explain, process variable values in a system. A model may be either defined by a set of parameters and functional relationships (parametric) or a set of data and algorithmic estimation procedures (nonparametric). For instance, the following polynomial model is an example of a parametric model:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2 + b_4 x_1^2 + b_5 x_2^2 \ ,$$

where y is the variable to be estimated, x_1 and x_2 are the predictor variables, and bi are the coefficients.

To completely define this model for a given set of training observations, the polynomial coefficients, bi, are optimized to minimize some objective function, usually the sum of the squared error (SSE). Once the optimal polynomial coefficients have been estimated, the model is completely specified by the above

equation and the estimated coefficients. Therefore, a parametric model may be roughly defined as a model that can be completely specified by a set of parameters and a functional relationship for applying the parameters to new data to estimate the response.

A nonparametric model, by contrast, stores historical data exemplars in memory and processes them each time a new query is made. For instance, rather than modeling a whole input space with a parametric model such as a neural network or linear regression, local nonparametric techniques may be used to construct a local model in the immediate region of the query. These models are constructed "on the fly," not beforehand. When a new query is made, the algorithm locates historical exemplars in its vicinity and performs a weighted regression with the nearby observations. Kernel regression (KR) is one such type of nonparametric model.

This section discusses the various kernel regression algorithms in detail, beginning with a top-level description of the steps used in KR. Following this discussion, methods for similarity quantification, specifically distance measures and kernel functions, are discussed. This discussion will serve as a foundation for the comprehensive descriptions of the different model architectures that are presented in the final subsections of this discussion.

1.3.1 Method overview

In statistics and empirical modeling, the process of estimating a parameter's value by calculating a weighted average of historical, exemplar observations is known as KR (Atkeson et al., 1997a). Generally, KR may be most compactly represented by the so-called Nardaraya (1964)-Watson (1964) estimator. For a simple single-input, single-output (SISO) regression model, where the input x is used to estimate the output y, the Nardaraya-Watson estimator is as follows:

$$\hat{y}(x) = \frac{\sum_{i=1}^{n} \left[K(X_i - x) Y_i \right]}{\sum_{i=1}^{n} K(X_i - x)} , \tag{1.1}$$

where

n is the number of exemplar observations in the KR model;

X_i and Y_i are the input and output for the ith exemplar observation;

x is a query input;

$K(X_i - x)$ is a weighting or kernel function, which generates a weight (similarity) for a given difference of a query from an exemplar vector; and

$\hat{y}(x)$ is an estimate of y, given x.

To understand KR, the mechanics of Eq. (1.1) must be understood. To do this, the steps used in KR are enumerated below and then related back to Eq. (1.1).

For a query observation of the model inputs, the KR estimation process can be structured into three steps. First, the distance of the query from each of the input exemplars is calculated. Next, the distances are supplied as inputs to a kernel function, which converts the distances to weights (similarities). Finally, the weights are used to predict the model output as a weighted average of the output exemplars. These steps are depicted in Fig. 1-2.

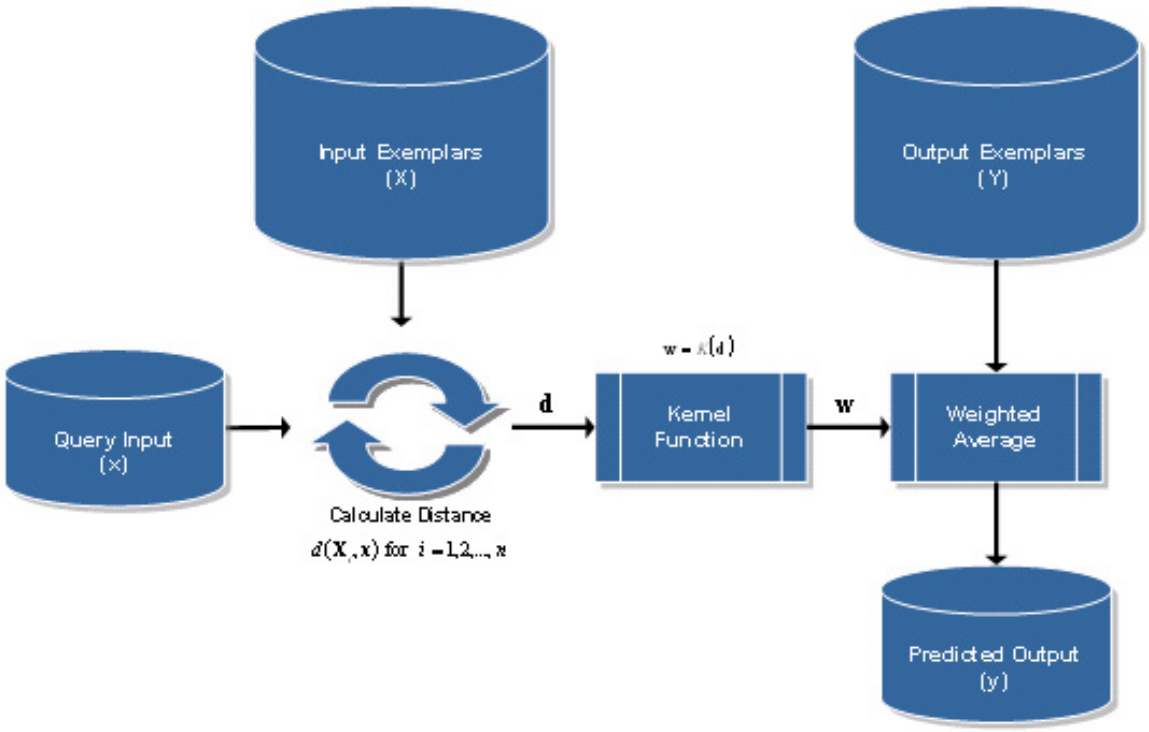


Fig. 1-2. Process diagram for the KR prediction algorithm.

For the sake of clarity, the process presented in Fig. 1-2 is further discussed in Table 1-1. To begin, exemplar inputs ***X*** and outputs ***Y*** are considered. These observations represent the "memory" of the KR model and are therefore often referred to as memory observations or memory vectors.

The KR prediction process can also be thought of as answering the question: "Based on observed inputs ***X*** and outputs ***Y***, what will be the system output for a new query input ***x***?" Before this question can be answered, two additional questions must be considered:

1. How similar is the query to the known inputs ***X*** (Step 1 and Step 2)?
2. From the similarities of the query to the known inputs ***X***, the most likely model outputs from the exemplar set can be identified. How can these likeliest outputs be aggregated to estimate the model output (Step 3)?

Table 1-1. The three-step KR prediction process

Step 1—Distance Calculation The query input, **x**, is an observation of the inputs to the KR model and will be used to predict the output. To do this, the distance between the query and each exemplar input must be determined. This is accomplished by evaluating a distance measure with each exemplar input and the query input as arguments. The distance calculation for the *i*th exemplar observation is represented by $\boldsymbol{d}(X_i,x)$. This calculation is repeated for each of the exemplar inputs. This means that the result of this entire operation is a vector of *n* distances **d**, representing the distance of the query to all of the *n* input exemplars.
Step 2—Similarity Quantification At this point, the distances of the query to each of the input exemplars have been evaluated and must be converted to weights or similarities. This is accomplished by evaluating a similarity or kernel function for each of the distances. For the vector of *n* distances **d**, the kernel function results in a vector of *n* weights $\boldsymbol{w}$, which represents the similarity of the query to each of the input exemplars.
Step 3—Output Estimation In the final step of the prediction process, the similarities of the query to each of the input exemplars are combined with the output exemplars to obtain estimates of the output. For KR this is accomplished by calculating a weighted average of the output exemplars using the similarities of the query to the input exemplars as weighting parameters.

Now that the general process used in KR has been presented, this process can be related to Eq. (1.1), beginning with the distance calculation. The distance can be seen to be simply the difference of the input exemplar and the query:

$$d(x_i, x) = |x_i - x| \ .$$

Finally, the estimated similarities are used to perform a weighted average of the output exemplars. The sum of the weighted output exemplars is divided by the sum of the weights. This operation is simply a normalization that allows for the prediction to be represented as a combination of the output exemplars, where each exemplar can have an influence of 0 to 1 (i.e., 0–100%).

In this section, a top-level description of the KR prediction process was presented. In the next section, the methods used to quantify similarity (i.e., calculate the weights) are discussed.

1.3.2 Distance measures

A common distance function is the Euclidean distance, which is also known as the L^2-norm. For a single input, the Euclidean distance for the *i*th input exemplar and the query is given by

$$d(X_i, x) = \sqrt{(X_i - \cdots x)^2} \ . \qquad (1.2)$$

For p inputs, the Euclidean distance is given by

$$d(X_i, x) = \sqrt{(X_{i,1} - x_1)^2 + (X_{i,2} - x_2)^2 + \ldots + (X_{i,p} - x_p)^2}\ , \tag{1.3}$$

where X_i is the ith exemplar observation of the p inputs, and Xi,j is the ith exemplar observation of the jth input, and

x is the query observation of the p inputs, where x_j is the query observation of the jth input.

There are alternative distance metrics that can be used instead of the Euclidean. For instance, the adaptive Euclidean distance recently developed at The University of Tennessee (Garvey 2006) drops observations that lie outside of the training range (i.e., outside the minimum and maximum input exemplars) from the distance calculation. Chapter 7 explores the different distance metrics and examines the impact that they have on model performance.

Now that methods for calculating the distance of the input query to the input exemplars have been introduced, a method to use these distances to infer similarity is presented. In the next section, the different kernel functions that can be used to map distances to similarities are discussed.

1.3.3 Kernel functions

To transform the distance into a similarity or weight, a kernel function is used. In general, a kernel function should have large values for small distances and small values for large distances. In other words, when a query point is nearly identical to a reference point, its distance should be small, and, therefore, that particular reference point should receive a large weight, and vice versa. One commonly used function that satisfies this criterion is the Gaussian kernel (Fan and Gijbels, 1996),

$$K(h) = \frac{1}{\sqrt{2\pi h^2}} e^{-d^2 / 2h^2}\ . \tag{1.4}$$

Here, h is commonly referred to as the kernel's bandwidth and is used to control what effective distances are deemed similar. For mean-centered, unit variance scaled data, the bandwidth generally has values of $0<h<2$. It can be seen in Fig. 1-3 that the Gaussian kernel with the smaller bandwidth ($h = 0.1$) will only generate large weights when the distance is very close to zero, while the kernel with the larger bandwidth is less specific and will assign significant weights for a larger range of distances.

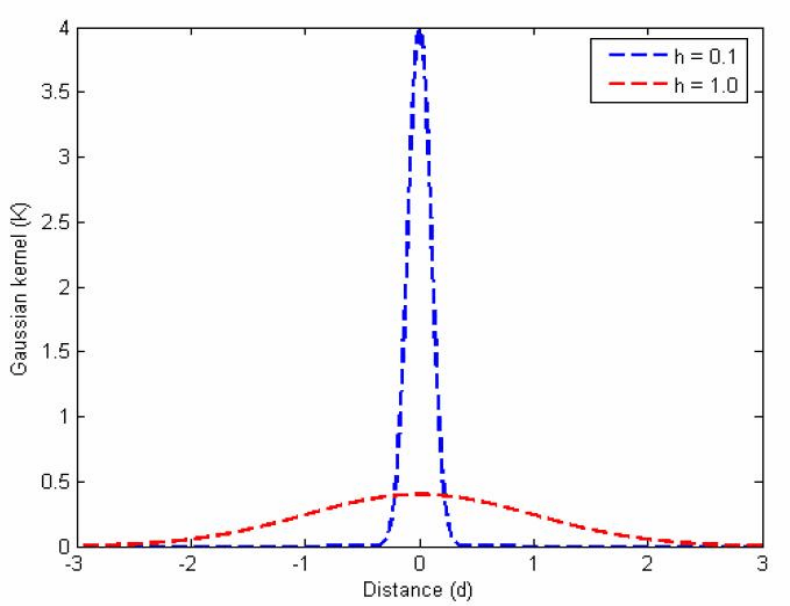


Fig. 1-3. Example Gaussian kernels.

Other kernel functions include the inverse distance, exponential, absolute exponential, uniform weighting, triangular, biquadratic, and tricube kernel (Atkeson et al. 1997a), examples of which may be seen in Fig. 1-4. More advanced kernel functions include the Hermite kernel (Zavaljevski and Gross 2000a, 2000b) and the asymmetric Gaussian kernel (Mackenzie and Tieu 2004). Although each function may have advantages or disadvantages in certain situations (Cleveland and Loader 1994a), the Gaussian kernel function is generally an adequate selection. In fact, the work of Scott (1992) and Cleveland and Loader (1994b) show that kernel function selection plays a noncritical role in the performance of locally weighted models. For this reason, all of the AAKR models presented in this report employ the Gaussian kernel.

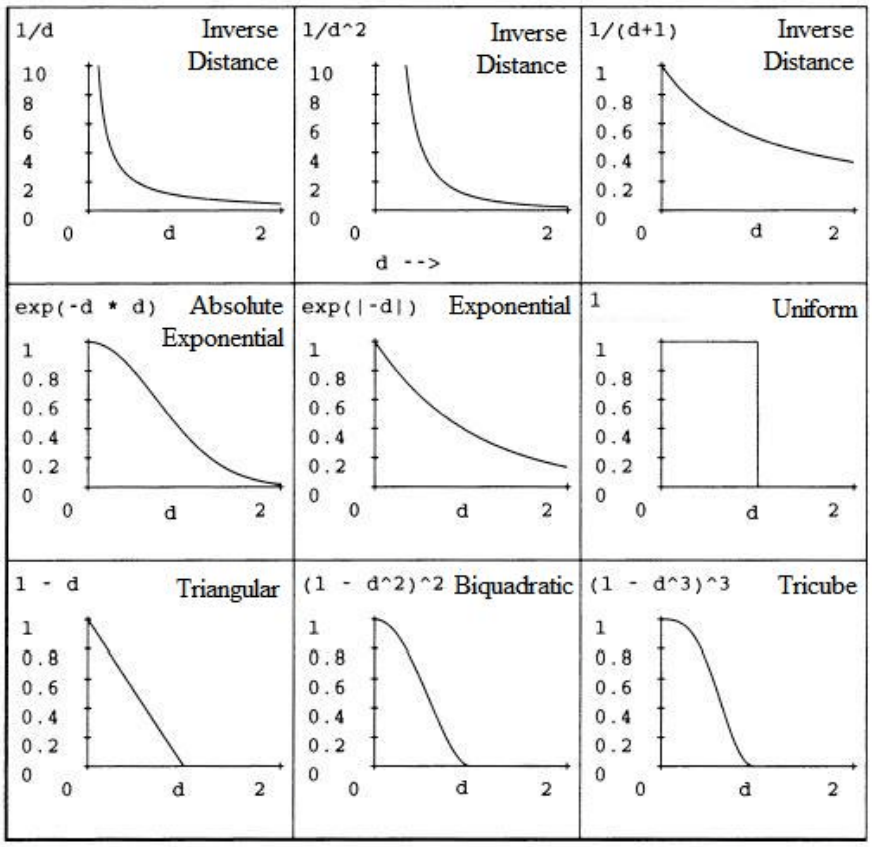


Fig. 1-4. Examples of alternative kernel functions.

Now that the methods for mapping an input query to similarities have been discussed, the different model architectures may be examined in more detail. In the next section, the different KR architectures will be defined, and the relevant equations for prediction are presented.

1.3.4 Model architecture

There are three different KR architectures, which are characterized by the number and type of inputs and outputs. These model architectures are inferential, hetero-associative, and auto-associative KR. As seen in Fig. 1-5, an inferential model uses multiple inputs to infer an output, while a hetero-associative model uses multiple inputs to predict multiple outputs, and an auto-associative model uses inputs to predict the "correct" values for the inputs, where "correct" refers to the relationships and behaviors contained in the exemplar observations. Although the auto-associative architecture is the architecture most suited for OLM, the KR prediction algorithm for each of the model architectures will be examined, because understanding each architecture provides a solid foundation for understanding general KR's extension to the auto-associative case. The description of each architecture focuses on how the three-step process described earlier changes for the different model architectures. These discussions do not draw from a single reference but are an aggregation of the work of Hardle (1989), Wand and Jones (1996), Fan and Gijbels (1996), and Atkeson et al. (1997a, 1997b).

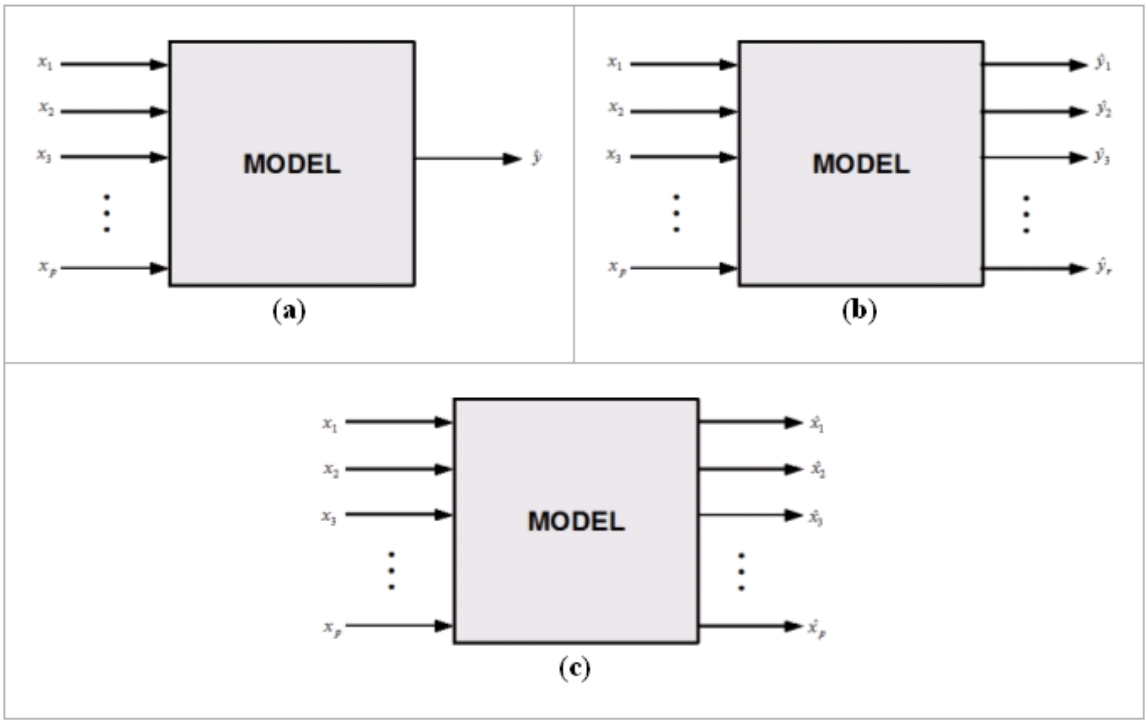


Fig. 1-5. Illustration of (a) inferential, (b) hetero-associative, and (c) auto-associative model architectures.

Before examining the different KR architectures, several conventions and notational simplifications need to be discussed. For the remainder of this section the query observation is assumed to be within the training range, which allows the distance metric to be represented by Eq. (1.3). Also, in KR, the weighted sum of the outputs is normalized by the sum of the weights. To simplify the notation, the following definition is made:

$$a = \sum_{i=1}^{n} w_i \, . \tag{1.5}$$

Using this definition and the definition of the weights [Eq. (1.2)], Eq. (1.1) can be rewritten as:

$$\hat{y}(x) = \frac{1}{a} \sum_{i=1}^{n} \left(w_i X_i \right) \, . \tag{1.6}$$

1.3.5 Inferential kernel regression

The first architecture that will be considered is the inferential KR model. In this architecture, p inputs (predictor variables) are used to predict a single output (response variable). The output may or may not be one of the inputs, but in most cases it is not. Inferential KR models most often use system

parameters to infer the value of another, related parameter. The three step KR prediction process is now reviewed.

Step 1—Distance Calculation

Equation (1.3) can be directly applied to the p inputs. For convenience, Eq. (1.3) is reproduced below:

$$d(X_i, x) = \sqrt{(X_{i,1} - x_1)^2 + (X_{i,2} - x_2)^2 + \ldots + (X_{i,p} - x_p)^2} \quad .$$

The result of calculating the distances for the n input exemplars is a vector of n distances:

$$d = \begin{bmatrix} d(X_1, x) \\ d(X_2, x) \\ M \\ d(X_n, x) \end{bmatrix} \quad . \tag{1.7}$$

Step 2—Similarity Quantification

Next, the Gaussian kernel [Eq. (1.4)] is evaluated with the calculated distances **d**.

$$K(h) = \frac{1}{\sqrt{2\pi h^2}} e^{-d^2 / 2h^2} \quad .$$

The result is a vector of n weights:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ M \\ w_n \end{bmatrix} \quad . \tag{1.8}$$

Step 3—Output Estimation

Finally, Eq. (1.6) can be applied directly for prediction of the one output:

$$\hat{y}(x) = \frac{1}{a}\sum_{i=1}^{n}(w_i Y_i) = \frac{1}{a}(w_1 Y_1 + w_2 Y_2 + \ldots + w_n Y_n) \ . \tag{1.9}$$

1.3.6 Heteroassociative kernel regression

At this point, the inferential KR model can be modified to handle multiple outputs. In general, a hetero-associative model uses *p* inputs to predict *r* outputs. Again, the outputs may or may not be inputs.

Because the first two steps of the KR prediction process measure the similarity of the query inputs to the input exemplars, the only change occurs in the third step of the prediction process. The first two steps are as previously described.

Step 3—Output Estimation

Therefore, for a vector of *n* weights, the predictions of the hetero-associative KR model are calculated for the *r* outputs according to the following equation, where $Y_{i,j}$ is the ith exemplar of the jth output and where $\hat{y}(x)$ is a row vector of length r.

$$\hat{y}(x) = \frac{1}{a}\left[\sum_{i=1}^{n}(w_i Y_{i,1}) \ \sum_{i=1}^{n}(w_i Y_{i,2}) \ldots \sum_{i=1}^{n}(w_i Y_{i,r})\right] \ . \tag{1.10}$$

1.3.7 Auto-associative kernel regression

The final KR formulation is auto-associative KR (AAKR), which estimates the "correct" versions of an input vector. In this case, the dimension of the input vector (p) is the same as the dimension of the output vector. For the sake of clarity, the process diagram for KR prediction (Fig. 1-2) is modified for an auto-associative architecture in Fig. 1-6. The AAKR diagram does not include output exemplars; instead, the input exemplars perform the same action as the output exemplars.

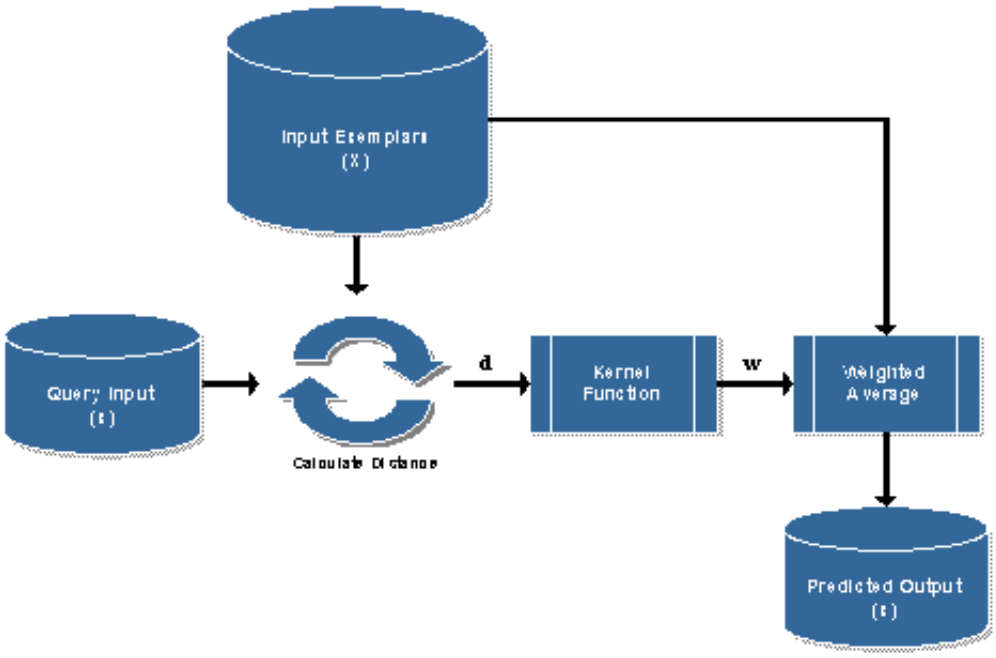


Fig. 1-6. Process diagram for the AAKR prediction algorithm.

As with hetero-associative KR, the only difference between AAKR and inferential KR lies in the final step of the prediction process.

Step 3—Output Estimation

For AAKR, the outputs are predicted as a weighted average of the error-free exemplar vectors. Putting this into equation form yields the following equation, where $X_{i,j}$ is the ith exemplar of the jth input:

$$\hat{x}(x) = \frac{1}{a}\left[\sum_{i=1}^{n}(w_i X_{i,1}) \quad \sum_{i=1}^{n}(w_i X_{i,2}) \ldots \sum_{i=1}^{n}(w_i X_{i,p})\right] . \tag{1.11}$$

As seen in Eq. (1.11), an auto-associative architecture is able to predict a group of correct sensor values even when supplied with a group of sensor values that is usually corrupted with process and instrument noise and also contains faults such as sensor drift or complete failure. As such, AAKR is especially suited for OLM and is the model used to carry out each limiting case study. Each case study presented in this report is characterized by the model's characteristics, the model's performance metrics, and its uncertainty. The model results for each case study are presented in tables. Short definitions are given below for some of the model metrics reported in the model results tables in the following sections. These definitions are grouped according to the table sections shown in Table 1-2.

Table 1-2. Categories of model characterization

Model
Data cleaning
Number of memory vectors
Optimal kernel width
Vector selection method
Metrics
Accuracy (% of span)
Auto-sensitivity
Cross-sensitivity
EULM detectability (% of span)
SPRT detectability (% of span)
Uncertainty
Analytic (% of span)
Coverage
Monte Carlo (% of span)
Coverage

1.4 Model Characteristics

As shown in Table 1-2, an AAKR model is characterized by the data cleaning, the number of memory vectors, the optimal kernel width, and the vector selection method. Each of these modeling aspects will now be briefly described.

Data cleaning can be defined as a process that identifies bad data points and then either removes them or replaces them, using the estimates of an algorithm. Many data-cleaning methods exist to detect and correct NaNs, stuck data, and outliers; all of which have been found to occur in data extracted from power plant data historians (Hines and Davis 2005). Data cleaning may only be performed if inspection of the data suggests it is necessary. The effect of noise in the data sets is investigated in Chapter 6. Some of these cleaning methods and the effects of not using or improperly applying them are discussed in Chapter 7.

The **number of memory vectors** or observations that are selected from the training set to be used in an empirical model is an important part of model development and optimization. The number of memory vectors retained in a memory based model controls how many operating points are used to represent the process or equipment being monitored. The impact of using too few or too many memory vectors is examined in Chapter 9.

The **kernel bandwidth** is used to compare query vectors with memory vectors in memory based models. The bandwidth effectively controls how many memory vectors are used to make predictions. Large bandwidths produce smoother model predictions, as many memory vectors are used to infer a parameter's value. Conversely, small bandwidths produce rough and/or inconsistent predictions because a limited number, if any, of the memory vectors are used to infer a parameter's value. In other words, the bandwidth controls how far a query observation may be from a memory vector to be deemed similar. For

the studies reported in this volume, the optimal kernel bandwidth was always used. The bandwidth was optimized by minimizing the error of the test data set. This is commonly termed cross validation optimization.

The model architecture uses a **vector selection method** to identify memory vectors for model development. There are several available methods, but the most commonly used ones are vector ordering, minimum/maximum selection, and a combination of the two. The combination selection method is the default vector-selection method and is used to develop the models presented in the case studies. In reporting the case studies, the combination selection method is denoted by an *x* in the table. The effect on model performance of using different vector selection methods is examined in Chapter 8.

1.5 Performance Metrics

The performance of auto-associative OLM systems has historically been quantified in terms of accuracy, with more recent emphasis on detectability and sensitivity measures. Accuracy measures the ability of a model to correctly and accurately predict sensor values and is normally presented as the mean squared error (MSE) between sensor predictions and the measured sensor values. Two sensitivity measures are used to quantify a model's ability to make correct sensor predictions when an input sensor's value is incorrect due to some sort of fault. Finally, Error Uncertainty Limit Monitoring (EULM) detectability and Sequential Probability Ratio Test (SPRT) detectability quantify the smallest sensor calibration fault and anomaly that may be identified by an empirical model, respectively. An ideal model would be accurate, would have sensor predictions that are not appreciably affected by degraded inputs, and would be able to detect small sensor faults and anomalies. As these metrics will be used to determine and compare the performance of the models applied to the limiting case studies, the remainder of this section provides the mathematical framework of the performance metrics in more detail.

1.5.1 Accuracy

The **accuracy** metric is simply defined as the mean squared error (MSE) between the model's predictions and the desired output. A model is developed with training data, and then test data are used to compute the accuracy. This metric is a measure of the model's error for data it has not seen before. The equation for a single variable is simply:

$$A = \frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2 \quad , \tag{1.12}$$

where

N is the number of test observations,
$\hat{x}_i$ is the model prediction of the ith test observation, and
x_i is the ith observation of the test data.

Although this metric is termed "accuracy," it is actually a measure of error, and a low value is desired. The accuracy metric is the most commonly cited metric because it represents a model's performance for unfaulted input data. However, since the purpose of empirical modeling in OLM systems is to identify sensor and process faults, model performance under faulted conditions must be quantified.

1.5.2 Sensitivity metrics

In general terms, sensitivity quantifies the model's response to faulted inputs. To calculate the **sensitivity** of a prediction, a model's response using fault-free input data is first calculated. Next, each

input variable is sequentially artificially faulted, and the model outputs are recorded. These predictions using faulty input data are then used to determine the model's sensitivity metrics.

Model sensitivity is a measure of the change in an output prediction ($\hat{x}_i$) produced by a change in an input (x_i):

$$S_i = \frac{\Delta \hat{x}_i}{\Delta x_i} \quad . \tag{1.13}$$

Auto-sensitivity (S_{auto}), also occasionally referred to as "robustness," is a measure of an empirical model's ability to make correct sensor predictions when its respective input sensor value is incorrect due to some sort of fault. Therefore, this metric involves the following values: sensor *i*'s prediction with no fault in the input ($\hat{x}_i$), sensor *i*'s prediction with a faulted input ($\hat{x}_i^{drift}$) sensor *i*'s unfaulted input value (x_i), and the drifted input value (x_i^{draft}). The measure is averaged over an operating region defined by k samples. Using these definitions, the autosensitivity for sensor *i* is given by the following:

$$SA_i = \frac{\sum_{k=1}^{N} \left| \frac{\hat{x}_{ki}^{drift} - \hat{x}_{ki}}{x_{ki}^{drift} - x_{ki}} \right|}{N} \tag{1.14}$$

If a model's autosensitivity is 1, then the model's prediction follows the fault, resulting in a residual of zero, and the fault cannot be detected. If a model's robustness value is nonzero, which is usually the case, the residual will underestimate the size of the sensor fault and the OLM system drift limits may need to be compensated accordingly.

The next performance metric is **cross-sensitivity**, also referred to as spillover. This value measures the effect that a faulty sensor input (i) has on the predictions of sensor (j). This is illustrated by Eq. (1.15), in which *j* is the index of the unfaulted variable whose cross-sensitivity (SC_j) metric is being calculated:

$$SC_j = \frac{\sum_{k=1}^{N} \left| \frac{\hat{x}_{kj}^{drift} - \hat{x}_{kj}^{t}}{x_{ki}^{drift} - x_{ki}^{t}} \right|}{N} \quad . \tag{1.15}$$

The plots presented in Fig. 1-7 more clearly illustrate the concept of sensitivity. Two plots are included to illustrate the differences between an empirical model with small (upper plot) and large (lower plot) sensitivity metrics. These sensitivity metrics could be either auto- or cross-sensitivity, depending on which input fault is causing the deviation. In both plots, the points indicate the true process parameter value. The blue line drifting to the bottom at the highest rate indicates an artificially drifted sensor value that is supplied as an input to the empirical model, and the red upper line represents the model predictions

of the process parameter. The model predictions with a small sensitivity (upper plot) lie very near the normal, undrifted data. This indicates that the model is not significantly affected by the drifted input and is able to accurately predict the parameter's actual value when supplied with faulty input. This model is considered to be a "robust" model because the prediction is robust to measurement errors. Conversely, the plot for the model with a large sensitivity metric (lower plot) shows the model's predictions lie very near the artificially drifted values. Such a model would be of little use in instrument calibration validation because its predictions follow the faulted inputs rather than correcting them.

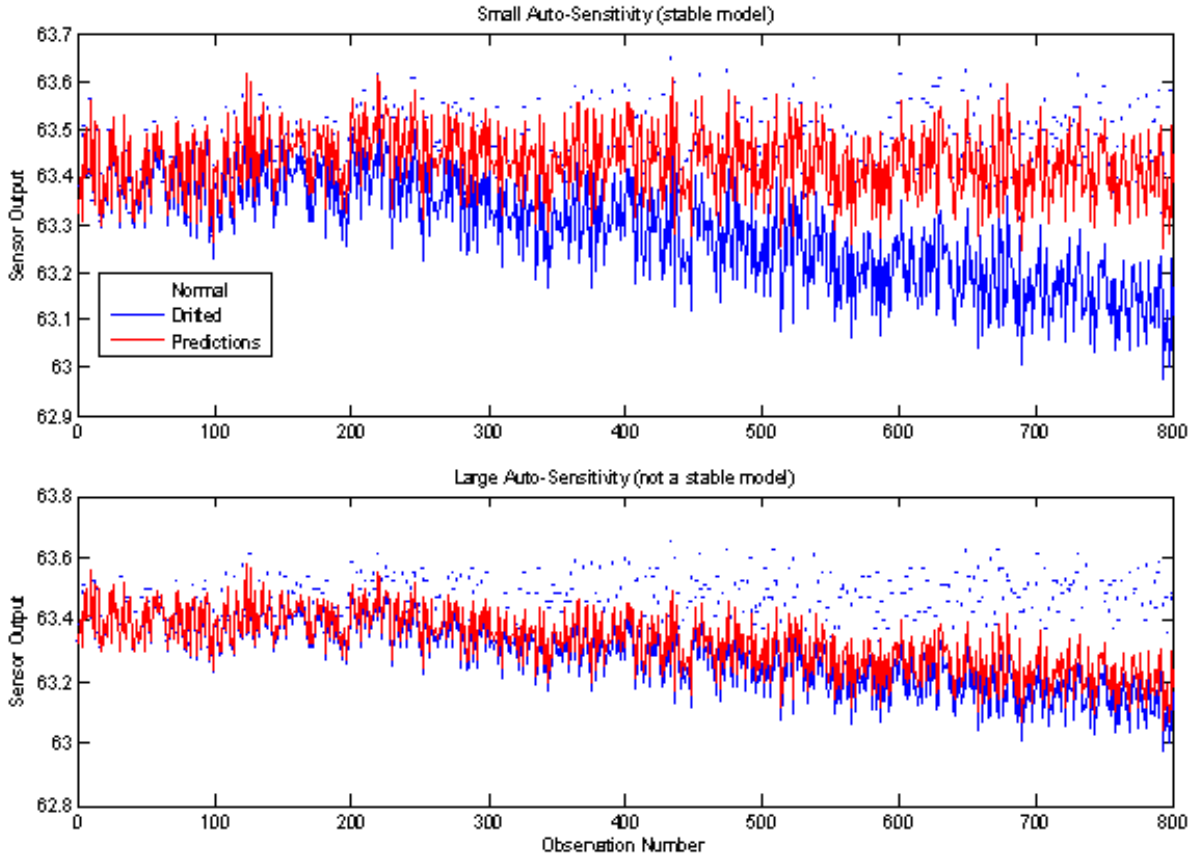


Fig. 1-7. Illustration of sensitivity performance metric.

Note that the accuracy metric is calculated for each variable, that the auto-sensitivity metric is calculated for the artificially drifted variable, and that the cross-sensitivity metric is calculated for each variable except the artificially drifted variable. For a model with five variables, if a single variable is drifted, then there are five accuracy values, one autosensitivity (robustness) value, and four nonzero (cross-sensitivity) spillover metric values. To quantify the sensitivity values for each variable, the algorithm is structured so that all the variables are sequentially perturbed. The auto-sensitivity metric and the mean cross-sensitivity metric are calculated for each variable.

1.5.3 Detectability metrics

Detectability metrics measure the smallest fault that can be detected by an empirical model. Two fault detectability metrics are used to evaluate model performance: Error Uncertainty Limit Monitoring (EULM) detectability and Sequential Probability Ratio Testing (SPRT) detectability. These detectability metrics are positive values that indicate the detectability as a percentage of the mean sensor value.

Before presenting the details of these detectability metrics, the generic process used in instrument calibration monitoring system is presented in Fig. 1-8.

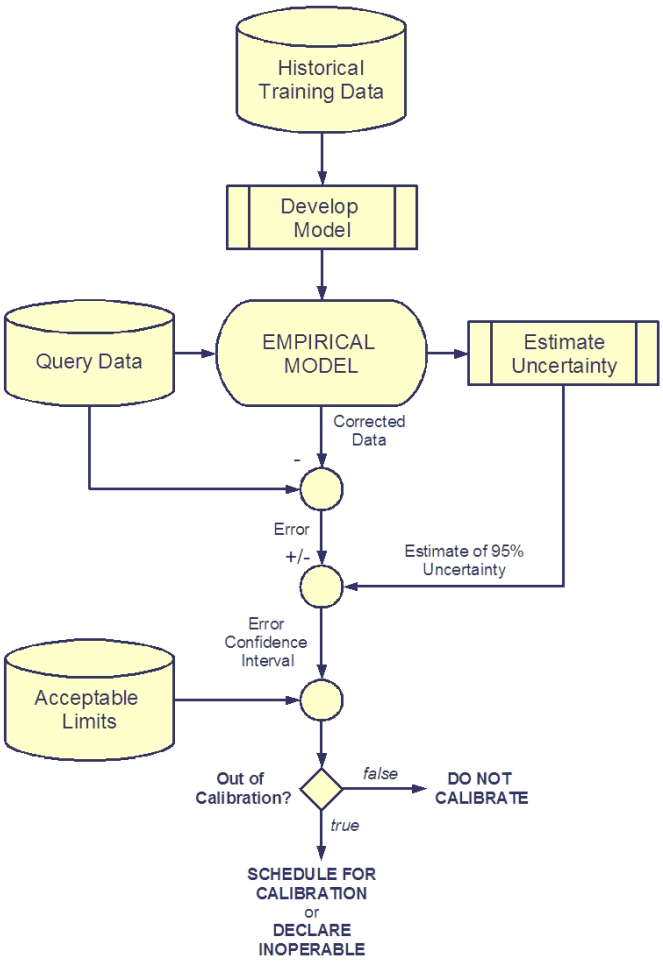


Fig. 1-8. Process diagram for generic instrument calibration system.

It can be seen that the first step in developing a calibration monitoring system is to collect training data and develop an empirical model. Next, query data are presented to the developed model, which corrects the data. The actual values are subtracted from the corrected values (where corrected values refer to a model's estimates), providing the model's prediction error, or bias. In addition, the 95% uncertainty, or variance, of the model's estimates is calculated and combined with the error to form its 95% confidence interval (CI). This CI is then compared to acceptable calibration limits and may result in one of three possible choices: (1) not to calibrate, (2) to schedule a calibration, or (3) to declare the sensor to be inoperable. From this discussion, it is clear that the empirical model performs the most critical task of the calibration monitoring system by computing the process parameter estimate for a given system state. In addition, the empirical model performs a correcting action and can effectively employ an auto-associative architecture.

From this discussion, it is clear that the extent to which an empirical model can detect a sensor fault is affected by two factors: its predictive uncertainty and its robustness. This dependency will now be used in an example to fully develop the detectability performance metrics.

Error uncertainty limit monitoring (EULM) is a method of fault detection that monitors the uncertainty of the prediction errors relative to some specified tolerance. A sensor is identified as "faulted" when the drift and model uncertainty exceeds the specified tolerance. This detection method is most generally applied to sensor calibration validation tasks where a sensor is allowed to drift above or below its nominal value by a certain percentage.

To understand EULM, it is necessary to first suppose that an empirical model of a sensor group has been developed, whose uncertainty of sensor *i* (95% CI) is found to be 1% of its nominal value. In the ideal situation, the sensor's prediction would be insensitive to input faults, and the smallest expected fault that the model could detect would be the magnitude of its uncertainty or 1%. In other words, when the residual grew to >1%, one could be 95% certain that it was due to a drift and not due to the uncertainty of the prediction. Since the ideal is rarely a reality, an empirical model's predictions can be expected to drift slightly when an input is faulted. When the auto-sensitivity is greater than zero, the sensor must drift by more than 1% for the residual to have a magnitude of 1%. In Eq. (1.16), auto-sensitivity for the *i*th sensor is rewritten as the change in the prediction for a drifted input divided by the drift amount.

$$\Delta x_i^{drift} = x_i^{drift} - x_i \quad and \quad \Delta \hat{x}_i^{drift} = \hat{x}_i^{drift} - \hat{x}_i$$
$$S_{Ai} = \frac{\Delta \hat{x}_i^{drift}}{\Delta x_i^{drift}}. \tag{1.16}$$

The residual is the difference between the input and the output with the input being the normal input plus the drift, and the output being the normal output plus the portion of the drift (S_A) that appears on the prediction:

$$Residual_i = r_i = x_i + \Delta x_i^{drift} - \hat{x}_i - \Delta \hat{x}_i^{drift} \tag{1.17}$$

If the model is accurate, the prediction equals the actual value of x_i, and Eq. (1.17) further simplifies to the following:

$$r_i = \Delta x_i^{drift} - \Delta \hat{x}_i^{drift} \ . \tag{1.18}$$

Using Eq. (1.16), a measure of the residual can be obtained in terms of autosensitivity:

$$r_i = \Delta x_i^{drift} - SA_i(\Delta x_i^{drift}) = \Delta x_i^{drift}(1 - SA_i) \ . \tag{1.19}$$

Equation (1.19) says that the residual is only equal to a percentage of the actual drift. For example, if the sensitivity is 0.2, then the sensor must drift by 1% /1–0.2) for the residual to equal 1%. From this consideration, the actual EULM detectability metric is derived and shown in Eq. (1.20):

$$D_{EULM\,i} = \frac{U_i}{Span_i} \Big/ (1 - SA_i) \ , \tag{1.20}$$

where U_i is the uncertainty in the ith sensor and $Span_i$ is the span of the ith sensor.

The detectability has units of percentage of sensor span and is a function of both model uncertainty and model auto-sensitivity. A greater auto-sensitivity (S_A) causes the denominator of Eq. (1.20) to get smaller and the detectability (D_i) to get bigger; meaning that only larger drifts can be confidently detected. An optimal model would have an auto-sensitivity equal to zero.

The Sequential Probability Ratio Testing (SPRT) fault detection method examines an error distribution and detects changes in its mean and variance. For a detailed explanation of using SPRT for sensor surveillance, see Volume 2 of this NUREG or an article by Humenik and Gross (1990). **SPRT Detectability** measures the smallest process parameter change that can be detected with a given false alarm rate. Stated a more general way, SPRT tests whether the sequence of residuals is being generated by a random process with a mean of zero or by a process with a non-zero mean due to some fault condition. This measure is generally used for process monitoring, not instrument calibration verification (ICV). It does not directly take uncertainty into consideration.

The general procedure for the SPRT is to first calculate the likelihood ratio, which is given by the following equation where $\{x_n\}$ is a sequence of n consecutive observations of x:

$$L_n = \frac{probability\ of\ observing\ \{x_n\}\ given\ H_1\ is\ true}{probability\ of\ observing\ \{x_n\}\ given\ H_0\ is\ true} = \frac{p(\{x_n\}/H_1)}{p(\{x_n\}/H_0)} \ . \tag{1.21}$$

The likelihood ratio is then compared to a lower limit (A) and an upper limit (B) defined by the false alarm probability (α) and missed alarm probability (β) as follows:

$$A = \frac{\beta}{1-\alpha} \quad and \quad B = \frac{1-\beta}{\alpha} \ . \tag{1.22}$$

If the likelihood ratio is less than A, then the system's normal mode is classified as belonging to H_0. Conversely, if the likelihood ratio is greater than B, it is classified as belonging to the system's degraded mode H_1 and a fault is registered. If the ratio is between A and B, the state is indeterminate. For this work, the SPRT is applied to the residual between a sensor measurement and an empirical model's prediction of the parameter. The residuals are assumed to be normally distributed with a mean of 0 and variance of σ^2, which is an estimate of the random variation of the sensor signal. Therefore, the probability distribution function (pdf) for the normal mode of the residuals is given by the following:

$$p(x/H_0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) . \tag{1.23}$$

From this description, two degradation modes are readily apparent and are shown in Fig. 1-9. In the first plot, there is a mean shift up (+M), and the second plot shows a mean shift down (-M). The random uncertainty is denoted by the spread of the Gaussian function. The SPRT simply determines whether the residual sequence is more probably generated from the normal or from the faulted distributions.

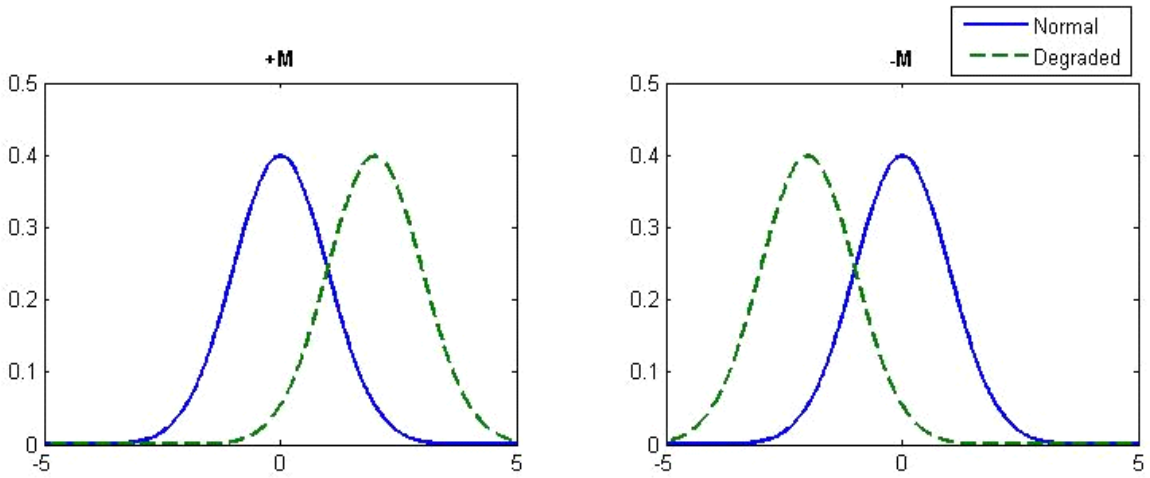


Fig. 1-9. Illustration of degraded modes for a normal distribution.

The derivation of the likelihood ratios is beyond the scope of this paper but can be found in Humenik (1990). The natural logarithms of the likelihood ratios are compared to *ln A* and *ln B* in most implementations of the SPRT algorithm.

The magnitude of a sensor change caused by an anomaly or fault that can be reliably detected by the SPRT is defined as the magnitude of M. If the observed values consistently lie near ±M, the residual sequence is more likely to be generated from the ±M distribution than from the normal distribution around 0.

For this work, the optimal M value is determined numerically by applying the SPRT to unfaulted test data and locating the M value that results in a false-alarm probability that is nearest the desired false alarm probability . If the optimal M value is estimated to be $\hat{M}$, then the SPRT detection performance metric in fraction of sensor span is given by Eq. (1.24):

$$D_{SPRT_i} = \frac{\hat{M}}{Span_i} . \tag{1.24}$$

For the case studies presented, the false alarm probability was set to 0.05 or 5%, and the missed alarm probability was set to 0.10 or 10%. The missed alarm value is allowed to be fairly high, because if the algorithm does not detect a fault at time t, it will probably detect it at time t+1.

1.5.4 Uncertainty

The **analytic** and **Monte Carlo** methods for calculating OLM uncertainty are described in the previous NUREG volumes and by Hines (2005). Analytic uncertainty is estimated with the analytic equations derived from the model's mathematical architecture. This method produces an uncertainty estimate with each prediction. Monte Carlo uncertainty is estimated by applying a Monte Carlo resampling technique. With Monte Carlo techniques, the training data are resampled multiple times, and for each of these resampled datasets, a new model is constructed. The variation between all of these models is then taken as a measure of the variance portion of the total uncertainty. The Monte Carlo methods produce an average uncertainty over the operating region. Because the Monte Carlo methods measure the uncertainty of all possible models, rather than just the model at hand, their uncertainty estimates are inherently slightly larger than those calculated by the analytic methods. However, research has shown that both techniques generally provide a conservative estimate of model uncertainty (Rasmussen 2003).

In OLM, the uncertainty is calculated for the residuals (the difference between the actual signal and the model predictions) of each model. This uncertainty is used to construct a confidence interval centered at zero, the expected value of the denoised residuals. The residual coverage is then calculated as the fraction of denoised residuals contained within the confidence interval. A 95% confidence interval should, on the average, result in 95% coverage. Denoising is discussed in Chapter 7.

1.6 Data-Set Description

Signal selection is a key step in model development. Several important considerations for this process are outlined in EPRI's *On-Line Monitoring of Instrument Channel Performance. Volume 1*: *Guidelines for Model Development and Implementation* (2004). Most nonparametric modeling architectures including AAKR, assume that each signal in a data set is correlated with at least one other signal. Optimal grouping techniques divide a large set of signals to be monitored into smaller, highly correlated groups. These smaller groups may consist of purely redundant signals or may contain several sets of redundant signals that are correlated with each other. In addition to signal correlation, data-set size (the number of signals) is an important consideration in signal selection and grouping. Even if a large group of signals can be identified as highly correlated, models with many signals may be computationally intensive. Computer memory and model run-time may become limiting factors for model size. Models typically include between 3 and 30 signals, although models with as many as 80 signals are generally acceptable. Two very different types of data sets are used to monitor nuclear power plant processes: small data sets with one set of redundant sensors and large data sets with several distinct sets of redundant sensors. The results using both data set types are presented here: one small redundant sensor set (Pressurizer-Level) and one large nonredundant sensor set (reactor protection system loop A). The sensor models were chosen from Volume 2 of EPRI's report on modeling guidelines (EPRI 2004).

The Pressurizer-Level model includes three redundant Pressurizer-Level sensors (Fig. 1-10). Table 1-3 includes an estimation of the signal noise (see Chapter 6 for further information) and the correlation coefficients of the three sensors. It can be seen that the correlation coefficients for each pair of sensors are extremely high.

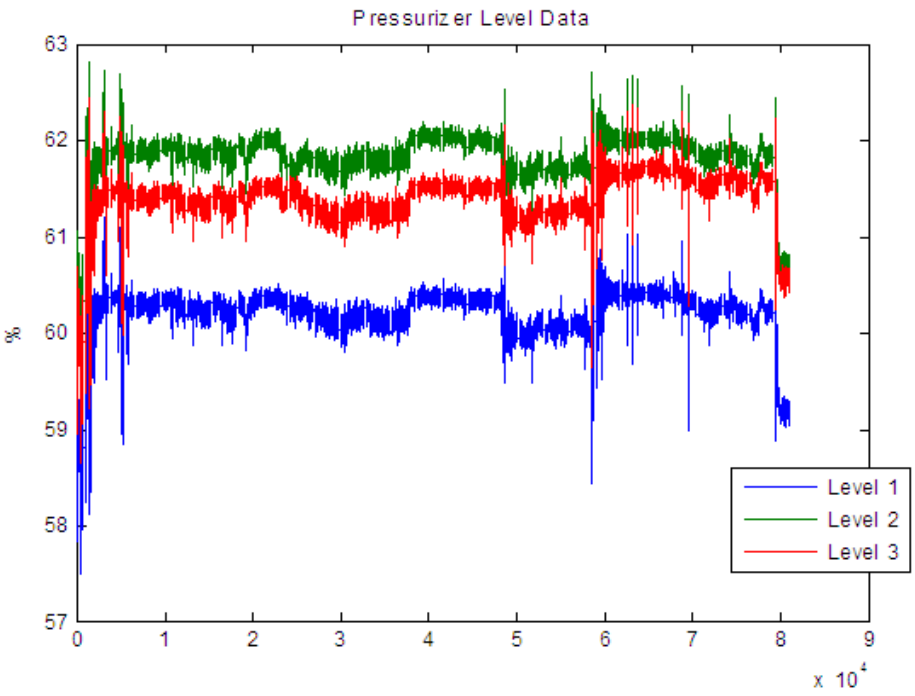


Fig. 1-10. Data for the Pressurizer-Level model.

Table 1-3. Data characteristics for Pressurizer-Level model

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Data			
Signal noise estimate (%)	0.039	0.0366	0.038
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000

The reactor protection system (RPS) Loop A models include nine sensors (Fig. 1-11): two feedwater flow sensors (signals 1 and 2), two steam flow sensors (signals 3 and 4), two turbine-pressure sensors (signals 5 and 6), and three steam pressure sensors (signals 7, 8, and 9). The four redundant sensor groups have high intercorrelations and are also highly correlated with each other. Table 1-4 includes an estimation of the signal noise and the correlation coefficients of the three sensors. It can be seen that the correlation coefficient between three sets of the redundant sensors is high: the feedwater flow sensors, turbine pressure sensors, and steam pressure sensors. However, the three steam pressure sensors are only moderately correlated with the other three groups. Additionally, the two steam flow sensors have the smallest intercorrelations (0.681). This is likely due to the high level of noise inherent in steam-flow sensors.

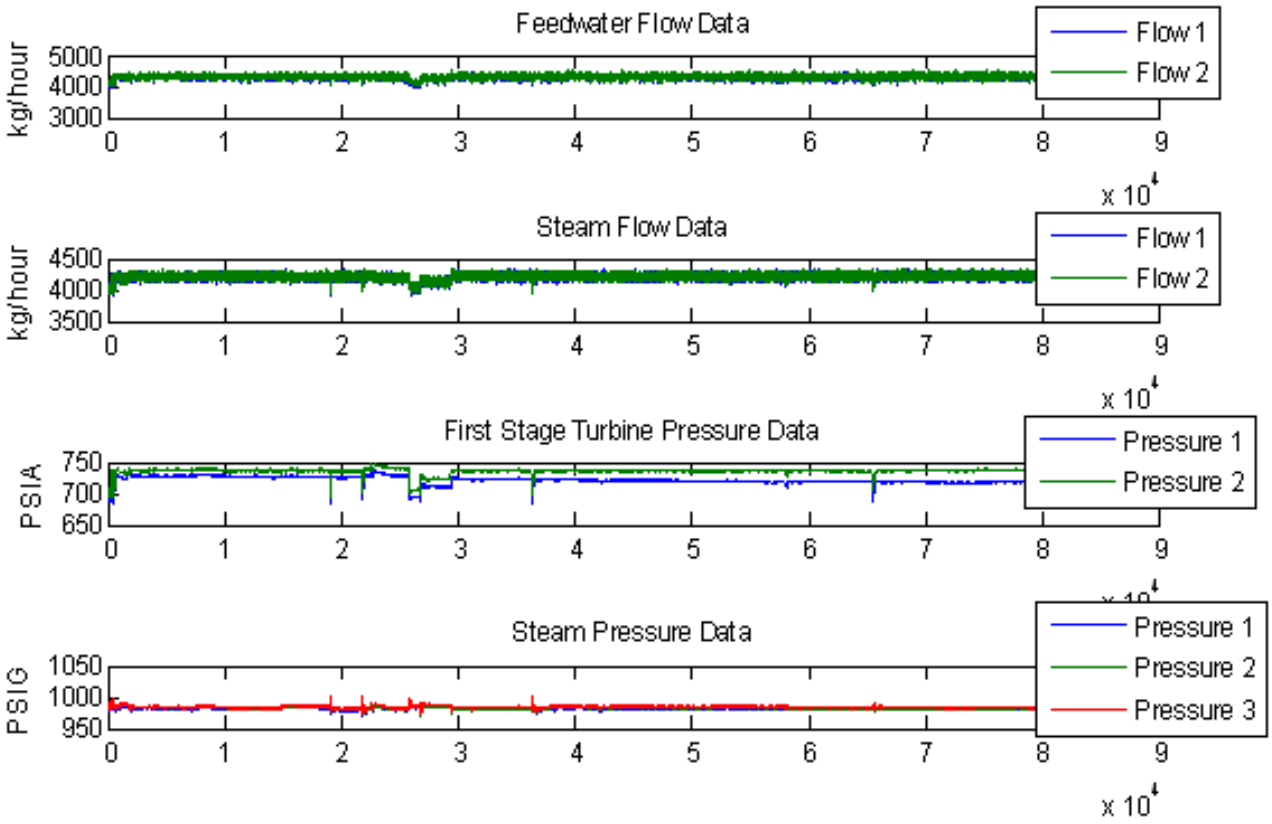


Fig. 1-11. Data for RPS Loop A model.

Table 1-4. Data characteristics for RPS Loop A model

RPS Loop A data									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Signal noise estimate (%)	0.558	0.541	0.607	0.608	0.039	0.038	0.028	0.027	0.033
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000

Data were collected for these 12 variables, along with reactor power level data, for the 14-month period from March 2001 through April 2002. The models developed with this data are valid only for reactor operation in the "high-power phase," that is, a power level greater than 960 MW(e). The "high-power" data for each model are divided into three sets: training data, test data, and validation data. The

training data are used to develop the model. Test data are used to optimize the model; the results presented here include an optimization of the kernel bandwidth for each model. Finally, the validation data are used to characterize the performance of the model, including performance metrics, model uncertainty, and model coverage. The training data were chosen to be the first 30,000 observations shown in Figs. 1-10 and 1-11 above. This includes "high-power" data from March 2001 to June 2001 with a 5-minute sampling rate. The test data were chosen to be the following 22,500 observations. This includes data from July 2001 to mid-August 2001, with a 3-minute sampling rate. Finally, the validation data were chosen to be the remaining 28,500 observations. This includes data from mid-August 2001 to mid-October 2001. These data ranges were chosen after a visual inspection of the data indicated that no significant data faults were present in any of the three ranges.

Page intentionally blank

2. MODELING METHODS

The method used for model development and analysis is outlined in Fig. 2-1. Three data sets are used for each case study: training data, test data, and validation data. Before model development begins, these data should be visually inspected to determine if any outliers are present. Automated methods for data cleaning have also been developed to identify common data collection and storage problems, but only visual inspection is used in this study.

In the first step of this analysis, training data are used for initial model development. In this step, exemplar observations are chosen from the training data to form the subset of memory vectors; this accounts for all the "training" needed by an AAKR model. The second step involves optimizing the model architecture. This optimization is accomplished using the test data set. The models presented in this research are optimized only for the kernel bandwidth. Models can also be optimized for other parameters, including the number of memory vectors, the vector selection method, and the distance measure. Finally, the validation data set is used to quantify performance measures for the optimized model. This involved evaluating five model metrics: accuracy, auto-sensitivity, cross-sensitivity, EULM detectability, and SPRT detectability. The model is also analyzed for both analytic and Monte Carlo uncertainty estimates.

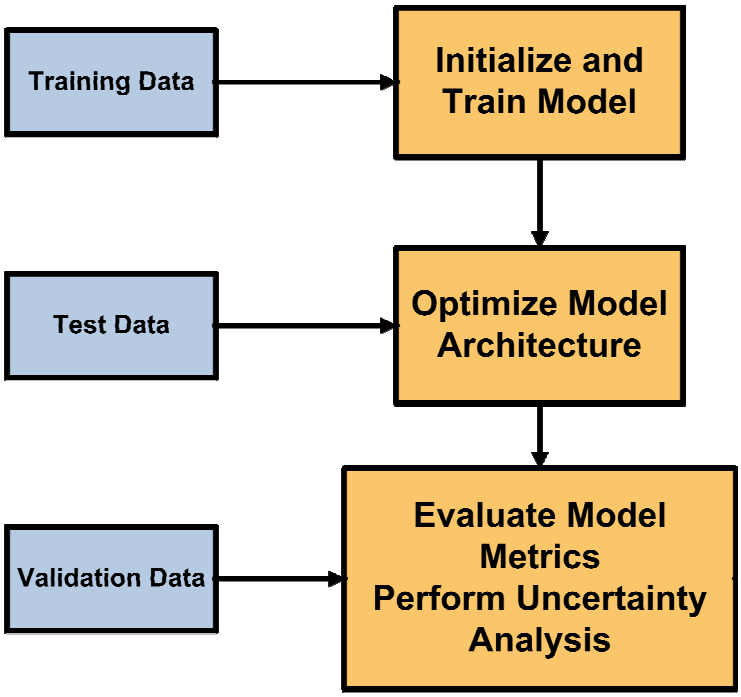


Fig. 2-1. Method for model development and analysis.

The seven case studies are:

1. Faulty Training Data (Chapter 4),
2. Query Data Outside the Training Region (Chapter 5),
3. Effects of Noisy Data (Chapter 6),
4. Effects of Denoising (Chapter 7),
5. Effects of Different Vector Selection Methods (Chapter 8),
6. Effects of the Number of Memory Vectors (Chapter 9), and
7. Effects of Distance Calculation Method (Chapter 10).

Six of the seven case studies can be evaluated using a similar model development procedure. The first two case studies, evaluating data outside the training region and training with faulty data, involve only manipulation of the validation and training data sets, respectively. The third case study, effects of noisy data, also involves only manipulation of the data sets, namely adding additional Gaussian noise to the data. The fourth case study, investigating the effects of data denoising methods, involves an additional data preprocessing step: denoising the data sets with different methods. Case studies 5, 6, and 7 effects of different vector selection methods, training with different amounts of memory vectors, and using a robust distance measure, can be performed with only small changes in the first step above: model initialization and training. In each of these cases, models are developed under various conditions to test the effects on various performance parameters.

3. BASELINE MODELS

This section presents the results of correctly developed or "ideal" AAKR models for both Pressurizer-Level data and RPS loop A data. The data and model architecture used to develop and test these models meet the assumptions and requirements discussed in the previous NUREG volumes. These assumptions include having a correctly specified OLM model, which was developed with error-free data and only evaluates data that lies within the training range.

3.1 Baseline Pressurizer-Level Model Results

The Pressurizer-Level model includes three redundant sensors. The results of this model are shown in Table 3-1. It is important to note that the accuracy metric, which is characterized by the mean squared error of the model, is less than 1% of the sensor span. This small error value indicates that the model's predictions match the target value well. Because the data used to train the model are known to be error-free, this accuracy metric value shows that the model is correctly specified and is well suited for this application. The two detectabilities indicate that EULM and SPRT drift tests can detect a sensor drift or anomaly much less than 1%. In addition, the residual coverage of the model with both analytic and Monte Carlo uncertainty estimates is above the nominal 0.95 level, giving additional evidence that the uncertainty estimates are correct and are conservative for the test data.

Table 3-1. "Ideal" Pressurizer-Level model results

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Model			
Data cleaning	No	No	No
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.055	0.0377	0.102
Autosensitivity	0.466	0.712	0.434
Cross-sensitivity	0.301	0.274	0.355
EULM detectability (% span)	0.333	0.401	0.582
SPRT detectability (% span)	0.00186	0.00185	0.00176
Uncertainty			
Analytic (% of span)	0.107	0.071	0.203
Coverage	0.993	0.953	0.995
Monte Carlo (% of span)	0.121	0.092	0.209
Coverage	0.997	0.959	0.998

3.2 Baseline RPS Loop A Model Results

The RPS Loop A model includes nine sensors, in four redundant sets. The results of this model are shown in Table 3-2. It is important to note that the accuracy metric, which is characterized by the mean squared error of the model, is less than 1% of the sensor span with the largest being due to high noise levels in signal 5. The EULM detectability values show that a drift in the range of 0.1% to 3.5% can be detected, depending on the signal. The larger EULM detectabilities are due to large auto-sensitivity values. The SPRT detectability values indicate that anomalies on the order of 0.003% can be detected. Finally, the residual coverage of the model with both analytic and Monte Carlo uncertainty estimates is well above the nominal 0.95 level, again showing the uncertainty prediction methods are conservative for this data set.

Table 3-2. "Ideal" RPS Loop A model results

RPS Loop A									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Model									
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.261	0.246	0.290	0.229	0.919	0.120	0.027	0.037	0.046
Autosensitivity	0.420	0.451	0.620	0.714	0.160	0.082	0.405	0.370	0.433
Cross-sensitivity	0.120	0.119	0.098	0.116	0.079	0.046	0.130	0.125	0.150
EULM detectability (%)	1.620	1.660	2.680	3.580	2.430	0.278	0.089	0.131	0.178
SPRT detectability (%)	0.0031	0.0031	0.0035	0.0034	0.0018	0.0019	0.0005	0.0006	0.0006
Uncertainty									
Analytic (% of span)	0.840	0.818	0.897	0.899	1.840	0.235	0.043	0.067	0.083
Coverage	1.000	1.000	1.000	1.000	1.000	0.987	0.975	0.996	0.991
Monte Carlo (% of span)	0.550	0.520	0.554	0.544	1.860	0.299	0.075	0.095	0.104
Coverage	0.999	0.999	0.997	0.999	1.000	0.997	0.997	0.999	0.998

3.3 Conclusions

This chapter has presented the results of the "ideal" AAKR models. These models were developed with error-free data and only applied to data which fell inside the training region. The selection of how many memory vectors to include, the kernel width, and the vector selection method was also optimized. For these baseline cases, it can be seen that the Monte Carlo uncertainty values closely match the analytic uncertainty and that both uncertainty calculation methods result in good coverage. The detectability metrics of the models also show that the models are reliable and capable of detecting small drifts. However, the largest detectability metrics are for the feedwater and steam flow sensors. This result is expected because flow sensors generally have high noise levels. Finally, the low accuracy metric values for both models indicate that the models provide good predictions of the unfaulted test data. Because

several of the EULM detectability metrics are above a nominal 1%, these signals may not be good candidates for OLM-based sensor calibration extension. Additional research, analysis, and/or optimization of the models may be necessary if the OLM techniques are to be used for calibration extension. In many cases, the drift assumption used in the set point analysis is near 1%, and this is a general goal for EULM detectability.

Page intentionally blank

4. FAULTY TRAINING DATA

4.1 Introduction

The inclusion of bad data points in a training set can invalidate a model. There are many types of common data errors in most nuclear power plant sensor data (EPRI 2000). Random data errors, missing data, stuck data, outliers, sensor drifts, loss of significant figures, and interpolation errors are just some of the data errors that have been observed in stored historical nuclear plant data that can affect an OLM system. Generally, these errors are simply an artifact of the data historian. They are not reflective of an error in a sensor. Interpolation errors occur when the data are taken from compression routines normally implemented in data archival programs. For example, the PI Data Historian from OSI Software creates a data archive that is a time-series database. However, typically all of the data are not stored at each collection time. Only data values that have changed by more than a given tolerance are stored along with their time stamp. This method requires much less storage but results in a loss of data fidelity. When data are extracted from the historian, the data values between the logged data points are calculated through a simple linear interpolation. For these reasons, data collected for model training should be actual measured data values and tolerances should be set to zero. This may also help in preventing loss of significant figures in training data.

The effects of three types of errors found in training data will be investigated here: outliers in data, stuck data, and data with a sensor drift. These three types of data faults were simulated in the training data sets previously described.

Random outliers were introduced into the training data set by adding a value up to half the mean sensor value to 30 data points (a mere 0.1% of the training set). The training data with outliers is shown in Fig. 4-1 for both the Pressurizer-Level data set and the RPS Loop A data set. Clearly, many significant outliers, which could easily be identified via visual inspection of the data, exist in both these groups.

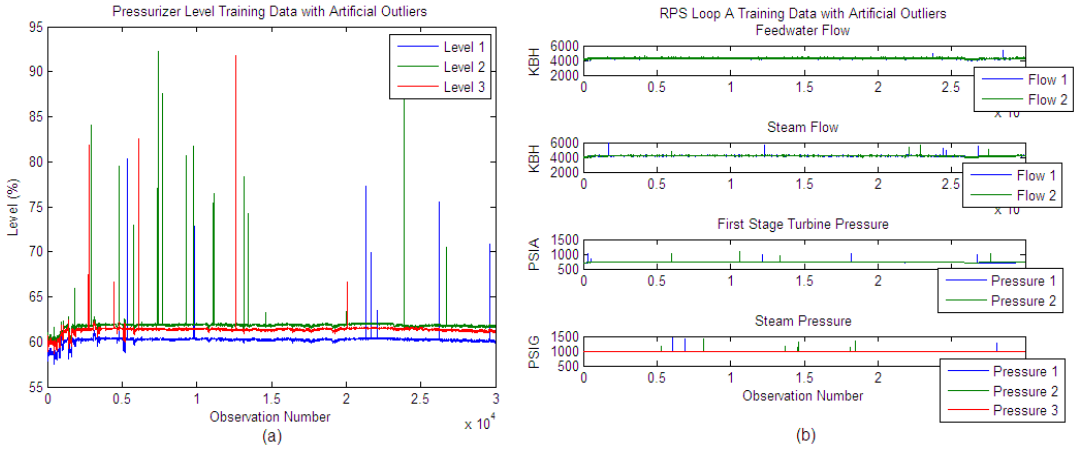


Fig. 4-1. Training data contaminated with outliers for (a) Pressurizer-Level and (b) RPS Loop A models.

Stuck data sensors were simulated in each data set in three different ways. First, a sensor was forced to take on its mean value for half the length of the training data; second, a sensor was forced to take its maximum value for half the length of the training data; and finally, a sensor was forced to take its minimum value for half the length of the training data. The training data for each of these cases is shown in Fig. 4-2 for both the Pressurizer-Level data set and the RPS Loop A data set. The Pressurizer-Level data signal 3 was forced to be stuck; the RPS Loop A data signal 7, the first steam pressure sensor, was forced to be stuck. Again, this type of fault could easily be identified through a visual inspection of the training data.

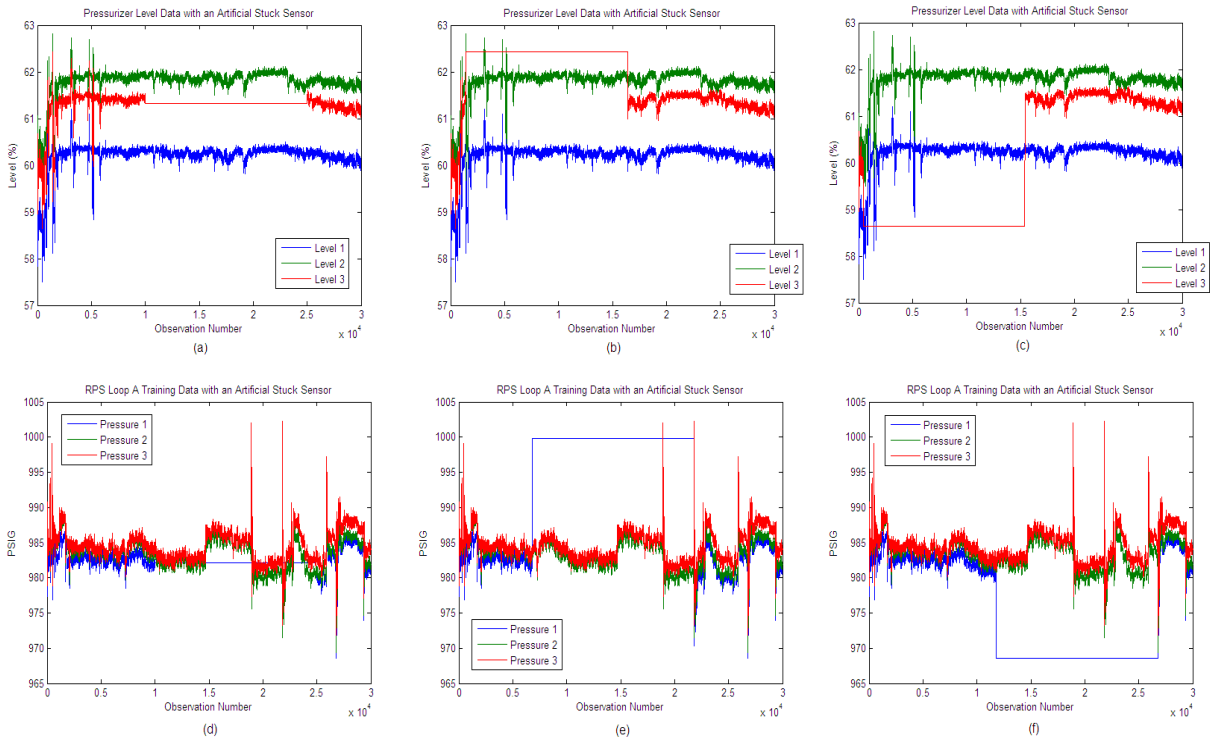


Fig. 4-2. Pressurizer-Level and RPS Loop A training data with a stuck sensor at (a,d) the mean value, (b,e) the maximum value, and (c,f) the minimum value.

The final type of training data fault investigated was sensor drift. An artificial linearly increasing sensor drift ending at a magnitude of 3σ (three times the standard deviation of the sensor signal) was introduced beginning at the time equal to one-third of the length of the training set and reaching its maximum value at the end. The training data for each of these cases is shown in Fig. 4-3. Sensor drifts of this size can easily be identified via visual inspection, if a valid prediction of the correct value exists, smaller drifts may not be as obvious.

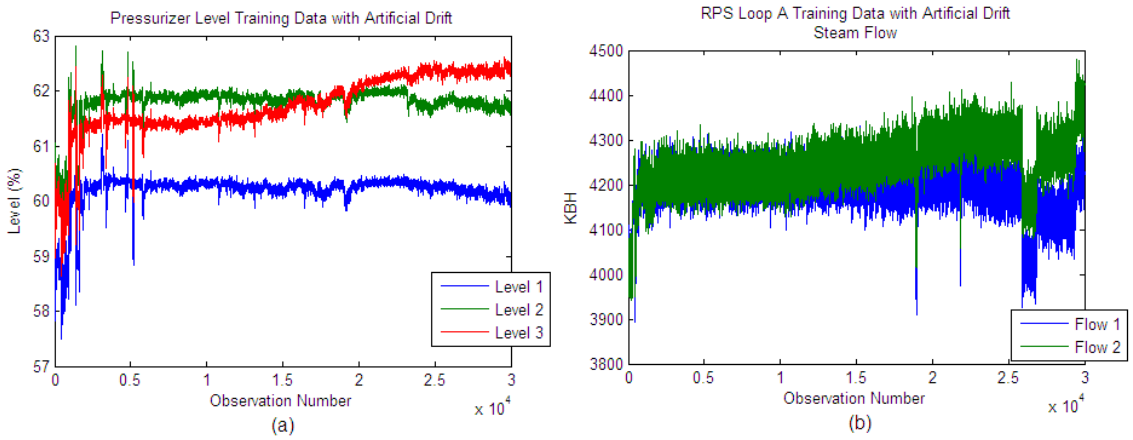


Fig. 4-3. Artificially drifted training data for (a) Pressurizer-Level and (b) RPS Loop A models.

Models were developed using each of the training data sets described above, with the test and validation data sets described in Chapter 3. The model results with each of these types of training data faults are summarized in the following section.

4.2 Results

Models were developed to investigate the model performance effects of three types of training data faults described above: outliers, stuck sensors, and drifted sensors. The results of these case studies, including possible indicators of each fault, are summarized below.

4.2.1 Outliers in the training data

Two models were developed using training data with 0.1% of the training data being outliers. The complete results of these models are in Appendix A.2; the important points are summarized here. The model analysis includes several indicators that the model has not been adequately developed. Three possible indicators are model uncertainty, residual coverage, and sensor signal correlations.

An important requirement of an OLM system is to minimize model uncertainty while maintaining an adequate coverage at or above the nominal 0.95 level. Often, a trade-off occurs between these two metrics. Adequate model coverage can be ensured by an accurate or conservative estimate of model uncertainty. However, because model uncertainty is a factor in determining the detectable drift limit, an optimally useful model would have a low uncertainty; unnecessary conservatism limits the model's applicability. As can be seen in Table 4-1 for the Pressurizer-Level model, although the residual coverage based on the analytic uncertainty is very high, the analytic uncertainty is much larger than that for the baseline model described earlier. In contrast, the Monte Carlo uncertainty estimate is small enough to give acceptable allowable drift limits; the coverage, however, does not meet the 95% requirements. Clearly, uncertainty and coverage can be used to identify an invalid model in this case, whether using the analytic or Monte Carlo uncertainty estimate.

The results for the RPS Loop A model, shown in Table 4-2, are not as clear. In this case, the analytic uncertainty for signals 7–9 is higher than that for the baseline model, and the Monte Carlo uncertainty estimates are similar to the baseline model, and all methods give good residual coverage. The coverage values show the estimates are valid and conservative. These results show that models with more inputs are more robust to outliers in the training data. This is expected because the additional inputs give additional redundant information that can be used to correct errors. This shows that uncertainty and coverage are not always good indicators of outliers in the data.

Table 4-1. Uncertainty and coverage for Pressurizer-Level model trained with outliers

Pressurizer-Level			
Uncertainty	**Signal 1**	**Signal 2**	**Signal 3**
Analytic (% of span)	0.367	0.715	0.442
Coverage	1.000	1.000	1.000
Monte Carlo (% of span)	0.068	0.072	0.068
Coverage	0.749	0.952	0.258

Table 4-2. Uncertainty and coverage for RPS Loop A model trained with outliers

RPS Loop A									
Uncertainty	**Signal 1**	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Analytic (% of span)	0.784	0.733	0.957	0.891	1.780	0.628	0.514	0.510	0.052
Coverage	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.977
Monte Carlo (% of span)	0.474	0.447	0.515	0.535	1.680	0.225	0.097	0.100	0.086
Coverage	0.999	1.000	0.997	1.000	1.000	0.973	0.997	0.985	0.999

Sensor correlation can also be used as an indicator of poor model training data when at least some *a priori* knowledge of the sensor interactions is available. As was noted in the data description section, the sensors in the Pressurizer-Level data set and each of the redundant sensor sets in the RPS Loop A data set have strong (> ~0.7) correlation coefficients. However, outliers in the training data sets degrade the correlation coefficients. In fact, as can be seen in Tables 4-3 and 4-4, the correlation coefficients are significantly degraded, despite the fact that each set is of redundant sensors. This suggests that correlation coefficients may be a valid indicator of corrupt data sets.

Table 4-3. Correlation coefficients for Pressurizer-Level training data with artificial outliers

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Correlation coefficients			
Signal 1	1.000	0.504	0.639
Signal 2	0.504	1.000	0.471
Signal 3	0.639	0.471	1.000

Table 4-4. Correlation coefficients for RPS Loop A training data with artificial outliers

RPS Loop A									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Correlation coefficients									
Signal 1	1.000	0.959	0.669	0.684	0.742	0.741	–0.096	–0.080	–0.269
Signal 2	0.959	1.000	0.677	0.690	0.753	0.747	–0.091	–0.070	–0.264
Signal 3	0.669	0.677	1.000	0.627	0.693	0.688	–0.122	–0.100	–0.314
Signal 4	0.684	0.690	0.627	1.000	0.708	0.701	–0.123	–0.107	–0.362
Signal 5	0.742	0.753	0.693	0.708	1.000	0.833	–0.103	–0.080	–0.278
Signal 6	0.741	0.747	0.688	0.701	0.833	1.000	–0.106	–0.089	–0.272
Signal 7	–0.096	–0.091	–0.122	–0.123	–0.103	–0.106	1.000	0.213	0.445
Signal 8	–0.080	–0.070	–0.100	–0.107	–0.080	–0.089	0.213	1.000	0.447
Signal 9	–0.269	–0.264	–0.314	–0.362	–0.278	–0.272	0.445	0.447	1.000

Two models were developed using training data containing outliers. Three model metrics could indicate corrupted training data: model uncertainty, residual coverage, and training data correlation coefficients. The use of "bootstrap" correlation coefficient probability distributions can be used as a tool to identify outliers. However, none of these metrics are a perfect indicator of outliers in the training data. Visual inspection of the training data is imperative to ensure that outliers are not included.

4.2.2 Stuck sensor in the training data

Six models were developed using training data with a stuck sensor: two with a sensor stuck at the mean value, two at the maximum value, and two at the minimum value. The complete results of all of these models are in Appendix A.2. Sensor signal correlation coefficients are the only clear and consistent automated indicator that a model may suffer from a stuck sensor in the training data set. However, automated data cleaning routines that specifically look for data values that do not change have been developed and can detect this fault type.

Of the three stuck sensor values investigated, a sensor stuck at the mean value had the least effect on model development. Because only "high-power" data are considered, the sensor signals can be fairly steady. This is true for the Pressurizer-Level data, as can be seen in Fig. 4-2(a), above; in fact, a sensor stuck at the mean value is a reasonable approximation to the actual signal. The RPS Loop A data, however, are more variable; a sensor stuck at the mean value in this data set is not a good approximation of the actual signal. Tables 4-5 and 4-6 show the correlation coefficients for the Pressurizer-Level and RPS Loop A data sets with stuck sensors Signal 3 and Signal 7, respectively. The Pressurizer-Level data show insignificant degradation in the correlation coefficients, mostly because the data have little variation. The RPS Loop A data, however, have an appreciable decrease in correlation.

Table 4-5. Correlation coefficients for Pressurizer-Level training data with signal 3 stuck at mean value

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Correlation coefficients			
Signal 1	1.000	0.987	0.974
Signal 2	0.987	1.000	0.960
Signal 3	0.974	0.960	1.000

Table 4-6. Correlation coefficients for RPS Loop A training data with signal 7 stuck at mean value

RPS Loop A									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.287	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.276	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.318	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.317	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.376	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.386	–0.216	–0.297
Signal 7	–0.287	–0.276	–0.318	–0.317	–0.376	–0.386	1.000	0.650	0.684
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.650	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.684	0.947	1.000

For the more drastic types of stuck sensor, a sensor stuck at the maximum value or at the minimum value, correlation coefficients become a much more reliable indicator of faulty data. As was seen in Fig. 4-2, either of these stuck sensor values results in a signal very different from the actual sensor signal. Tables 4-7 and 4-8 show the correlation coefficients for Pressurizer-Level and RPS Loop A data, respectively, with a sensor stuck at the maximum value. Significant degradation was seen for the correlations in both data sets as a result of these stuck sensors. Tables 4-9 and 4-10 show the correlation coefficients for a sensor stuck at the minimum value. Similar decreases in correlation coefficients are caused by this data fault.

Table 4-7. Correlation coefficients for Pressurizer-Level training data with signal 3 stuck at maximum value

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Correlation coefficients			
Signal 1	1.000	0.987	0.555
Signal 2	0.987	1.000	0.539
Signal 3	0.555	0.539	1.000

Table 4-8. Correlation coefficients for RPS Loop A training data with signal 7 stuck at maximum value

RPS Loop A									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	0.291	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	0.281	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	0.292	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	0.293	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	0.242	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	0.210	–0.216	–0.297
Signal 7	0.291	0.281	0.292	0.293	0.242	0.210	1.000	–0.022	–0.116
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	–0.022	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	–0.116	0.947	1.000

Table 4-9. Correlation coefficients for Pressurizer-Level training data with signal 3 stuck at the minimum value

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Correlation coefficients			
Signal 1	1.000	0.987	0.075
Signal 2	0.987	1.000	0.098
Signal 3	0.075	0.098	1.000

Table 4-10. Correlation coefficients for RPS Loop A training data with signal 7 stuck at the minimum value

RPS Loop A									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	-0.083	-0.183	-0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	-0.052	-0.161	-0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	-0.081	-0.239	-0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	-0.086	-0.242	-0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	-0.127	-0.195	-0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	-0.164	-0.216	-0.297
Signal 7	-0.083	-0.052	-0.081	-0.086	-0.127	-0.164	1.000	0.411	0.319
Signal 8	-0.183	-0.161	-0.239	-0.242	-0.195	-0.216	0.411	1.000	0.947
Signal 9	-0.273	-0.264	-0.328	-0.372	-0.302	-0.297	0.319	0.947	1.000

Training data correlation coefficients offer a possible indication of stuck data sensors. As was seen with the Pressurizer-Level data with a sensor stuck at the mean value, correlation coefficients cannot identify a stuck sensor that is a reasonable approximation of the actual signal that has little variation. However, as was seen with the other cases, if the stuck sensor signal is significantly different from the actual sensor signal, correlation coefficients can be a valuable tool for identifying stuck sensor data faults. Again, visual inspection of the training data should be employed to identify stuck sensor faults.

4.2.3 Drifted sensor in the training data

Two models were developed using training data with an artificial sensor drift. The Pressurizer-Level model contains a drift in the third sensor. The RPS Loop A model contains a drift in the fourth signal, the second steam flow sensor. Again, only one clear indicator of this fault is present in the model analysis: signal correlation coefficients.

Because each signal has at least one redundancy, *a priori* knowledge is available about the correlation coefficients of the sets of redundant signals; that is, these correlations should be high. Sensor drift in any one of these signals will cause a degradation in the correlation coefficients, as can be seen in Tables 4-11 and 4-12. The correlation coefficients in the original Pressurizer-Level data were on the order of 0.99. The drifted Pressurizer-Level sensor, Signal 3, has correlation coefficients on the order of 0.64 with both redundant sensors Signal 1 and Signal 2. The original steam flow sensors, Signal 3 and Signal 4, had a correlation coefficient of 0.68. Even this lower correlation showed significant degradation due to the sensor drift; the correlation of the drifted signal, Signal 4, with its redundant sensor, Signal 3, is 0.27.

Table 4-11. Correlation coefficients for Pressurizer-Level training data with a drifted sensor

Pressurizer-Level			
	Signal 1	**Signal 2**	**Signal 3**
Correlation coefficients			
Signal 1	1.000	0.987	0.637
Signal 2	0.987	1.000	0.638
Signal 3	0.637	0.638	1.000

Table 4-12. Correlation coefficients for RPS Loop A training data with a drifted sensor

RPS Loop A									
	Signal 1	**Signal 2**	**Signal 3**	**Signal 4**	**Signal 5**	**Signal 6**	**Signal 7**	**Signal 8**	**Signal 9**
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.324	0.820	0.815	-0.212	-0.183	-0.273
Signal 2	0.970	1.000	0.716	0.288	0.824	0.815	-0.199	-0.161	-0.264
Signal 3	0.716	0.716	1.000	0.273	0.797	0.788	-0.280	-0.239	-0.328
Signal 4	0.324	0.288	0.273	1.000	0.316	0.404	-0.266	-0.338	-0.206
Signal 5	0.820	0.824	0.797	0.316	1.000	0.990	-0.244	-0.195	-0.302
Signal 6	0.815	0.815	0.788	0.404	0.990	1.000	-0.253	-0.216	-0.297
Signal 7	-0.212	-0.199	-0.280	-0.266	-0.244	-0.253	1.000	0.970	0.956
Signal 8	-0.183	-0.161	-0.239	-0.338	-0.195	-0.216	0.970	1.000	0.947
Signal 9	-0.273	-0.264	-0.328	-0.206	-0.302	-0.297	0.956	0.947	1.000

Signal correlation coefficients are a possible indicator of sensor drifts in the training data. This, however, is only valid when groups of redundant sensors are included or when some other *a priori* knowledge of signal correlations is available. Correlation coefficients may not be effective in identifying smaller sensor drifts. Visual inspection of the training data can identify most drifts when redundant, fault-free signals are available.

4.3 Discussion and Recommendations

A major assumption in the use of empirical model based OLM is that the training data is error free. The above results showed the importance of carefully reviewing any acquired data that is going to be used in model development to assure its quality. Visual inspection of the data or focused automated techniques would identify each of the three types of faults investigated here. It is important that any nuclear plant using OLM be aware of the threat posed by bad data and have a designated person, in addition to automated techniques, to visually inspect any data that will be used for development of their OLM system. If data faults are identified in the training data set, some action must be taken before model development. If data anomalies such as outliers, interpolation errors, random data errors, missing data, or loss of significant figures are identified, the data should be processed to remove these faults. For this purpose, automated data cleanup utilities can be used to help remove these types of errors, by either removing bad data or replacing it with the most probable data value using some algorithm, such as linear interpolation. It is most common, however, to delete all bad data observations from the training data set. Most OLM software systems include automated tools for data cleanup; these tools easily identify outlying data but may be insensitive to data errors that occur within the expected region of operation, such as small

data drifts and stuck sensors if not specifically designed to detect them. If visual inspection of the data indicates errors that cannot be corrected by a data cleaning utility, a new training data set should be identified and evaluated for model development.

5. QUERY DATA OUTSIDE THE TRAINING REGION

5.1 Introduction

The next limiting case examined occurs when an OLM model attempts to predict values for data that do not fall within the training range. Almost all of the models currently being used for OLM are developed using training data. Even the nonparametric models are considered to use "training data," although they require no training. In such cases, "training data" refers to the data stored in memory which is recalled and processed each time a query is made. The predictions of the "true" sensor value from autoassociative kernel regression (AAKR) are in this way dependent upon the training data used to develop the model.

A basic assumption of nonredundant models is that the model is operating within its training range. In other words, it is always assumed that the data being predicted come from the same operating state as the training data. Unfortunately, several circumstances, including seasonal variations, equipment repair or failure, and system-operating changes can cause a change in operating conditions. When the operating conditions change, the training data and model may no longer be representative of the true operating state of the plant. If the training data are no longer valid, the model must be retrained, or in the case of most nonparametric models, the memory matrix must either be appended or replaced with new data that accurately represent the plant's current operating state. If the model is operating outside of its training range, no confidence can be given to the model's prediction, and the uncertainty estimates should increase to indicate that this fault is occurring; in addition, a reliability assessment module should be used. The term "reliability assessment module" was introduced by the Halden group and has become an industry standard. A reliability assessment module is a method that produces a metric related to the reliability of a prediction. This metric is usually partially dependent on whether the query is within the training data range.

This section examines and quantifies how operating outside the training region will affect three model metrics: accuracy, uncertainty, and detectability. The accuracy metric is characterized by the mean squared error of the model predictions. The objective in model development and evaluation is to keep this value small, indicating a small error. When models are evaluated on query data outside the training space, analytic uncertainty estimates should increase beyond what is acceptable for an OLM system. Monte Carlo uncertainty estimates are not considered an indicator for this type of data problem because those estimates are made prior to model implementation and are an average value not dependent on the query. Conversely, analytic uncertainty can be estimated for each model prediction. Finally, model detectability, the model's ability to correctly identify sensor drift, should be impacted by query data outside the training range. In theory, the accuracy metric value, analytic uncertainty, and detectability should inflate to a point at which the model can no longer be used for OLM when the query data are not contained within the training region.

Data can be considered to be outside the training region for two reasons. First and easiest to identify, data may lie outside the maximum and minimum range of the training data. Second, data collected during different operating modes may have different relationships among the non-redundant sensors. Because the models developed here are applicable only to "high-power" data, data that have been collected in lower power regions were identified as data outside the training region. Figures 5-1 and 5-2 show the training and query data for the Pressurizer-Level model and RPS Loop A model, respectively. Clearly, the query data sets contain regions that are well outside the training region. Models will be developed using the same training data and test data described in Chapter 3. These models will be evaluated using the query data shown below to assess the effect of such data on model performance.

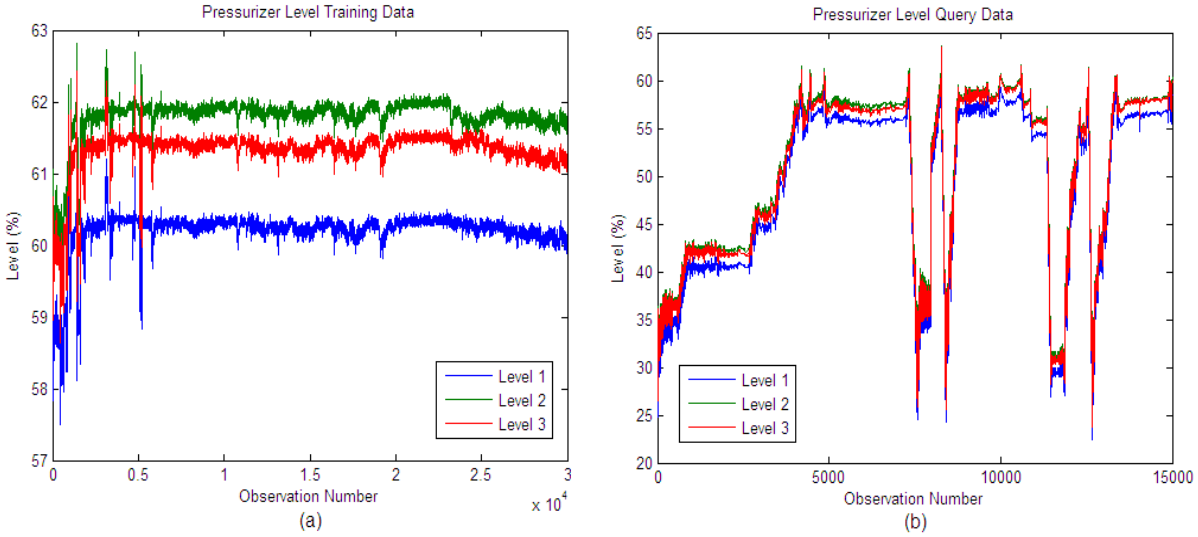


Fig. 5-1. (a) Training data and (b) query data for the Pressurizer-Level model.

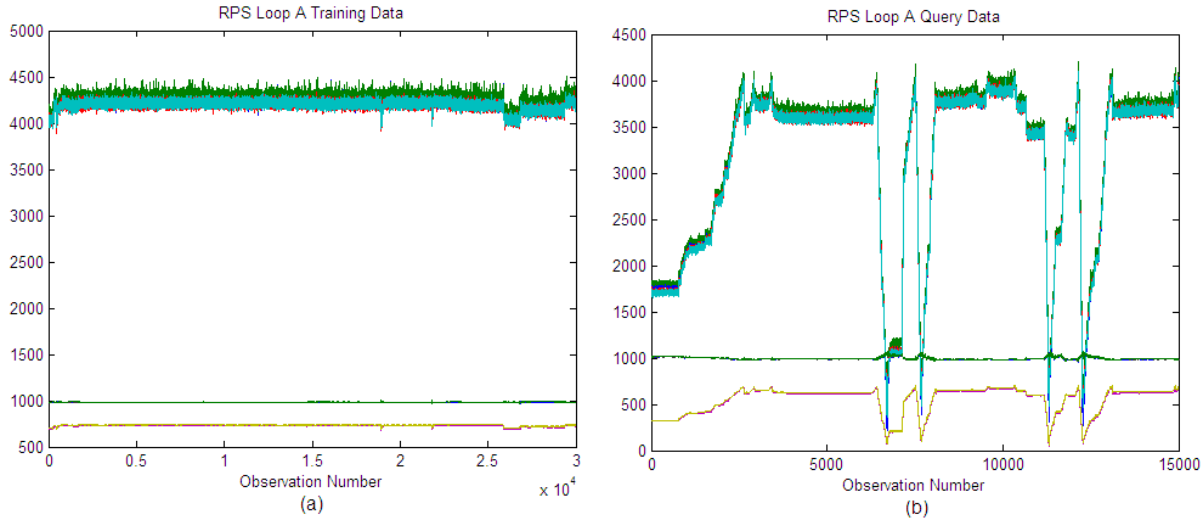


Fig. 5-2. (a) Training data and (b) query data for the RPS Loop A model.

5.2 Results

Models were developed using the training and test data sets described in Chapter 3. They were then evaluated using the query data sets shown above. As was expected, the three model metrics—accuracy, uncertainty, and detectability—showed significant degradation over the baseline models described. A comparison of each of these metrics for the baseline model and the current model are shown in Tables 5-1 and 5-2 for the Pressurizer-Level model and the RPS Loop A model, respectively. Full model results are included in Appendix A.3.

Table 5-1. Pressurizer-Level model performance with query data outside the training region

		Pressurizer Level			
		Signal 1	Signal 2	Signal 3	Average
Accuracy	Inside Training Region	0.055	0.038	0.102	0.065
	Outside Training Region	12.400	12.200	12.200	12.267
Analytic Uncertainty	Inside Training Region	0.107	0.071	0.203	0.127
	Outside Training Region	24.700	24.500	24.400	24.533
Detectability	Inside Training Region	0.333	0.401	0.582	0.439
	Outside Training Region	50.200	48.200	48.700	49.033

Table 5-2. RPS Loop A Model performance with query data outside the training region

		RPS Loop A									
		Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9	Average
Accuracy	Inside Training Region	0.261	0.246	0.290	0.229	0.919	0.120	0.027	0.037	0.046	0.242
	Outside Training Region	28.000	27.500	26.900	27.000	27.400	27.300	1.060	1.040	1.050	18.583
Analytic Uncertainty	Inside Training Region	0.840	0.818	0.897	0.899	1.840	0.235	0.043	0.067	0.083	0.636
	Outside Training Region	56.000	55.000	53.800	53.900	54.700	54.500	2.120	2.080	2.100	37.133
Detectability	Inside Training Region	1.620	1.660	2.680	3.580	2.430	0.278	0.089	0.131	0.178	1.405
	Outside Training Region	85.200	82.400	83.300	83.100	81.100	79.200	3.030	2.930	3.050	55.923

The three metrics described above, accuracy, analytic uncertainty, and detectability, all increase significantly for models evaluated with query data outside the training region. Any of these three metrics can be used as indicators that this type of data fault might be occurring. OLM systems should also include a reliability assessment module for evaluating the reliability of model predictions and alerting operators not to trust the OLM results.

5.3 Discussion and Recommendations

This section demonstrated that an AAKR model cannot effectively be used to evaluate query data outside the training region. In fact, no nonlinear models have the ability to correctly extrapolate beyond their training region. Linear models will extrapolate linearly and may be valid if the relationships are strictly linear. When plant operating conditions change, a nonlinear model, using training data from a different operating mode, is no longer effective for predicting the current operating state of the plant. When this occurs, a new model must be developed; however, validated fault-free data may not be readily available. For parametric models, this means model retraining or derivation of a new first-principles model. For most nonparametric models, the training data memory matrix may be either replaced or simply supplemented with new data that are characteristic of the plant's current operating state. Caution should always be used when adapting a model to new conditions. The new data used to adapt the model must be validated as fault-free data.

Three metrics were identified that can indicate that query data has moved outside the training data region: accuracy, analytic uncertainty, and detectability. When these metrics increase beyond acceptable limits for OLM systems, query data should be evaluated to determine if the data are still in the model's training region.

In addition to monitoring these three metrics, OLM systems should include a reliability assessment module to evaluate the reliability of a model prediction. A typical reliability metric would range from 0 to 1, with a metric of zero indicating that the model prediction is not reliable and a metric of 1 indicating

that it is reliable. Reliability assessment should be performed in two steps. First, the assessment should determine whether the query vector is within the training data range. If the vector is not, the prediction reliability should immediately be set to 0.0. However, if the query is within the training range, the reliability metric should be a function of the similarity between the query and memory vectors. Typically, this function relies only on the greatest similarity value between the query vector and the memory vectors; that is, the reliability is related to the distance between a query vector and the closest memory vector. High reliability occurs when the query is very similar to the training data. With this type of construction, a reliability assessment measure can identify when query data do not fall within the training region and OLM operation should not be trusted.

Page intentionally blank

6. EFFECTS OF NOISY DATA

6.1 Introduction

Another important data quality issue is the level of noise present in each sensor. This noise can include any random variable such as process noise, measurement noise, and electronic noise. Process noise is the natural perturbation of a parameter about its true or expected value. This type of noise is particularly evident in flow and level sensors that have natural fluctuations due to "pressure waves" about the intended value. Measurement noise refers to the noise inherent in the sensor. Electronic noise includes the noise resulting from data transfer from the sensor to a central processor. These three noise contributors combine to cause random errors in the data that cannot be deterministically modeled. This case study investigates the effect of increasing noise levels in the data by adding Gaussian noise to the already noisy plant data described earlier. The general expectation is that as the noise level in a sensor set increases, model performance is degraded.

The noise level in each sensor is characterized by the signal-to-noise (SNR) ratio. This ratio is estimated by smoothing the signals with a median filter. The ratio is then given as the variance of the smoothed signals divided by the variance of the noise that was removed. A high signal-to-noise ratio indicates a low level of noise, while a low signal-to-noise ratio indicates a high level of noise. Very complicated methods can be developed to give more exact estimates of the signal noise (such as wavelets filtering), but a simple SNR estimate is sufficient for this analysis.

6.2 Results

As expected, as the level of noise increases, model performance is degraded. Figure 6-1 shows the average EULM detectability of the three PZRL sensors as a function of the average signal-to-noise ratio. Low signal-to-noise ratios (high levels of noise) result in high EULM detectabilities; while low signal-to-noise ratios result in acceptable detectability metrics. For these data, acceptable EULM detectability metrics correspond to signal-to-noise ratios of approximately 3 or greater.

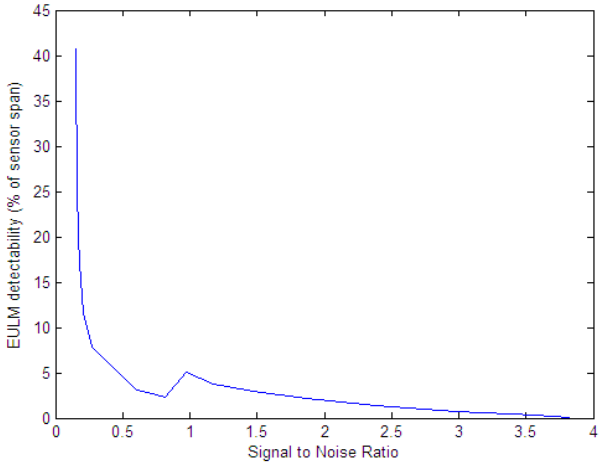


Fig. 6-1. Signal to noise ratio vs EULM detectability for the Pressurizer-Level model.

6.3 Discussion and Recommendations

This type of study can easily be performed with other data sets to determine the level of noise that will not degrade model performance beyond acceptable limits. If high noise levels are unavoidable, the effects on model performance can be reduced by denoising the data prior to model development and implementation. Several methods exist for data denoising; the next case study examines these methods in more detail.

7. EFFECTS OF DENOISING

7.1 Introduction

An important component of OLM uncertainty analysis is signal denoising. A sensor signal is considered to have both a deterministic component and a random component. The deterministic component is the information-carrying part of the signal and contains the process dynamics. The random component is the random part of the signal and is believed to be made up of sources such as the process, measurement, and electronic noise. In general, the deterministic component is considered to be the information-carrying part of the input, and the random component is considered any additional, noninformation part. However, this definition of the random component does not always hold true. As discussed later in this chapter, estimating the noise (or random component) is a complicated, yet critical task in any OLM system, particularly when the noise component adds a significant proportion of variance to the signal such as during steady state operation.

In denoising, a signal is smoothed so that the contaminating noise is removed and only an estimate of the "true" signal remains. Denoising is an important step in estimating the bias. It is also needed to estimate the value of σ_ε^2, which is commonly termed the "irreducible error" because it cannot be controlled through the modeling process and is due to random disturbances in the measurement process. In this study related to nuclear power, noise is assumed to be normally distributed with a mean of 0 and variance of σ_ε^2. Therefore, the only term that describes noise is σ_ε^2. σ_ε^2 is used in calculating the analytic variance, the squared bias, and in the estimate of the total uncertainty when prediction intervals are used. At first, estimating σ_ε^2 may seem rather straightforward, but unfortunately, many problems arise when using actual data to make this determination.

Many previous analyses have simply employed a filtering technique to determine the noise level in a set of data. The filtered signal is subtracted from the unfiltered signal to produce a noise estimate. Although this method is often valid, several factors must be considered. The first is the sampling rate. If the data are not sampled at a rate faster than the process dynamics, then filtering may remove process dynamics that should not be considered to be noise. To elaborate, consider Fig. 7-1, which presents temperature data from the secondary side of the nuclear power plant. In the figure, the units of time are not specified. If the time on the x-axis is in units of seconds, and the plant is in normal steady state operation, then it would not be physically possible for the temperature to change quickly enough to create the large fluctuations exhibited by the data. In this case, low pass filtering could be performed without losing information about the actual process. However, if the time is in days, or perhaps even hours, the process variable might vary at the rate shown in the data. Thus, filtering would not be allowed. Engineering judgment must always be used before applying a standard filtering technique to data since filtering requires foreknowledge about the data and the underlying process. Filtering should, therefore, only be used when there is certainty that the process dynamics will not be lost.

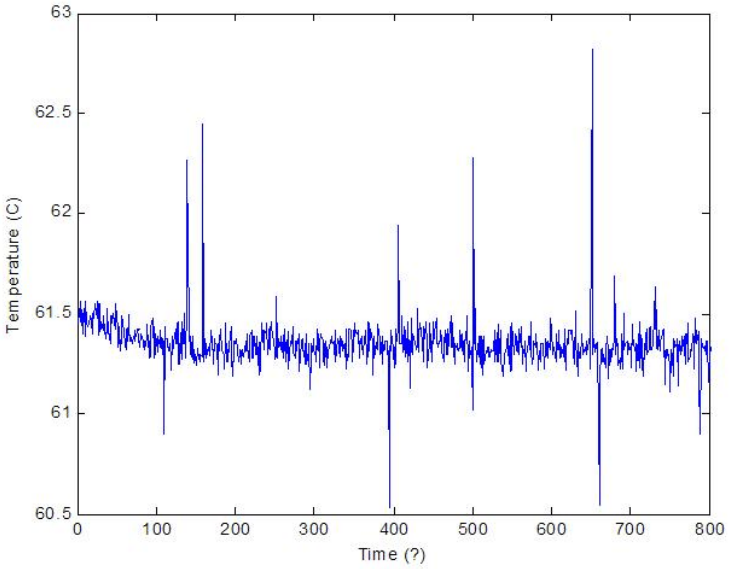


Fig. 7-1. Plot of temperature data.

Another problem with using standard filtering to estimate noise is that it cannot distinguish between common process noise and independent instrument noise. As depicted in Fig. 7-2, redundant sensor groups in nuclear plants contain common as well as independent noise sources. As shown in the figure, each redundant sensor contains a source signal indicative of the actual process, such as a steam generator level (denoted by P in the figure). However, redundant sensors will not generally produce exactly identical readings, even though they are measuring the same process because the sensors are contaminated with noise. In Fig. 7-2, the measured signals are shown as being corrupted with both common and independent noise. Although it is given the term "noise," common noise is actually reflective of the process. Some examples of common noise are water-level fluctuations due to turbulent flow or possibly the high-frequency boiling component of a signal measured by a steam generator level detector, if the detector can respond in that range. Each sensor also contains some unique independent noise. Independent noise is frequently caused by electrical noise contamination of the specific sensor signal. Because common process noise is indicative of a process variable's state, it then is apparent that irreducible error should only include independent noise and not common process noise. The figure shows each sensor as having only a single common and independent noise source. In actuality, however, a sensor can be contaminated by multiple common and independent noise sources. It is this reason that noise is commonly modeled by a Gaussian distribution. The central limit theorem states that as independent noise sources are added, regardless of their distributions, the resulting distribution tends toward Gaussian.

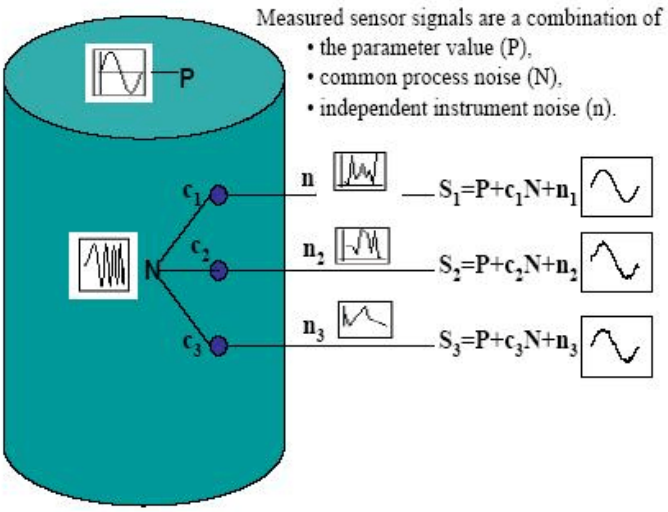


Fig. 7-2. Diagram of real world signals containing both common and independent noise.

When conventional frequency-based filtering is being performed, there is no guarantee that only independent noise is removed. Frequently, common process noise, which should not be filtered, is also removed because it may contain frequency content similar to independent noise. Because there is usually little prior knowledge that is useful in differentiating between common and independent noise, it is difficult to select proper window sizes or filtering criteria. Generally, the small addition of common process noise in the error term is considered negligible. However, in situations where the common process noise is exceedingly large, such as flow or steam generator level sensors, the predictive uncertainty may be inflated to the point where the OLM drift limits are exceeded. Therefore, standard smoothing methods may be unsuited for OLM applications. To deal with this problem, an optimization algorithm may be used to determine the best filtering parameters.

In this study, a genetic algorithm (GA) is used to determine optimal smoothing parameters for several filtering techniques, including median filtering, wavelet denoising, kernel smoothing, and infinite impulse response filtering. GAs are a valuable function optimizer because they are less susceptible to getting "stuck" at local optima than gradient search methods. GAs are inspired by Darwin's theory of evolution. GAs solve optimization problems by an evolutionary process that finds the best, or fittest, solution (survivor) out of many populations.

A GA starts with a set of solutions, represented by chromosomes, called a population. Solutions from one population are allowed to reproduce, either asexually where the entire chromosome remains the same, or two chromosomes can mate, where the two parents trade genes (features) that are to the right of the crossover point to form a whole new chromosome, or mutate and are combined with random chromosomes to form the next generation population. During mutation, small elements of DNA, also referred to as bits in the chromosomes, are changed. It is hoped that the next generation population will better meet the objective function than the previous population. Chromosomes that are allowed to crossover are selected according to their fitness; the more suitable they are, the more chance they have to reproduce and have their features survive to the next generation. This process is repeated until the stopping criterion (i.e., number of iterations or improvement of the best solution) is satisfied. Overall, GAs provide an alternative method to traditional gradient descent techniques for solving optimization problems. Many real world problems involve finding optimal parameters, a process that might prove difficult for traditional methods but is ideal for GAs [Leung 1996]. However, it is important to realize that GAs are simply a search technique, not function optimizers. A GA promises convergence but not optimality, and like all tractable optimization techniques, cannot be guaranteed to find a local minimum

[Chung 1994] for general nonlinear problems. A more complete discussion of GA optimization can be found in Haupt and Haupt [2004].

In this study, the aforementioned filtering methods were applied to different types of data (for both real and simulated redundant nuclear sensors) to evaluate the GA performance for this application and also to see if any method consistently outperformed the others. Redundant signals were used because safety requirements make them common in the nuclear industry and also because they provide more information regarding the true noise statistics. The results obtained with the GA were compared to those obtained using default settings for the MATLAB functions:

1. median filtering: window size of 5,
2. kernel filtering: window size of 10 and bandwidth of 0.5, and
3. wavelet filtering: maximum decomposition level of 10.

The total analytical uncertainty was also computed for each smoothing method to determine what (if any) impact the different denoising techniques would have on the estimation of OLM uncertainty.

To obtain an estimate of the total uncertainty, an autoassociative kernel regression (AAKR) model was developed. To develop the model, the data were first divided into training and test sets using the Venetian blind method in which the data are divided into alternating blocks. The training data were used to develop the model and were also denoised to obtain an estimate of σ_ε^2. The test data were then input to the model and used to obtain an estimate of uncertainty. Although σ_ε^2 was calculated using denoised training data, analysis showed that for the data sets used in this study it was still applicable for the test data. When the test and training data were switched, it was found that the results were not statistically different. The measure of the total uncertainty was obtained by computing the 95% prediction interval for the model using the following equation:

$$95\%PI = t_{n\text{-}p,\alpha/2}\sqrt{\sigma_\varepsilon^2 + Var\left(\hat{x}\right) + \left[Bias\left(\hat{x}\right)\right]^2}\,, \tag{7.1}$$

where $t_{n-p,\alpha/2}$ is the t-statistic which approximates the normal distribution for n-p degrees freedom and confidence level 1-α, $\mathrm{Var}(\hat{x})$ is the variance of the model predictions $\hat{x}$, and $\left[Bias(\hat{x})\right]^2$ is the squared bias (the model's systematic error), which is computed using the direct calculation method [NUREG/CR-6895 Volume 2].

The fitness function attempts to maximize the amount of noise removed from each signal while minimizing the correlations between the removed noise, the correlation between the noise and its corresponding denoised signal, and the autocorrelation of the noise. It is obvious that the amount of noise removed should be maximized, but the other criteria require explanation. If the noise removed form each signal were correlated with the noise removed from other signals, it is probable that the correlated components were from common, process sources. If the noise and denoised signals were correlated, it is probable that the noise also contained process dynamics. Lastly, it is assumed that the instrument noise is uncorrelated with itself through time. The fitness function is given by:

$$Fitness = -w_1 \sum_i \sigma^2_{\varepsilon,,i} + w_2 \sum_i ABS\left|corr\left(noise\ i, noise\ j\right)\right| + w_3 \sum_i ABS\left|corr\left(noise\ i, sig\ j\right)\right| + \sum_i ABS\left|autocorr\left(noise\ i\right)\right| \quad . \quad (7.2)$$

The weights in the fitness function were set so that all terms were given equal importance. This was done because the magnitude of each term was found to be similar when denoising the data set using the default smoothing parameters. The equation was then normalized and the weights set so that the optimized fitness function equaled zero.

The GA was run using the Matlab genetic algorithm tool. The population size was set based upon the number of filtering parameters being optimized. In general, it was set to a larger value so that the filtering techniques that had tendencies to become unstable would have a greater chance of finding an actual solution with each generation. The number of iterations was determined by running the GA at least once with 100 iterations; if it appeared that the solution had not yet converged, the iterations were increased until the solution seemed stable. For all of the filtering techniques, the GA selected the smoothing parameters for each signal individually, rather than assuming universal smoothing parameters for all signals in a redundant group.

Because some of these denoising techniques were not described in the earlier two volumes of this series, a brief description of each technique is provided in the following sections.

7.1.1 Median filtering

For median filtering, the GA determined the optimal window size. Median filtering is a nonlinear digital filtering technique. A median filter is based upon moving a window over a signal. The values in the window are sorted into numerical order, and the output is selected as the median value within the input window [Terrell 1988].

7.1.2 Wavelet denoising

For wavelet denoising, the GA was used to determine the optimal thresholding selection method, the threshold rescaling method, the threshold type (hard or soft), and the decomposition level. The wavelet denoising technique investigated in this work was developed by Miron [2001] and was initially used to produce artificial data sets that closely resemble raw nuclear plant data. This approach decomposes a signal, *S,* into an approximation, *A*, and set of detail components, *D*. If the decomposition level is *L*, then this approximation may be written as follows:

$$S_L = A_L + \sum_{i=1}^{L} D_i \ . \quad (7.3)$$

The approximation component represents the low-frequency or system behavior portion of data, while the detail components represent the high-frequency or noise portions of the decomposed approximations. As a signal is decomposed, at first only the high-frequency portions are removed. However, as the decomposition level increases, the lower-frequency portions of the signal begin to be filtered along with the noise. Noise is removed by applying different denoising settings, such as wavelet type, thresholding strategy, re-scaling, etc., to the decomposition [Miron 2001]. Only when these criteria are considered optimal by the GA, is it assumed that none of the process dynamic is included in the noise estimate. At this point, denoising is stopped, and the noise-free signal is reconstructed with this set of

optimal coefficients. Volumes 1 and 2 of this report describe the wavelet denoising process in greater detail.

7.1.3 Kernel smoothing

Kernel smoothing smoothes data estimates using a weighted average of points that are within the specified window size for the training set and that are local to the query point. The estimate is calculated as follows:

$$\hat{m}_h(x) = \frac{\sum_{i=1}^{n} [K_h(X_i - x) Y_i]}{\sum_{i=1}^{n} [K_h(X_i - x)]} , \tag{7.4}$$

where

h	is the bandwidth,
K	is the kernel, and
n	is the window size [Wand 1995].

For this application, a Gaussian kernel and fixed window size were used. The kernel bandwidth (h) controls the smoothness or roughness of a density estimate. A larger bandwidth results in a higher bias, while a smaller bandwidth increases the variance. The constant window size keeps the bias constant, but the variance becomes inversely proportional to local density. In kernel smoothing techniques that employ a nearest neighbor or adaptive window size, the variance is kept constant, but the bias varies inversely with local density [Wand 1995]. This case is not investigated here.

7.2 Results

In this section, the results comparing the filtering capabilities of GA-optimized techniques are presented. First, both techniques were used to filter a simulated data set. By using a simulated data set, the true noise statistics were known and used as a baseline in evaluating the performance of each technique.

For the simulated data set, a source, s, was considered that is described by the following equation:

$$s = .8 \times \sin(2\pi\sqrt{t}) + .2 \times t^2 , \tag{7.5}$$

where t is uniformly distributed on the interval [0,3].

Three redundant sensors were simulated by introducing a small bias to the actual signal (s) and then adding both a common and an independent noise source. In actual plant operations, common noise could result from the high-frequency boiling of a steam generator, while independent noise could be caused by the electrical contamination of an individual sensor. Here the common process noise, $N_c(0,0.1)$, is modeled by drawing random numbers from the Gaussian distribution with a mean 0 and variance 0.1. The independent noise, $L_i(0,v_i)$ is drawn from the Laplacian distribution with the variable-specific variance v_i. These noise sources are added to each channel with differing weights to model a

contaminated signal measured by a process sensor. Also, by varying the variance of the noise, we can determine if any of the techniques performances change with the different noise levels. Thus, the three simulated redundant channels are constructed with the following equations:

$$X = \begin{bmatrix} .8\times\sin(2\pi\sqrt{t}) + .2\times t^2 + N_c(0,0.1) + L_i(0,0.2) + 1.5 \\ .8\times\sin(2\pi\sqrt{t}) + .2\times t^2 + N_c(0,0.1) + L_i(0,0.04) + 0.0 \\ .8\times\sin(2\pi\sqrt{t}) + .2\times t^2 + N_c(0,0.1) + L_i(0,0.4) - 1.3 \end{bmatrix}. \quad (7.6)$$

The simulated signals are shown in Fig. 7-3; the "true" signals include the base signal and the common, process noise.

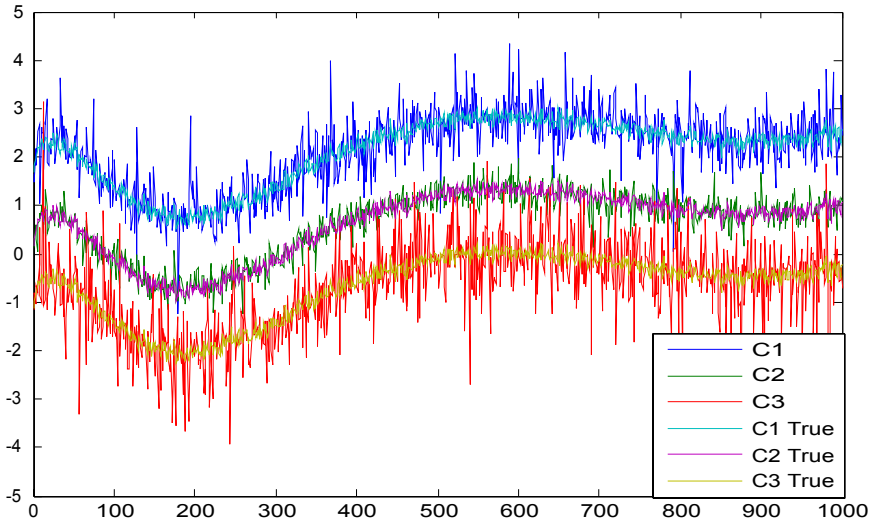


Fig. 7-3. Simulated signals.

The parameters used to evaluate denoising performance were

- the average autocorrelation of the noise estimates,
- the average absolute correlation of each noise estimate with the other noise estimates,
- the expected correlation of each noise estimate with its smoothed signal,
- the fractional variance reduction, and
- the variance of the estimated noise.

The first three parameters (the autocorrelation of the noise, the noise estimates' correlations with themselves, and the smoothed variables) measure the method's ability to remove noise that is not correlated with other noise estimates or with the process itself. The fractional variance reduction measures the extent of the smoothing. The final parameter (noise variance) shows the accuracy of each method's ability to estimate the true amount of noise in the signal.

Table 7-1 reports the results for the median filtering techniques. The results indicate that GA optimization did improve the filtering. The correlation parameters appeared not to have changed significantly between the techniques. However, the GA-selected window sizes allowed a greater fractional variance reduction, and the estimated noise variance was closer to the true variance. Although the GA did improve the median filtering techniques, the improvement was not all that substantial. In this case, the default window size of 5 appeared to provide adequate smoothing. However, the default settings cannot always be trusted, and this GA-based optimization provides a method for ensuring that the smoothing parameters are correctly set for whichever data set is at hand.

Table 7-1. Comparison of median filtering results

	Noise correlation w/itself	Average noise correlation w/other noise	Noise correlation w/denoised sig	Fractional variance reduction	True noise variance	Estimated noise variance
Default parameter = 5						
Channel 1	0.0195	0.0292	0.0611	0.2365	0.1743	0.1579
Channel 2	0.0196	0.0408	0.0636	0.0704	0.038	0.0438
Channel 3	0.02	0.0469	0.0774	0.3929	0.386	0.3718
Genetic algorithm parameters = [7, 4, 10]						
Channel 1	0.0196	0.0294	–0.0442	0.2448	0.1743	0.1719
Channel 2	0.019	0.0494	–0.0159	0.0752	0.038	0.0399
Channel 3	0.0188	0.0552	–0.0133	0.4393	0.386	0.3813

Figures 7-4, 7-5, and 7-6 also display plots comparing the GA optimization to the default settings for all the techniques. Examination of the graphs for wavelet denoising (Fig. 7-6) shows that high-frequency artifacts remain even when a GA is used to select the parameters. Although these artifacts do not appear to affect the estimation of σ^2_ε, their presence is still notable and may dissuade the techniques use in OLM applications.

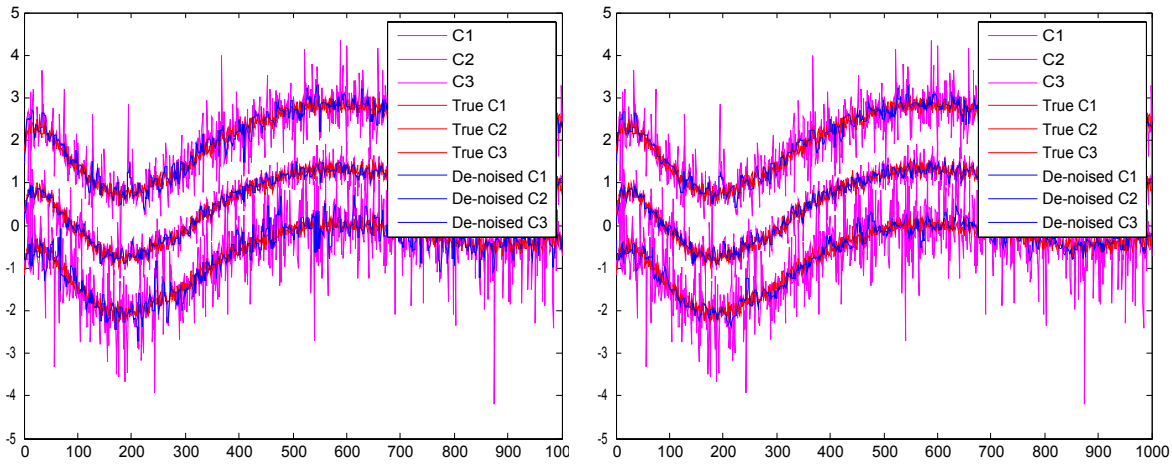
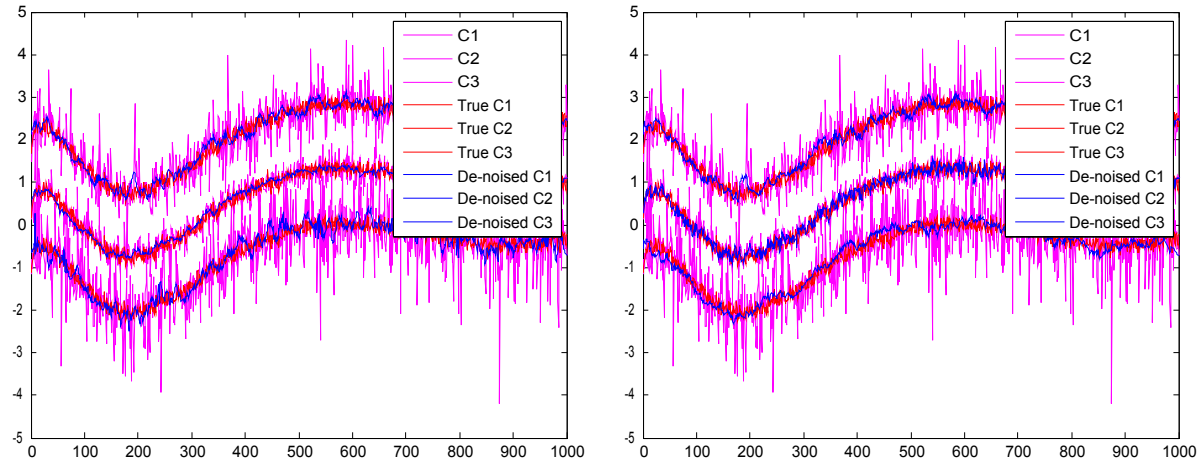
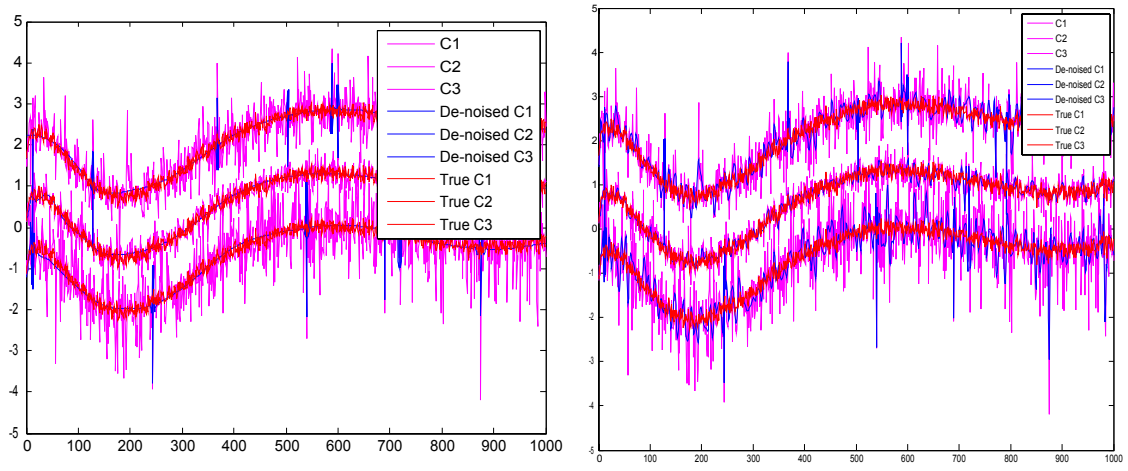


Fig. 7-4. Median filtering results on the simulated data set for (a) GA-optimized window sizes and (b) default window sizes.



(a) (b)

Fig. 7-5. Kernel smoothing results on the simulated data set for (a) GA-optimized bandwidths and window sizes and (b) default bandwidths and window sizes.



(a) (b)

Fig. 7-6. Wavelet denoising results on the simulated data set for (a) GA-optimized smoothing parameters and (b) default smoothing parameters.

Tables 7-2 and 7-3 show the results for the kernel smoothing and wavelet denoising, respectively. The tables show that the amount of noise removed using the GA technique proved to be closer to ideal. For this simulated study, kernel smoothing showed the best performance.

Table 7-2. Comparison of kernel smoothing results for simulated data set

	Noise correlation w/itself	Average noise correlation w/other noise	Noise correlation w/denoised sig	Fractional variance reduction	True noise variance	Estimated noise variance
Default parameter = 9						
Channel 1	–0.0031	0.1249	0.0177	0.2791	0.1743	0.1694
Channel 2	–0.0034	0.1309	0.0188	0.0976	0.038	0.0461
Channel 3	–0.0117	0.1066	0.0189	0.4378	0.386	0.3785
Genetic algorithm parameter = [992 15; 630 4; 820 25]						
Channel 1	0.0177	0.1085	–0.0024	0.2842	0.1743	0.1721
Channel 2	0.0221	0.1248	–0.0033	0.0805	0.038	0.0381
Channel 3	0.0179	0.0772	0.0067	0.4605	0.386	0.3822

Table 7-3. Comparison of wavelet denoising results for simulated data set

	Noise auto-correlation	Average noise correlation w/other noise	Noise correlation w/denoised sig	Fractional variance reduction	True noise variance	Estimated noise variance
Default results						
Channel 1	0.0182	0.007	0.1549	0.1972	0.1743	0.1188
Channel 2	0.0178	0.0044	0.1146	0.0795	0.038	0.0369
Channel 3	0.0177	0.0066	0.1092	0.3197	0.386	0.2711
Genetic algorithm results						
Channel 1	0.0217	0.0006	0.1617	0.2792	0.1743	0.1641
Channel 2	0.0206	0.0003	0.0723	0.1082	0.038	0.0489
Channel 3	0.0204	0.0023	0.0999	0.4306	0.386	0.3572

Because there was not a great variation between the GA-optimized and default smoothing methods, it is not surprising that the improvement from the GA had little impact on the overall OLM uncertainty estimate. Table 7-4 reports the computed uncertainty (using analytical variance and directly calculated bias) for the channels in the simulated set. The table shows that the uncertainty using the GA-optimized kernel smoothing technique most closely matches the true uncertainty. However, the other techniques (even those using default parameters) still come within 0.05% of the true value.

Table 7-4. Calculated analytic uncertainty estimates for each denoising technique

	Channel 1	Channel 2	Channel 3
Kernel smoothing w/default smoothing pars	0.2012	0.1063	0.3096
Kernel smoothing w/GA smoothing pars	0.21	0.101	0.3123
Default wavelet denoising	0.236	0.102	0.413
Wavelet denoising w/GA smoothing pars	0.2037	0.1	0.3077
True noise	0.212	0.099	0.3154

Next, all the techniques were used to filter actual data from a nuclear power plant. Although there is no definite gauge of each technique's performance when using genuine data because the true signal is unknown, it is still useful to apply the techniques to a real-world problem. The data (shown in Fig, 7-7) come from three redundant steam pressure sensors in a nuclear power plant. This data set was chosen because these sensors have been found to be some of the noisiest sensors in most OLM models.

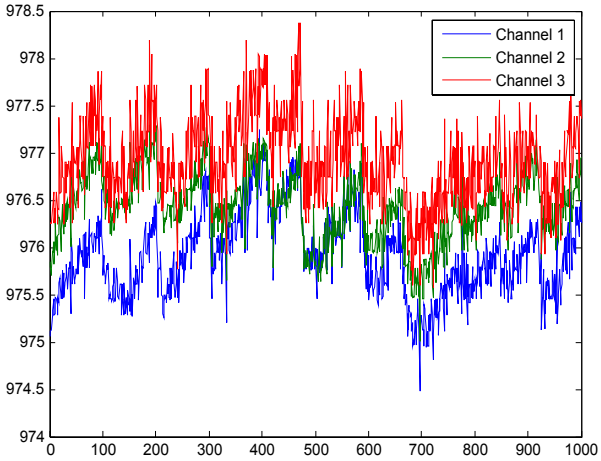


Fig. 7-7. Redundant steam pressure sensors.

Tables 7-5, 7-6, and 7-7 present the results of the techniques applied to the steam pressure signals. For all of the techniques, the correlations between the noise estimates are extremely high. This leads one to believe that some of the filtered noise is process noise and instrument noise. However, although the GA-optimized smoothing methods have high correlations, they are still lower than the same correlations for the smoothing methods using the default parameters. The plots of the techniques' performances on the real data set (Figs. 7-8, 7-9, and 7-10) show that the wavelet denoising methods still produce artifacts. Median filtering and kernel smoothing both performed well. Kernel smoothing seemed to perform slightly better, perhaps due to the extra degree of freedom given by its selection of both window size and bandwidth.

Table 7-5. Kernel smoothing results for steam pressure data

	Noise auto-correlation	Average noise correlation w/other noise	Noise correlation w/denoised sig	Fractional variance reduction	Estimated noise variance
Default parameter					
Channel 1	0.0244	1	–0.0486	0.2034	0.0519
Channel 2	0.0244	1	–0.0486	0.2034	0.0337
Channel 3	0.0244	1	–0.0486	0.2034	0.0582
Genetic algorithm parameter					
Channel 1	0.0183	0.6794	–0.0439	0.0545	0.0329
Channel 2	0.0173	0.4431	–0.052	0.0707	0.0279
Channel 3	0.0178	0.4369	–0.0871	0.1823	0.0434

Table 7-6. Median filtering results for steam pressure data

	Noise auto-correlation	Average noise correlation w/other noise	Noise correlation w/denoised sig	Fractional variance reduction	Estimated noise variance
Default parameter					
Channel 1	0.0205	1	–0.0627	0.1246	0.0325
Channel 2	0.0205	1	–0.0627	0.1246	0.0217
Channel 3	0.0205	1	–0.0627	0.1246	0.0365
Genetic algorithm parameter					
Channel 1	0.0243	0.7788	–0.0425	0.1043	0.0281
Channel 2	0.0227	0.899	–0.0015	0.0914	0.0126
Channel 3	0.0208	0.7785	–0.0216	0.1295	0.0342

Table 7-7. Wavelet denoising results for steam pressure data

	Noise auto-correlation	Average noise correlation w/other noise	Noise correlation w/denoised sig	Fractional variance reduction	Estimated noise variance
Default results					
Channel 1	0.0176	1	0.0089	0.1609	0.0365
Channel 2	0.0176	1	0.0089	0.1609	0.0231
Channel 3	0.0176	1	0.0089	0.1609	0.0515
Genetic algorithm results					
Channel 1	–0.0018	0.8511	0.0196	0.0824	0.0196
Channel 2	–0.0018	0.682	0.0197	0.1146	0.0185
Channel 3	–0.0016	0.6415	0.021	0.2099	0.0411

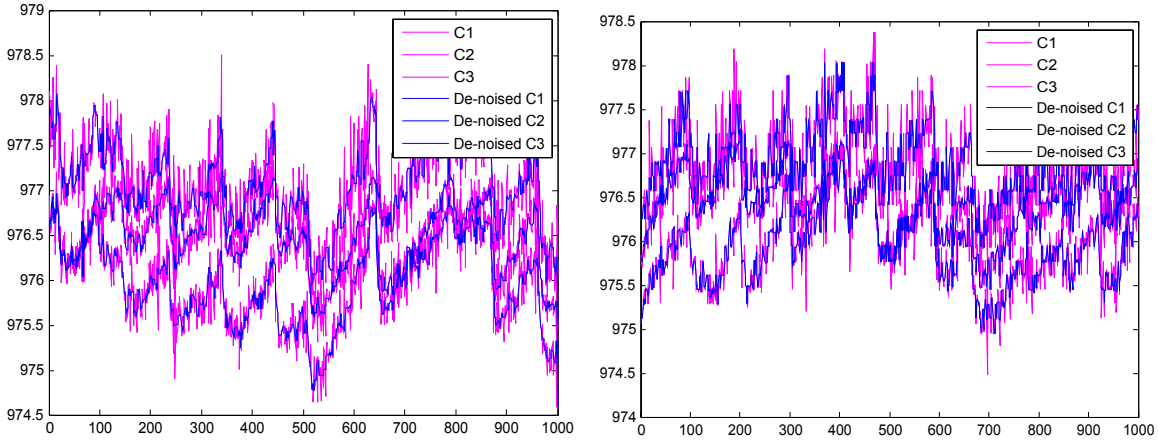


Fig. 7-8. Median filtering results on the steam pressure data set for (a) GA-optimized window size and (b) default window size.

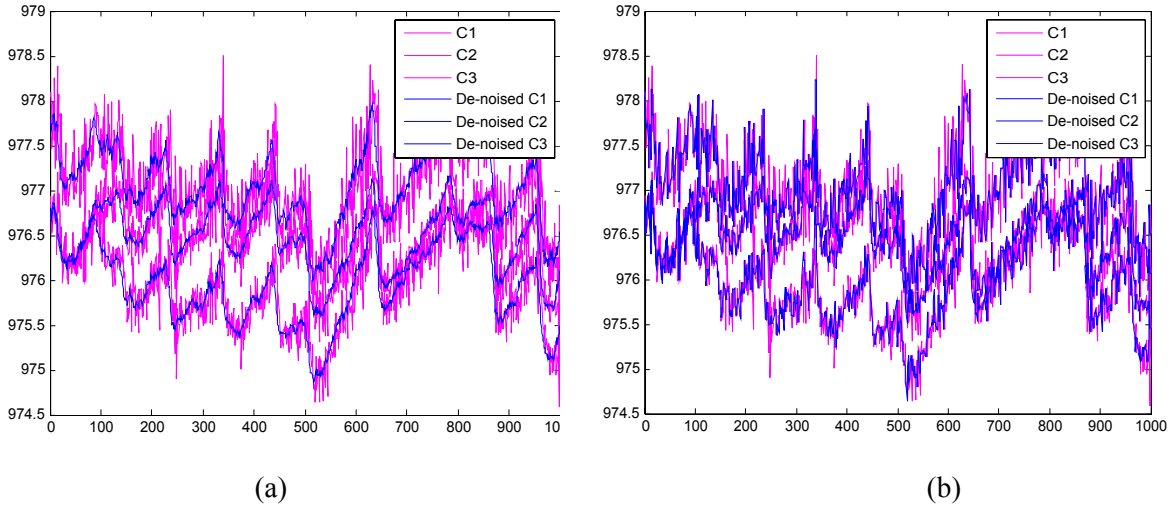


Fig. 7-9. Kernel smoothing results on the steam pressure data set for (a) GA-optimized parameters and (b) default parameters.

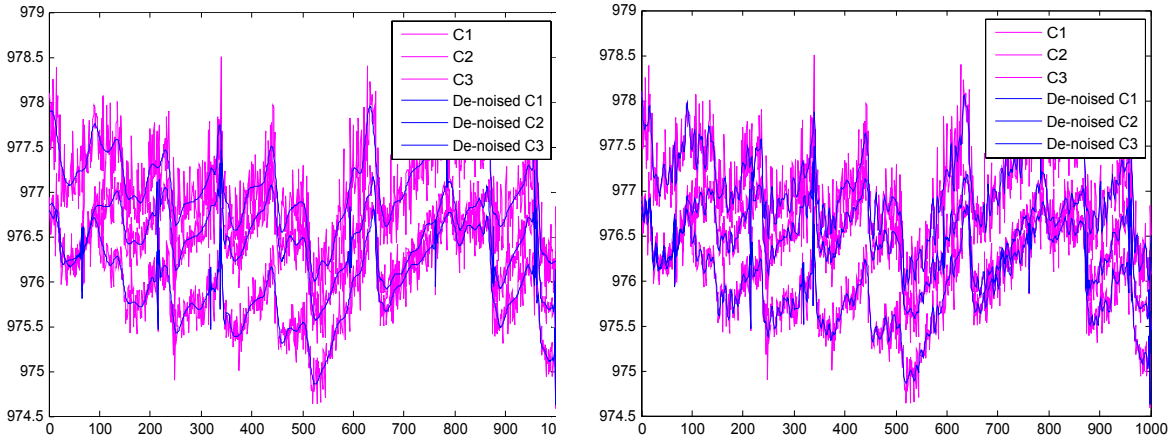


Fig. 7-10. Wavelet denoising results on the nuclear data set for (a) GA-optimized parameters and (b) default parameters.

Table 7-8 presents the calculated analytic uncertainty (%) for the channels in the steam pressure set. For this data set, the default settings consistently yielded larger uncertainties than using the GA-optimized parameters. However, the actual impact on uncertainty is fairly negligible, as the greatest difference in uncertainty for Channel 1 is 0.09%, for Channel 2, 0.17%, and for Channel 3, only 0.02%.

Table 7-8. Calculated analytic uncertainty estimates for nuclear steam pressure sensors

	Channel 1	Channel 2	Channel 3
Kernel smoothing w/default smoothing pars	0.3665	0.2949	0.1157
Kernel smoothing w/GA smoothing pars	0.298	0.123	0.1042
Default wavelet denoising	0.3901	0.2413	0.1155
Wavelet de-noising w/GA smoothing pars	0.3181	0.2252	0.0936
Default median filtering	0.3423	0.1818	0.1022
Median filtering w/GA smoothing pars	0.3341	0.1783	0.1070

7.3 Discussion and Recommendations

Overall, this study introduced an optimization method to improve signal denoising and evaluated the effect on estimated uncertainty. The method used a genetic algorithm (GA) to determine optimal smoothing parameters for the various filtering techniques. By correctly choosing these smoothing parameters, the removed noise had maximum variance, while still having a low autocorrelation, a low correlation to its corresponding signal, and a low correlation to the other removed noise signals. Although the noise removed using the GA technique proved to be more ideal, it had no significant impact on OLM uncertainty estimates. The main concern in OLM is the correct calculation of the OLM uncertainty, because this value is used to set the drift limit. However, the results of this study showed that the improved filtering had little influence on the uncertainty. They also showed that the wavelet denoising technique may be unsuitable for OLM. Even when optimized by a GA, wavelet denoising still

produced unwanted artifacts. Overall, each method gave acceptable denoising results. Median filtering performed well, but is limited in its filtering capabilities by only having window size as its single controllable parameter. Finally, kernel smoothing seemed the most suited to this application and conveniently lent itself to GA optimization.

Although the results of this study favor kernel smoothing, this report does not intend to imply that other techniques are less acceptable for OLM denoising. As shown by these results, denoising does not have a significant an impact on OLM uncertainty estimates. However, to guarantee that this fact holds true for a specific OLM denoising technique, it is strongly recommended that the de-noising technique is run with simulated data that have had artificial noise added so that the true noise value is known. By using these simulated data, one can verify that the denoising technique gives a fairly accurate estimate of the noise variance. One can also calculate the OLM uncertainty estimate using the "true" noise variance and the denoised noise variance estimate. This test will show the impact that the specific denoising technique has on overall OLM uncertainty. If the uncertainty is seen to vary greatly between the uncertainty calculated with the "true" noise variance and the estimated noise variance obtained through denoising, then a different, more accurate denoising technique is needed.

Page intentionally blank

8. EFFECTS OF DIFFERENT VECTOR SELECTION METHODS

8.1 Introduction

Vector selection, also called instance selection, performs a critical role in nonparametric model development. Predictions are made by comparing a query observation with historical observations called memory vectors. Because a query vector must be compared to each memory vector, large training data sets (commonly on the order of 50,000 observations) can easily lead to unacceptable computational loads. To lessen the computational burden, a subset of exemplar observations from the training set is chosen to be the full set of memory vectors. This subset selection poses two questions: first, how are the exemplar observations chosen, and, second, how many observations should be chosen? This chapter addresses the first of these questions by comparing three common methods of vector selection. The question of how many memory vectors to choose is addressed in the following chapter.

Several methods exist for memory vector selection. The most common of these methods include min-max selection, vector ordering, combination min-max and vector ordering, fuzzy c-means clustering, and Adeli-Hung clustering [Hines and Garvey 2006]. The two clustering methods, fuzzy c-means clustering and Adeli-Hung clustering, are in the research phase and have not been integrated into commercial OLM systems. The remaining three methods, min-max selection, vector ordering, and a combination of the two, are less computationally intensive and are used in commercially available systems. The following sections describe min-max vector selection, vector-ordering vector selection, and the combination of min-max and vector ordering.

8.1.1 Min-max vector selection

In min-max vector selection, exemplar or prototype vectors are selected according to the following procedure. First, the data are broken into a series of bands whose number is defined by the following ratio:

$$n_b = \frac{n_m}{2p} \quad . \tag{8.1}$$

Here, n_m is the number of vectors to be selected, p is the number of signals in the data, and n_b is the number of bands. In the next and final step, the minimum and maximum values for each signal are selected from each band.

This selection method was initially developed to facilitate the use of nonparametric modeling for industrial process surveillance. At that time computers were not powerful enough to evaluate a memory matrix containing thousands of training points, so a reduced set was needed. This reduced set was constructed by selecting the optimal exemplar vectors as described by Gross [Gross et al., 1998];

> "… [min-max selection] is optimal in the sense that [it] contains, at most, 2N vectors, where N is the number of signals or data points in the system; and these vectors span the full range that all sensors or data sources have noted during the available training period. Whenever two or more sensors or data sources simultaneously attain maxima or minima, the resulting number of training vectors will be less than 2N."

The selection method discussed in this section is an adapted form of the aforementioned method. In essence, min-max vector selection examines the p signals as being a collection of n_b operating regions and then selects the minimum number of memory vectors from each of the n_b bands [Liu and Zhang 2001].

For the sake of clarity, a schematic of the min-max vector selection algorithm is presented in Fig. 8-1. This example demonstrates the min-max vector selection procedure for selecting eight vectors from two signals (i.e., $n_m = 8$ and $p = 2$). Using this information, the appropriate number of sequential data bands is

$$n_b = \frac{8}{2 \times 2} = 2 \ . \tag{8.2}$$

This indicates that the data should be divided at its midpoint; specifically, Band I is the first half of the data, and Band II is the second half. Finally, the minimum and maximum values of the signals are selected from those two bands.

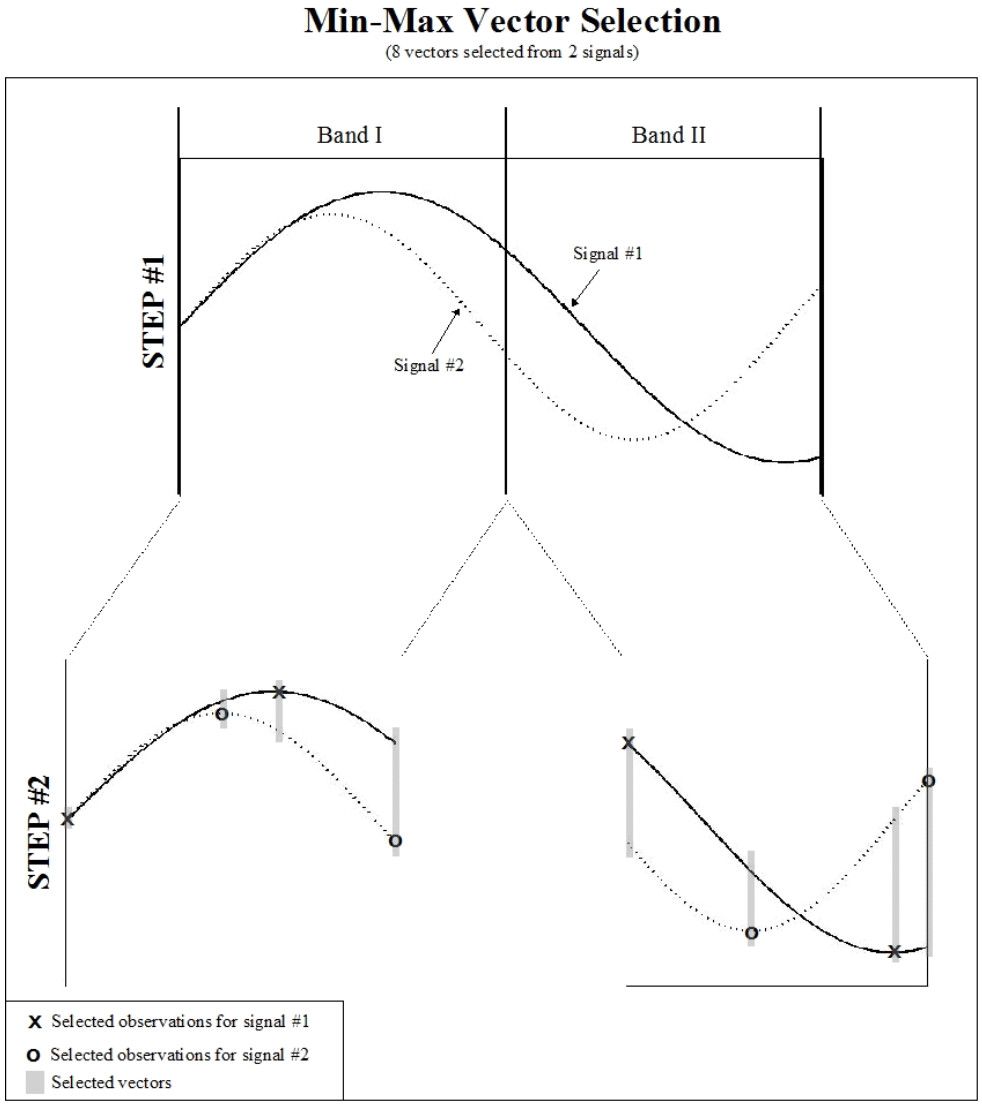


Fig. 8-1. Illustration of min-max vector selection method.

Before continuing, it is important to note that, in min-max vector selection, the selected vectors are guaranteed to bound the training data's operating range, but there is no such guarantee that the selected vectors will sufficiently cover the intermediate values between local minima and maxima.

8.1.2 Vector ordering

Vector ordering selection chooses vectors by first ordering the vectors according to some criteria and then periodically sampling the ordered vectors to yield a specified number of vectors (nm) [Zavaljevski and Gross 2000], [Gross et al., 2002]. Traditionally this selection method has been applied following the selection of the minimum and maximum vectors over the data's operating range [Herzog et al., 1998]. For this work, the vectors are ordered according to their Euclidean norm (N) in ascending order (i.e., smallest value first). If the vectors contain p signals, then the Euclidean norm of the ith vector is defined as:

$$N_i = \sqrt{X_{i,1}^2 + X_{i,2}^2 + \ldots + X_{i,p}^2} \quad . \tag{8.3}$$

Here, $X_{i,j}$ is the ith observation of the jth signal, and N_i is the Euclidean norm for the ith vector or observation of the p signals.

The Euclidean norm can also be interpreted as the distance of the ith vector from the origin. Therefore, this vector-selection method is inherently related to the location of the origin, and it is suggested that the data be mean-centered and unit-variance scaled prior to selecting vectors to control this effect.

To determine the size of the sequential sampling steps (n_s), simply divide the total number of vectors (n) by the number of vectors to be selected (n_m), that is,

$$\Delta n_s = \frac{n}{n_m} . \tag{8.4}$$

After ordering, the exemplar vectors are selected by sampling every n_s-sorted vector.

For the sake of clarity, a schematic of the vector-ordering selection algorithm is presented in Fig. 8-2. This example demonstrates the vector-ordering selection procedure for selecting eight vectors from two signals (i.e., $n_m = 8$ and $p = 2$). The first step in the selection procedure is to calculate the Euclidean norms. Next, the Euclidean norms are sorted and, finally, periodically sampled to select eight memory vectors. Notice that for n observations of the two signals, the step size or the number of points between the selected observations, Δn_s is equal to $n/8$.

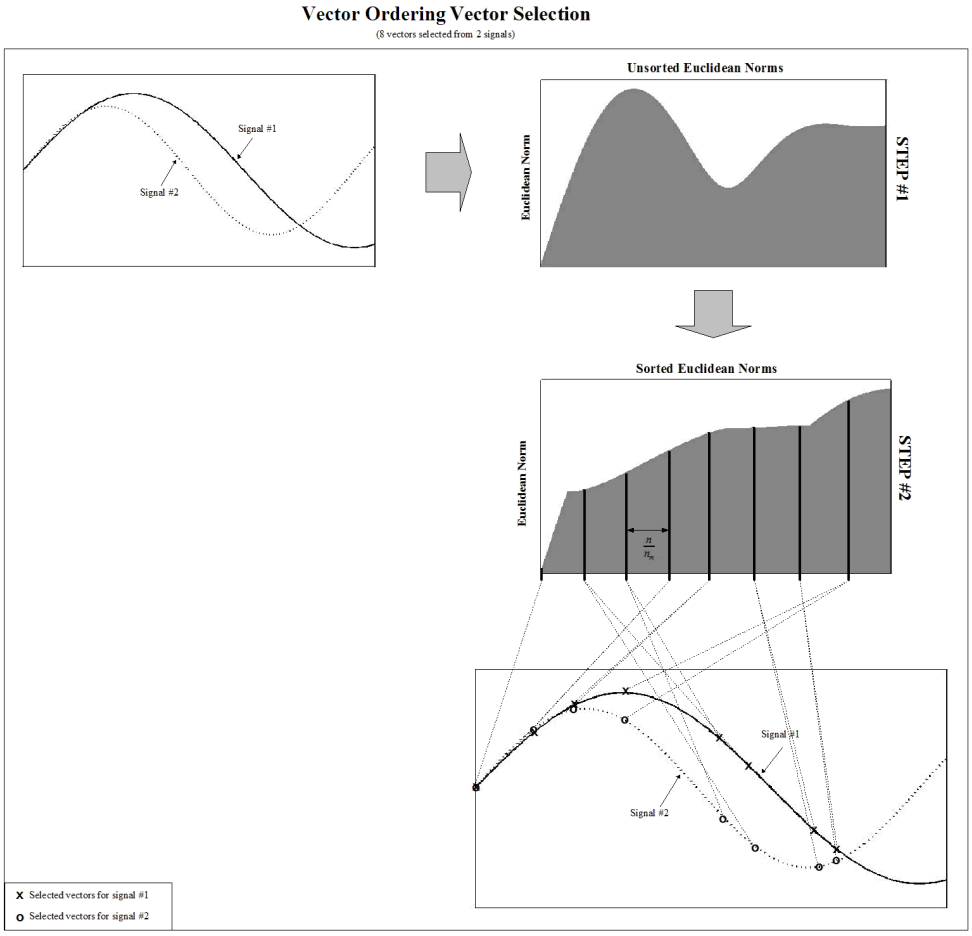


Fig. 8-2. Illustration of vector ordering vector selection method.

8.1.3 Combination of Min-max and vector ordering

The combination of these two methods to identify n_m memory vectors involves first extracting the minimum and maximum observations for each of the p signals, which results in at most $2p$ chosen observations. If one observation contains the minimum or maximum value for more than one signal, that observation is chosen only once. The remaining memory vectors are then chosen through the vector-ordering method, without replacement of the previously chosen vectors.

8.2 Results

Three models were constructed for each data set, one with each of the three vector selection methods described above. The results of these models are summarized below; full model results are contained in Appendix A.4.

8.2.1 Pressurizer-level model results

Models were developed using each of the three vector selection methods: min-max, vector ordering, and combination min-max/vector ordering. The subset of memory vectors chosen using each of these methods is shown in Fig. 8-3. It is clear from Fig. 8-3(a) that the min-max method does not choose memory vectors that cover the entire training region well. The memory vectors cover the maximum and minimum values, but do not include intermediate points in each signal. In contrast, Fig. 8-3(b) shows that

the vector-ordering method does a good job of choosing memory vectors in the intermediate range of the signals. However it does not fully enclose the range of the data; the early maximum and minimum points are not well represented in the memory vector subset. The final vector selection method, a combination of min-max and vector ordering shown in Fig. 8-3(c), strikes a balance between these two extremes. Several observations are chosen in the initial spikes of minima and maxima, but several intermediate values are also chosen. Based on these graphical representations, the combination vector selection method appears to best cover the training region, both in range and in intermediate observations.

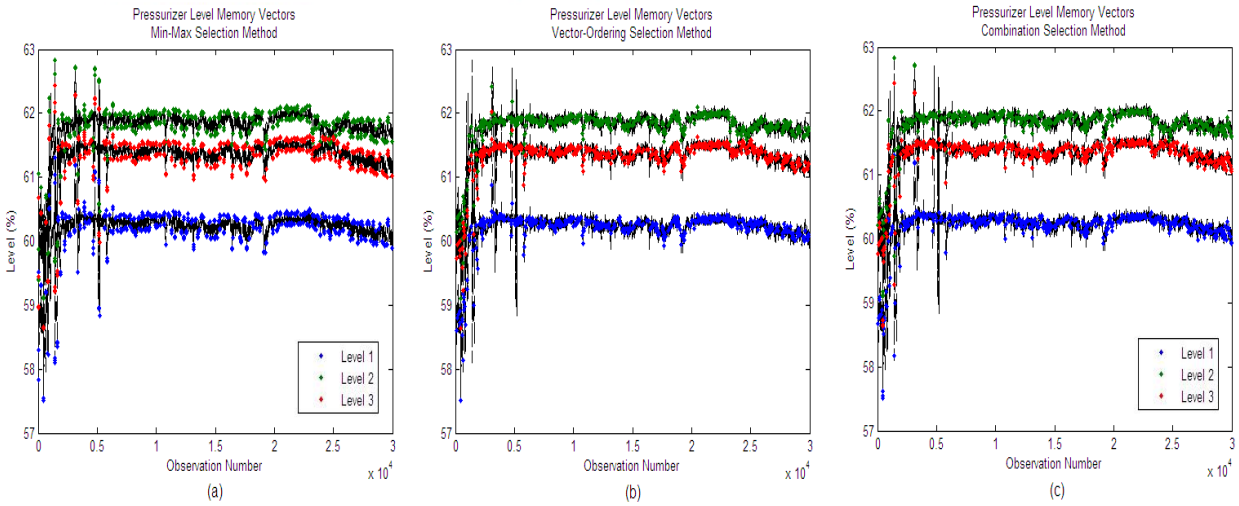


Fig. 8-3. Memory vectors for the Pressurizer-Level model selected by (a) min-max, (b) vector ordering, and (c) combination vector selection methods.

Table 8-1 summarizes the model metrics and uncertainty analysis for each model. As can be seen, with the exception of the Monte Carlo estimate of the model uncertainty, the three models have effectively equivalent performance. The Monte Carlo uncertainty estimate for the min-max selection method is much larger than the same estimate for the other two methods. This is likely due to the nature of Monte Carlo sampling that would have a greater effect on the minimum and maximum values in each signal range, but less effect on the intermediate values. The min-max method should be expected to give a larger uncertainty estimate with the Monte Carlo method. Despite this, each of these three vector selection methods would be adequate for model development. However, the combination selection method is considered a better choice for the reason stated above: it is most effective in choosing memory vectors that span the entire training region. This will result in more accurate predictions and better reliability assessment values, as discussed in Chapter 5.

Table 8-1. Results of Pressurizer-Level models with different vector selection methods

	Pressurizer-Level		
	Min-Max	**Vector ordering**	**Combination**
Metrics			
Accuracy (% of span)	0.061	0.064	0.065
Autosensitivity	0.595	0.539	0.537
Cross-sensitivity	0.322	0.310	0.310
EULM detectability (%)	0.316	0.260	0.439
SPRT detectability (%)	0.00046	0.00051	0.00182
Uncertainty			
Analytic (% of span)	0.115	0.124	0.127
Coverage	0.975	0.979	0.977
Monte Carlo (% of span)	0.404	0.134	0.141
Coverage	1.000	0.984	0.981

8.2.2 RPS Loop A model Results

Again, models were developed using each of the three vector selection methods: min-max, vector ordering, and combination min-max/vector ordering. The subset of memory vectors chosen using each of these methods is shown in Fig. 8-4. For clarity, the data are shown for only one set of redundant sensors, the first-stage turbine pressure. Similar results were seen in the other seven sensors. These plots lead to the same conclusions as the Pressurizer-Level data plots. The memory vectors chosen by the min-max vector selection method cover the maximum and minimum values very well but do not include as many intermediate points in each signal. Conversely, the vector-ordering method chooses memory vectors in the intermediate range of the signals, but does not cover the extrema of the data. The combination of min-max and vector ordering gives a good balance between these two extremes. Several observations are chosen in the initial spikes of minima and maxima, but several intermediate values are also chosen. As before, the plots below indicate that the combination vector selection method covers the training region, both in range and in intermediate observations, better than either min-max or vector ordering methods alone.

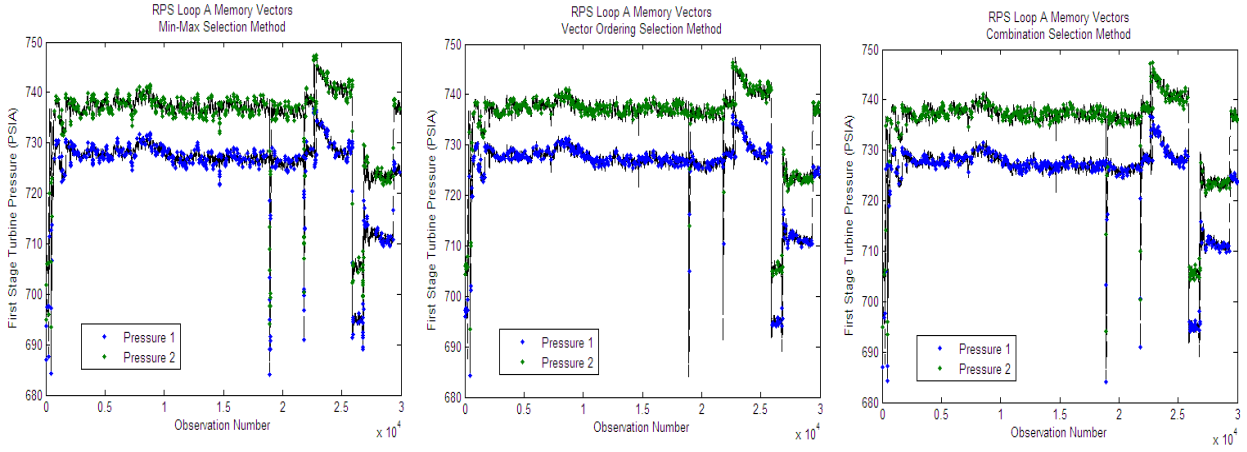


Fig. 8-4. Memory vectors for RPS Loop A first state turbine pressure sensors selected with (a) min-max, (b) vector ordering, and (c) combination vector selection methods.

The model metrics and uncertainty analysis for each model are summarized in Table 8-2. The results for the RPS Loop A models are very similar to those for the Pressurizer-Level models. Again, with the exception of the Monte Carlo estimate of the model uncertainty, the three models have effectively equivalent performance. Any of these three vector selection methods would be adequate for model development. However, the combination selection method is considered a better choice for the same reason stated above: it is most effective in choosing memory vectors that span the entire training region, resulting in more accurate predictions and greater reliability assessment values.

Table 8-2. Results for RPS Loop A models with different vector selection methods

	RPS Loop A		
	Min-Max	**Vector ordering**	**Combination**
Metrics			
Accuracy (% of span)	0.246	0.243	0.242
Autosensitivity	0.471	0.387	0.406
Cross-sensitivity	0.144	0.098	0.109
EULM detectability (%)	1.481	1.149	1.405
SPRT detectability (%)	0.00223	0.00185	0.00206
Uncertainty			
Analytic (% of span)	0.648	0.629	0.636
Coverage	0.994	0.993	0.994
Monte Carlo (% of span)	0.846	0.521	0.511
Coverage	1.000	0.999	0.998

8.3 Discussion and Recommendations

The three vector selection methods investigated, min-max, vector ordering, and combination, were able to develop models with comparable performance. Any of these selection methods would likely be acceptable in an OLM system. However, the combination method is chosen as the best vector selection method. This method was shown to choose memory vectors that more completely covered the training region, including both maximum and minimum signal observations as well as intermediate observations for each signal. This full coverage of the operating region will result in more accurate predictions and better reliability assessment values for future queries.

9. EFFECTS OF THE NUMBER OF MEMORY VECTORS

9.1 Introduction

Similarity based modeling methods, such as AAKR, are nonparametric modeling techniques that utilize the similarity of a query vector to historical observations to infer the model's response. For large data sets, the creation of local models may become cumbersome, because each training vector must be compared to the query vector. To alleviate this computational burden, vector sampling may be employed in order to identify a subset of the training data that is representative of the underlying process. In addition to reducing run time and computational complexity, an appropriate subset of the training space can improve model accuracy by avoiding excessive noise and over-fitting [Wilson and Martinez, 2000]. Model development involves identifying the appropriate number of memory vectors. Too few memory vectors will give a fast run time, but poor model performance. Too many memory vectors will give improved model performance, but at the expense of computational speed. A balance must be struck between model performance and model run time. The appropriate balance here depends on the specific needs of the OLM system.

Models were developed using the combination vector selection method described in the previous chapter to choose subsets of exemplar vectors of different sizes. Models were developed with 10, 25, 50, 100, 500, 3,750, 7,500, and 15,000 memory vectors from a total of 30,000 observations in the training set. The results of these models are summarized in the following section.

9.2 Results

One objective of an OLM system is to minimize the accuracy metric value, characterized by the mean squared error of error-free test data. Model uncertainty must also be minimized, while maintaining an acceptable residual coverage. These two objectives should be balanced with minimizing model run time. An overly complex model could require both storage capacity and run time large enough to make it unsuitable for OLM systems. The following sections summarize results of Pressurizer-Level and RPS Loop A models developed to investigate the effect of memory vector subset size on these metrics. Eight models were developed for each set of sensors. The training data sets described earlier include 30,000 observations. These observations were sampled using the combination min-max/vector-ordering vector selection method to identify memory vector subsets with 10, 25, 50, 100, 500, 3,750 (an eighth of the training observations), 7,500 (a quarter of the training observations), and 15,000 (half of the training observations). For each subset of memory vectors, the accuracy metric was evaluated. In addition, analytic uncertainty values were estimated. Model run time needed for each of these estimates was also measured. Full results for each model are tabulated in Appendix A.5.

9.2.1 Results of the Pressurizer-Level models

Eight models were developed with the memory vector subsets described above. These models were evaluated for accuracy, uncertainty, and run time. The objective of this analysis is to identify a memory vector subset that gives a balance between minimizing the accuracy metric value or uncertainty, while also minimizing model run time.

Figure 9-1 gives a plot of total run time vs the model's value of the accuracy metric for the eight models. The total run time includes the time needed to develop, train, and test the model as well as evaluate both the analytic and Monte Carlo uncertainty estimates. In an actual OLM system, only one uncertainty method need be employed, and model run time will be reduced accordingly. In addition, the run time of interest would be the prediction time for a new query observation. For continuous monitoring, this time is constrained by the data sampling rate. As can be seen, this analysis gives a strong L-shaped curve, which facilitates identifying the optimum number of memory vectors. The optimum

value corresponds to the point that gives the elbow of the curve: 500 memory vectors, as indicated on the plot. It is clear to see from this plot that fewer memory vectors give decreased performance, and more memory vectors result in increased run time with only a small improvement in performance. A subset of 500 memory vectors gives an appropriate balance between the two.

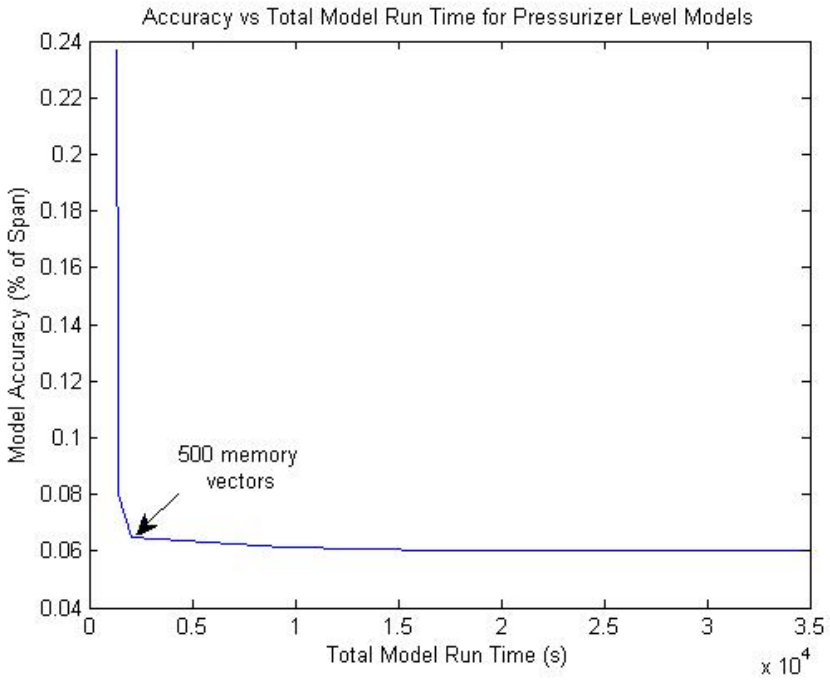


Fig. 9-1. Run time vs accuracy for the Pressurizer-Level models.

Figure 9-2 shows a plot of the run time needed to evaluate the analytic uncertainty estimate vs that same uncertainty. Monte Carlo uncertainty estimate time would also increase with the number of memory vectors; however, run time for these estimates is not considered because these estimates are computed beforehand, offline and would not contribute to online model run time. Again, this plot makes a clear L-curve with elbow corresponding to the model with 500 memory vectors. It is obvious that fewer memory vectors give higher model uncertainty, which degrades the applicability of an OLM system. Conversely, more memory vectors give longer model run times with only minimal improvement in model performance. Again, 500 memory vectors give a good balance between uncertainty and run time.

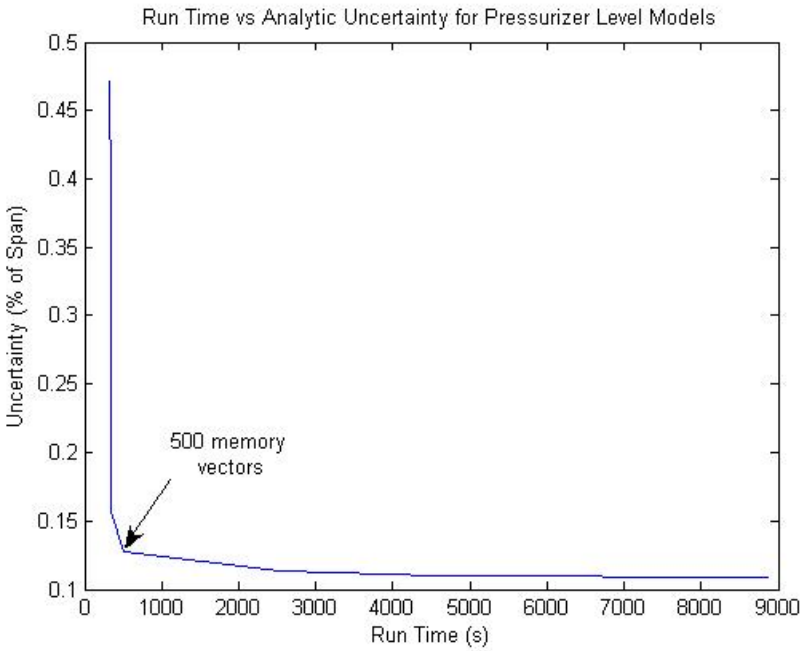


Fig. 9-2. Run time vs analytic uncertainty for Pressurizer-Level models.

Analysis of both model accuracy and model uncertainty vs model run time indicates that 500 memory vectors is an appropriate number to give greatest model performance with least model run time for this Pressurizer-Level data set.

9.2.2 RPS Loop A models

Again, eight models were developed using the memory vector subsets described above. These models were evaluated for the same three metrics: accuracy, uncertainty, and run time. Analysis of these three metrics was performed in order to identify a memory vector subset that gives a balance between minimizing accuracy or uncertainty, while also minimizing model run time.

Figure 9-3 gives a plot of total run time versus model accuracy for the eight models. The total run time includes the time needed to develop, train, and test the model as well as evaluate both the analytic and Monte Carlo uncertainty estimates. In an actual OLM system, only one uncertainty method need be employed, and model run time will be reduced accordingly. Although this analysis does not give as clear an optimum point as the Pressurizer-Level models, 500 memory vectors has been identified as the optimum number of memory vectors. This is because the next model (that with 3750 memory vectors) gives only ~14% (0.20% vs 0.24%) improvement in model accuracy with an increase of ~460% in model run time (4.7 h vs 0.83 h). There may exist a more optimum number of memory vectors between 500 and 3750. However, for the memory vector subsets tested, a subset of 500 memory vectors gives an appropriate balance between the two. It is clear in the plot that fewer memory vectors give decreased performance, and more memory vectors result in increased run time with contrastingly small improvements in model performance.

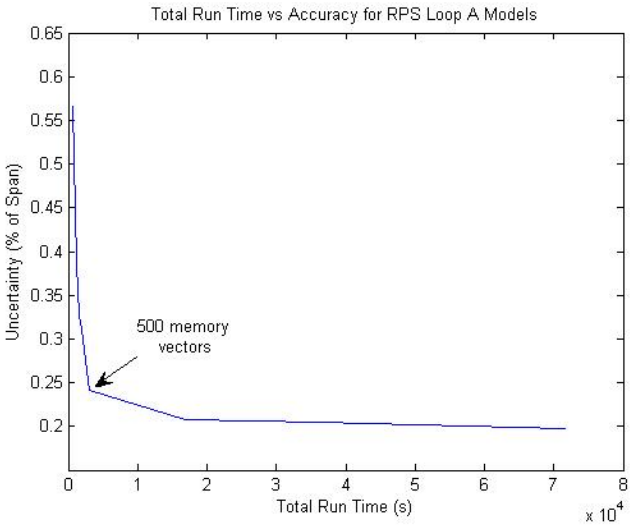


Fig. 9-3. Run time vs accuracy for the RPS Loop A models.

Figure 9-4 shows a plot of the run time needed to evaluate the analytic uncertainty estimate vs that same uncertainty. Again, this plot does not make as clear L-curves as in the Pressurizer-Level models. However, the optimum model performance is associated with models developed with 500 memory vectors, as indicated. It is clear that fewer memory vectors result in degraded model performance due to higher model uncertainty estimates. The next subset of memory vectors evaluated contained 3750 observations. This model results in 20% improvement for analytic uncertainty estimates. These improvements, however, come with a 500% increase in model run time. Another subset of memory vectors, with a number of observations between 500 and 3750 could give a more optimum result. However, for the models evaluated, 500 memory vectors give a good balance between uncertainty and run time for this data set.

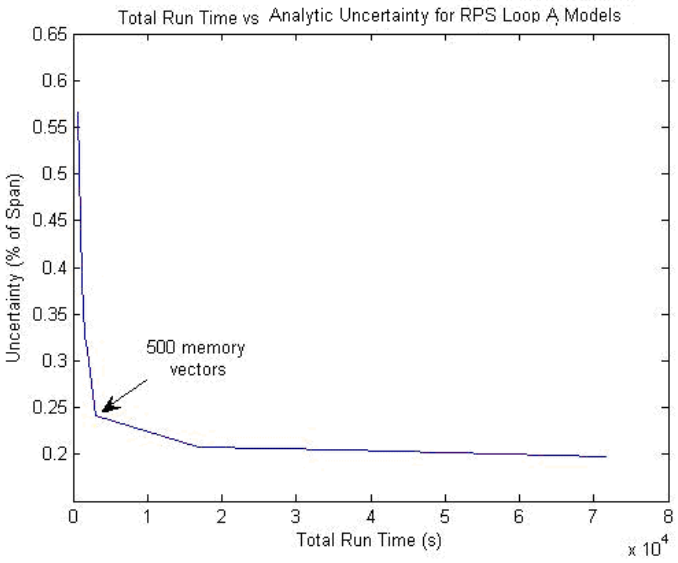


Fig. 9-4. Run time vs analytic uncertainty for RPS Loop A models.

Analysis of both model accuracy and model uncertainty vs model run time indicates that, for the cases considered here, 500 memory vectors is an appropriate number to give greatest model performance with least model run time for the RPS Loop A data set considered here.

9.3 Discussion and Recommendations

The analysis presented in this section has highlighted the importance of using an appropriate number of memory vectors in model development. A model with too few memory vectors results in degraded model performance; model accuracy and uncertainty estimates can be high enough to preclude use in an OLM system. A model with more memory vectors than needed, however, results in increased model run time. These models can be too computationally intensive for practical applications. For continuous monitoring, the run time for one query vector, including prediction and uncertainty estimation, is constrained by the data sampling rate. It was seen for the two data sets investigated here that 500 memory vectors was an appropriate number for adequate model development. This is generally considered to be true. For data sets with more than 1000 observations in the training data set, 500 memory vectors is an appropriate amount for model development. For training data sets with less than 1000 observations, a subset of memory vectors with half the number of training observations should be adequate. However, care should be taken when developing a model with any amount of training data to ensure that the vector selection method results in a subset of memory vectors that spans the entire operating region, including sensor maxima, minima, and intermediate ranges. In cases in which accuracy is extremely important and run time is not limited, larger memory matrices will probably be beneficial.

Page intentionally blank

10. EFFECT OF DISTANCE CALCULTION METHOD

10.1 Introduction

The applicability of an OLM system is directly related to the ability of an empirical model to correctly predict sensor values when it is supplied with faulty query data. For this reason methods should be developed to ensure that robust empirical models can be developed. The use of robust distance measures is one possible solution to this problem. Two robust distance functions are presented here for use in nonparametric, similarity based models, such as auto-associative kernel regression. The performance of auto-associative OLM systems is measured in terms of its accuracy, auto-sensitivity, and cross-sensitivity. The accuracy metric measures the ability of the model to correctly predict sensor values and is normally presented as the mean squared error (MSE) between the prediction and the correct sensor value. Auto-sensitivity measures the ability of the model to make correct sensor predictions when the respective sensor value is incorrect due to some sort of fault. Cross-sensitivity measures the effect a faulty sensor input has on the other sensor predictions in the model. An ideal system would be accurate and would not have sensor predictions affected by degraded inputs.

The most basic form of the AAKR modeling technique makes use of the Euclidean distance or L^2-norm given by Eq. (10.1):

$$u_j = \sqrt{\sum_{i=1}^{n} \left(x_q^i - m_j^i \right)^2} , \tag{10.1}$$

where

u_j is the distance between the query vector (x) and jth memory vector (m_j),
n is the number of variables in the data set,
x_q^i is the ith variable of the query vector,
m_j^i is the ith variable of the jth memory vector.

Because this distance function squares the individual differences, the effects of a faulty input may be amplified, resulting in parameter predictions that are more affected by input variations and are therefore less robust. To improve robustness, distance measures that are not affected by errant sensor readings are desired, and two robust distance functions have accordingly been investigated.

The first robust distance function is the L^1-norm, which is defined by Eq. (10.2):

$$u_j = \sum_{i=1}^{n} \left| x_{qi} - m_{ij} \right| . \tag{10.2}$$

Notice that rather than the squares of the individual differences, the L^1-norm sums the absolute values of these distances. This alteration will be shown to provide a modest improvement in robustness, but the distance measure can still be affected by faulty input. The next robust distance function attempts to dynamically remove faulty input from the distance calculation and therefore should provide the largest improvement to model robustness.

The second robust distance function is called the robust Euclidean distance and is defined by the following equation:

$$u_j = \sqrt{\sum_{i=1}^{n} (x_q^i - m_j^i)^2 - \max_{i=\{1,\ldots,n\}} \left[(x_q^i - m_j^i)^2 \right]} \ . \tag{10.3}$$

Here $\max_{i=\{1,\ldots,n\}} \left[\left(x_q^i - m_j^i \right)^2 \right]$ is the maximum squared difference of the query vector from the *j*th memory vector. Simply speaking, one "bad performer" is assumed to exist, and its influence is removed from the calculation. The following example, with the query vector having a fault in sensor 2, can more clearly illustrate use of the robust distance measure:

$$\begin{aligned} x_q &= [0.9501 \quad 0.2311 \quad 0.6068 \quad 0.4860] \\ m_j &= [0.8913 \quad 1.7621 \quad 0.4565 \quad 0.0185] \end{aligned} \ .$$

The squared differences are calculated in Eq. (10.4):

$$\left(x_q - m_j \right)^2 = [0.0035 \quad 2.3438 \quad 0.0226 \quad 0.2185] \ . \tag{10.4}$$

The largest squared difference is 2.34 due to the faulty sensor. The robust Euclidean distance is defined to be the square root of the sum of the squared distances minus the largest squared difference:

$$u_j = \sqrt{2.5884 - 2.3438} = 0.4946 \, . \tag{10.5}$$

The robust Euclidean distance is simply the Euclidean distance with the largest distance or worst performer removed.

10.2 Results

As the specific model results below show, the use of a robust distance measure can improve model performance for both cross- and auto-sensitivity. This means that models developed with a robust distance measure are less affected by faulty query values; these models are considered to be more robust. Marginal degradations were shown for the accuracy metric and uncertainty of the robust models. Development of more robust models often results in a trade-off between sensitivity and model accuracy and uncertainty. The specific OLM application must be considered to determine if the degradation in accuracy and uncertainty is acceptable in light of the improved robustness. The uncertainty and coverage results given for both the Pressurizer-Level and RPS Loop A models are for the analytic uncertainty estimation. The Monte Carlo uncertainty estimation gave similar results. A full results table for all the models using a robust distance measure can be found in the Appendix A.6.

10.2.1 Pressurizer-Level model results

The use of either robust distance measure in developing the Pressurizer-Level model gives some model performance improvement on two model metrics: cross-sensitivity, and auto-sensitivity. As can be seen in Table 10-1, use of the L^1-norm showed a modest improvement for each of these metrics, as well as a slight reduction in the accuracy metric and a reduction in the model uncertainty without a significant reduction in the model coverage. The Robust Euclidean distance, however, showed a significant improvement for cross- and auto-sensitivity, with only small degradations in accuracy and model uncertainty. The use of the Robust Euclidean distance results in a model that is much more robust to faulty data queries. This model would be expected to more correctly predict the actual sensor values given a query observation with a drift, an outlier value, or any other fault.

Table 10-1. Comparison Pressurizer-Level model performance using different distance calculation methods

		Pressurizer Level			
		Signal 1	Signal 2	Signal 3	Average
Accuracy (% of Span)	Euclidean	0.055	0.0377	0.102	0.065
	L^1-norm	0.052	0.026	0.113	0.063
	Robust Euclidean	0.043	0.019	0.168	0.077
Cross-Sensitivity	Euclidean	0.301	0.274	0.355	0.310
	L^1-norm	0.24	0.275	0.352	0.289
	Robust Euclidean	0.099	0.075	0.245	0.140
Auto-Sensitivity	Euclidean	0.466	0.712	0.434	0.537
	L^1-norm	0.418	0.723	0.348	0.496
	Robust Euclidean	0.218	0.264	0.068	0.183
Uncertainty	Euclidean	0.107	0.071	0.203	0.127
	L^1-norm	0.101	0.045	0.224	0.123
	Robust Euclidean	0.081	0.033	0.334	0.149
Coverage	Euclidean	0.993	0.953	0.995	0.980
	L^1-norm	0.947	0.987	0.966	0.967
	Robust Euclidean	0.938	0.954	1.000	0.964

10.2.2 RPS Loop A model results

The effects of using a robust distance measure are very different for the nonredundant sensor model, as can be seen in Table 10-2. The L^1-norm distance measure resulted in a model with a slightly improved auto-sensitivity. However, the cross-sensitivity and accuracy metrics showed that performance was slightly degraded in these areas. The model uncertainty also increased slightly with the use of the L^1-norm. These changes, however, are very small and are unlikely to be statistically significant. Overall, performance of this model with the L^1-norm distance is equivalent to that with the Euclidean distance.

The Robust Euclidean distance showed more significant changes in model performance. This model showed improvement in both cross- and auto-sensitivity. Again, the use of the Robust Euclidean distance resulted in a model that was more robust to faulty data queries. In addition, the model uncertainty was

reduced without a significant change in the model coverage. Conversely, the model accuracy was slightly degraded due to the robust Euclidean distance.

Table 10-2. Comparison of RPS Loop A model performance using different distance calculation methods

		RPS Loop A									
		Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9	Average
Accuracy (% of Span)	Euclidean	0.261	0.246	0.290	0.229	0.919	0.120	0.027	0.037	0.046	0.242
	L^1-norm	0.272	0.260	0.298	0.252	0.972	0.117	0.026	0.041	0.047	0.254
	Robust Euclidean	0.234	0.232	0.317	0.290	1.050	0.124	0.022	0.041	0.045	0.262
Cross-Sensitivity	Euclidean	0.120	0.119	0.098	0.116	0.079	0.046	0.130	0.125	0.150	0.109
	L^1-norm	0.132	0.133	0.134	0.138	0.099	0.056	0.117	0.143	0.141	0.121
	Robust Euclidean	0.086	0.092	0.099	0.091	0.070	0.050	0.104	0.120	0.099	0.090
Auto-Sensitivity	Euclidean	0.420	0.451	0.620	0.714	0.160	0.082	0.405	0.370	0.433	0.406
	L^1-norm	0.417	0.445	0.645	0.727	0.032	0.171	0.454	0.271	0.411	0.397
	Robust Euclidean	0.393	0.378	0.584	0.636	0.047	0.140	0.421	0.442	0.342	0.376
Uncertainty	Euclidean	0.840	0.818	0.897	0.899	1.840	0.235	0.043	0.067	0.083	0.636
	L^1-norm	0.943	0.918	1.010	1.010	1.940	0.229	0.043	0.075	0.086	0.695
	Robust Euclidean	0.635	0.618	0.677	0.680	2.090	0.241	0.029	0.074	0.077	0.569
Coverage	Euclidean	1.000	1.000	1.000	1.000	1.000	0.987	0.975	0.996	0.991	0.994
	L^1-norm	1.000	1.000	1.000	1.000	1.000	0.989	0.981	0.996	0.983	0.994
	Robust Euclidean	1.000	1.000	0.999	1.000	1.000	0.983	0.892	0.998	0.973	0.983

10.3 Discussion and Recommendations

It was seen above that the use of a robust distance measure has performance advantages for the task of sensor drift detection. The L^1-norm showed small improvements in the model performance metrics cross-sensitivity and auto-sensitivity; the robust Euclidean distance measure gave larger improvement in these metrics. When robust distance measures are used, a trade-off can occur between model accuracy and model sensitivity. Often this degradation in accuracy is only slight compared to the improvement in model sensitivity. The acceptability of this trade-off depends on the specific OLM application. In the case of sensor condition monitoring, a small degradation in accuracy and uncertainty is an acceptable trade for greater model sensitivity.

11. CONSIDERATIONS FOR REDUNDANT SENSOR MODELS

Due to the common use of redundant sensors in nuclear power plants, the uncertainty estimation methods used for nonredundant models should be validated for redundant models. In this section, the term redundant models will be used for averaging techniques such as the Instrumentation and Calibration Monitoring Program (ICMP) algorithm developed by EPRI [1993], and the term nonredundant models will be used for models such as auto-associative kernel regression. Research has shown that redundant sensor models respond much differently to Monte Carlo uncertainty analysis methods and thus require additional investigation. This section will review redundant sensor model architectures, provide a detailed description of the ICMP algorithm, explore the use of both confidence intervals and prediction intervals, and conclude with a case study.

11.1 Generic Redundant Sensor Model Architectures

Empirical models designed for redundant sensors include simple averaging, the Instrumentation and Calibration Monitoring Program (ICMP) [EPRI 1993], and the parity space algorithm (PSA), also known as the generalized consistency check (GCC) [Upadhyaya et al., 1988]. Even though there are subtle differences in the model architectures, all perform a weighted average of the redundant sensors to estimate a single or multiple outputs. To begin this discussion, consider the multiple-input, single-output (MISO) model as presented in Fig. 11-1, where m_i is the i^{th} redundant sensor measurement, n is the number of redundant sensors, and $\hat{x}$ is the estimate of the parameter being measured by the n redundant sensors.

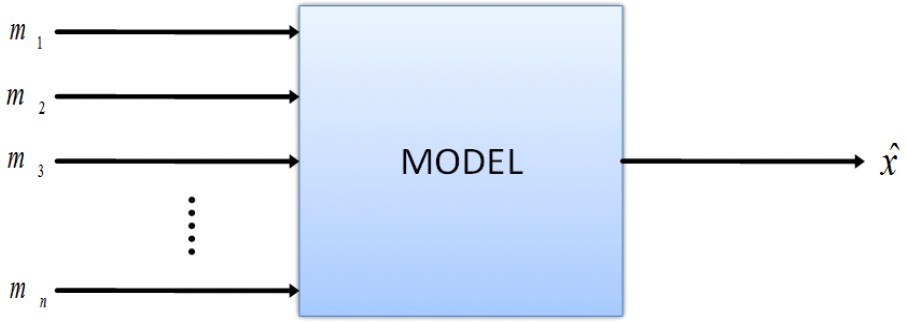


Fig. 11-1. Diagram of a generic MISO redundant sensor model.

In such a model, the parameter prediction for time t is a weighted average of the n redundant sensor measurements at time t.

$$\hat{x}(t) = \frac{\sum_{i=1}^{n} w_i(t) \times m_i(t)}{\sum_{i=1}^{n} w_i(t)}. \tag{11.1}$$

Here, the weights $w_i(t)$ are determined by evaluating the consistency of the measured value relative to the measurements of the other redundant sensors.

In a multiple-input, multiple-output (MIMO) redundant sensor model (Fig. 11-2), rather than assume that there are no inherent biases in the redundant sensors (training data), the model is trained to replicate these biases in its predictions. The trained model is then used to monitor the sensor outputs for future drifts. It is important to note that in MIMO models, the biases of the individual sensors are estimated from the training data, which may not be able to reconcile biases in the training data. In other words, if the model is trained with drifted sensors, the model will be able to detect a progression of the drift and will interpret the initial drifted bias as being its normal, undrifted state. As seen in Chapter 4, investigation of the training data is particularly important for any OLM model development.

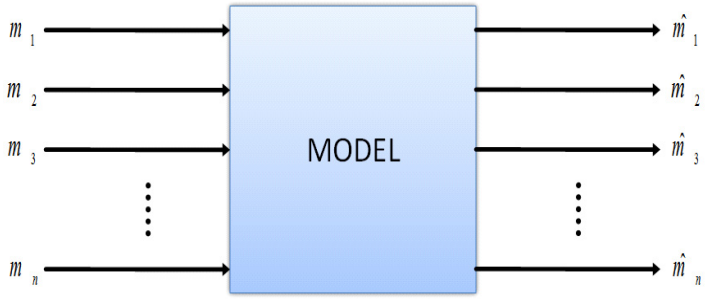


Fig. 11-2. Diagram of generic MIMO redundant sensor model.

If $\hat{b}_i$ is an estimate of the initial bias for the *i*th sensor, the model predictions for the *i*th sensor are the sum of the weighted average of the centered observations (i.e., observations minus the bias) and the sensor bias:

$$\hat{m}_i(t) = \frac{\sum_{j=1}^{n} \left| w_j(t) \times m_j(t) - \hat{b}_j \right|}{\sum_{j=1}^{n} \left| w_j(t) \right|} + \hat{b}_i \quad . \tag{11.2}$$

Comparing Eq. (11.2) to Eq. (11.1), it is clear that the MIMO model depends more on the training data since the sensor biases must be estimated during model development. For the sake of convenience, the description of ICMP presented in Vol. 2 is reproduced in the next section. The intent of this inclusion is to provide a description of a specific redundant sensor model architecture.

11.2 ICMP Review

The Instrumentation and Calibration Monitoring Program (ICMP) algorithm is a redundant algorithm used for OLM. It was developed by EPRI in the early 1990s and essentially was the original method used to perform OLM [EPRI 1993]. In ICMP, a weighted averaging algorithm is used to determine an estimate of the true process parameter. The ICMP algorithm assigns a consistency value, C_i, to each of the signals for each data sample evaluated. This consistency value denotes how much of the signal's measured value contributes to the process estimate. The value is based on the absolute difference between a given sensor and other sensors in the group. Thus, inconsistent signals contribute less to the process estimate. For example, for a group of three redundant sensors, the consistency value compares the output of each instrument to the output of the other two instruments. If the *i*th instrument's output is

sufficiently close to the output of both of the other instruments, its consistency value, C_i, will be 2. However, if the $i$rh instrument's output is only sufficiently close to one of the other instruments, then C_i will be 1. If the ith instrument's output is not close to either of the two remaining instruments, then the consistency value, C_i, will be 0. Overall, if a signal agrees within a tolerance to another signal in the group, they are declared to be consistent, and the consistency value for that signal, C_i, is found with Eq. (11.3).

$$\text{If } \left|m_i - m_j\right| \le \delta_i + \delta_j \text{, then } C_i = C_i + 1 \text{ ,} \tag{11.3}$$

where

C_i = the consistency value of the ith signal,
m_i = the output for signal i,
m_j = the output for signal j,
δ_i = the consistency check allowance for instrument i,
δ_j = the consistency check allowance for instrument j.

The values for the consistency check allowances are dependent on the uncertainty present in the signals, such as

$$\left|m_i - m_j\right| \le 2\delta \text{ .} \tag{11.4}$$

After the consistency values are calculated, the ICMP parameter estimate can be calculated as

$$\hat{x} = \frac{\sum_{i=1}^{n} w_i C_i m_i}{\sum_{i=1}^{n} w_i C_i} \text{ ,} \tag{11.5}$$

where

$\hat{x}$ = the ICMP parameter estimate for the given data sample, and
w_i = the weight associated with the ith signal.

The weight values are included to allow the user to apply a greater weighting to more accurate or reliable sensors within a redundant group. If there is no preference within the group, all weight values can be set to 1, reducing the equation to

$$\hat{x} = \frac{\sum_{i=1}^{n} C_i m_i}{\sum_{i=1}^{n} C_i}, \tag{11.6}$$

The consistency check factor controls the influence of an individual signal on the ICMP parameter estimate. If all sensors are considered equally consistent, the ICMP estimate is just the simple average of the redundant sensors. If a sensor's consistency value is zero, it will not influence the parameter estimate. If all sensors are inconsistent, the parameter estimate is undefined.

Once the parameter estimate is calculated, the ICMP algorithm evaluates the performance of each individual sensor relative to the parameter estimate. This is done through the use of an acceptance criterion.

$$\text{If } \left|\hat{x} - m_i\right| \geq \alpha_i, \tag{11.7}$$

then m_i has potentially drifted beyond desired limits, where α_i is the acceptance criterion for the ith signal.

When the deviation between a sensor's measurement and the current parameter estimate exceeds the acceptance criterion, that sensor is considered to have drifted out of calibration. At this point the sensor is assumed to have failed. Note that failing the acceptance criterion does not necessarily disallow the failed sensor's value to influence the ICMP estimate. The consistency check factor must also be exceeded, and it is not necessarily related to the acceptance criterion. The 2002 paper, *Monte Carlo Analysis and Evaluation of the Instrumentation and Calibration Monitoring Program,* further details the relationship between the acceptance criteria and the consistency check factor and also provides numerical examples of the ICMP algorithm for varying sensor groups [Rasmussen et al., 2002].

ICMP software was successfully installed at the Catawba and V.C. Summer Nuclear Stations [EPRI 2000]. Although these plants were using ICMP only as a performance monitoring and troubleshooting tool, they obtained positive results. These results helped to verify ICMP's diagnostic capabilities. The plants did note some of ICMP's inherent shortcomings. For instance, ICMP performed poorly when there was limited instrumentation, as is normally found on the secondary side of a nuclear plant. ICMP also is unable to detect common-mode drift failure (where all redundant instruments drift in the same direction at the same rate). However, as current calibration practices offer limited protection against common-mode failure, ICMP's inability to detect common-mode failure should not invalidate the technique. Still, due to some of ICMP's limitations, most plants (including V.C. Summer and Catawba) have migrated to the more advanced OLM modeling techniques, such as AANN, AAKR, and AAMSET. For more information about the algorithm and how to quantify its associated uncertainty, the interested reader is referred to *Monte Carlo Simulation and Uncertainty Analysis of the Instrument Calibration and Monitoring Program* [EPRI 1995] and *Monte Carlo Analysis and Evaluation of the Instrumentation and Calibration Monitoring Program* [Rasmussen 2002].

11.3 Confidence Intervals for Empirical Models

Recall that for nonredundant sensor models, the uncertainty is quantified by the 95% confidence interval, specifically:

$$U\left(\hat{x}\right) = t_{n-p,\alpha/2}\ \sqrt{Var\left(\hat{x}\right) + Bias^2}\ , \tag{11.8}$$

where n is the number of training observations; p is the number of variables used to infer y; $t_{n\text{-}p,\alpha/2}$ is the t-statistic that approximates the normal distribution for n-p degrees freedom and confidence level 1-α. Additionally, $\mathrm{Var}(\hat{x})$ is the variance of the model predictions $\hat{x}$ and Bias is the model bias. Furthermore, the bias is estimated by applying the bias-variance decomposition equations of the error (Chapter 4 of Vol. 2). Specifically, the squared bias is the difference of the mean squared error (MSE) and the sum of the prediction and noise variance, $\hat{\sigma}_{\varepsilon}^2$.

$$Bias^{\,2} = MSE\left(\hat{\mathbf{x}}\right) - Var\left(\hat{\mathbf{x}}\right) - \hat{\sigma}_{\boldsymbol{\varepsilon}}^2\ . \tag{11.9}$$

When using this equation, if the squared bias is less than zero (i.e., the prediction and noise variance more than compensate for the MSE), the bias is set to zero.

$$Bias = \begin{cases} 0 & Bias^2 < 0 \\ \sqrt{Bias^{\,2}} & \text{otherwise} \end{cases}\ . \tag{11.10}$$

In this case, the confidence interval coverage is calculated as the fraction of denoised residuals contained in the confidence interval centered about zero. To use the confidence interval, several assumptions are made. The two most important assumptions for the discussion at hand are:

1. The model predictions are sensitive to the training data (i.e., the variance of the model predictions is nonzero).
2. The noise variance can be estimated to a sufficient degree of accuracy to produce the theoretical 95% coverage.

11.4 Redundant Model Case Studies

Recent redundant sensor model case studies have shown that the first assumption, that the predictions are sensitive to training data, is not met. This means that the redundant sensor model architecture is "fixed" beforehand and is only marginally affected by the data used to train the model. This causes model prediction variances that are often very near zero. For example, one case showed that the redundant sensor model variance is on the order of 100 times smaller than the variance of a nonredundant model, whose inputs are the same. The end result is that the model variance ($\mathrm{Var}(\hat{x})$) is very small and the sensitivity of the uncertainty prediction to the noise variance estimates increases. For nonredundant models, the prediction variance is large enough to "dampen" the effects of imprecision in the estimates of the noise variance, but this is not the case for redundant sensor architectures.

In nonredundant models the prediction variance is nearly equal to zero and the imprecision in the estimates of the noise variance begins to degrade the validity of the uncertainty estimates. The end result is unreasonably small redundant sensor model uncertainties, which produce coverages significantly below the theoretical 95%.

Example case

To demonstrate the conclusions presented in this section, data collected from three redundant Pressurizer-Level sensors of an operating nuclear power plant is analyzed. The training and test data are presented in Fig. 11-3 below. The training data were used to create the empirical models, while the test data was used to simulate the models.

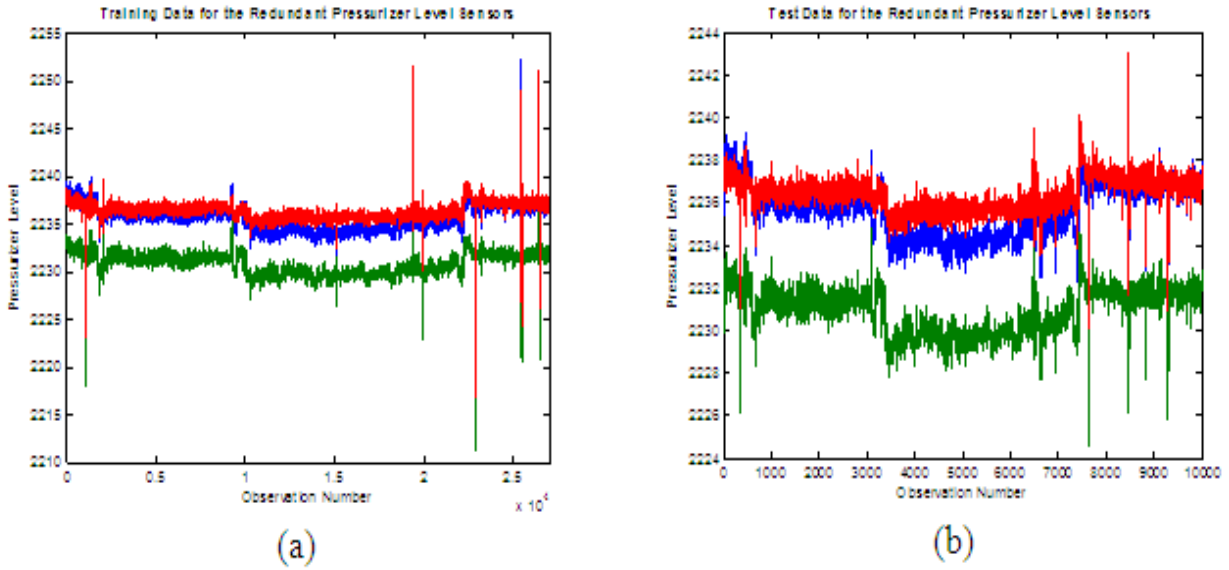


Fig. 11-3. (a) Training and (b) test data for the redundant Pressurizer-Level sensors.

The first step in this case study is to test the uncertainty estimation process presented earlier for redundant and nonredundant models. To do this, both a redundant and nonredundant model were trained with the training data. Next, the uncertainty was estimated for each model and their respective confidence interval coverages calculated. The results are presented in Table 11-1.

Table 11-1. Uncertainty and coverage results for nonredundant and redundant sensor models

		Sensor 1	Sensor 2	Sensor 3	Mean
Nonredundant	Variance	0.0738	0.0522	0.0867	0.0709
	Bias	0.0000	0.0000	0.0000	0.0000
	95% CI	0.5433	0.4570	0.5891	0.5298
	Coverage	0.9374	0.9723	0.8667	0.9255
Redundant	Variance	0.0003	0.0001	0.0003	0.0002
	Bias	0.1033	0.0000	0.2463	0.1165
	95% CI	0.2092	0.0225	0.4939	0.2419
	Coverage	0.4814	0.1951	0.8446	0.5070

As described earlier, the redundant sensor model variance is nearly equal to zero. Changing the data used to develop the model does not change the model predictions. This increases the predicted bias for two of the three sensors in the redundant sensor model. Finally, it can be seen that the resulting uncertainties for the redundant sensor model are significantly smaller than the uncertainties of the nonredundant sensor model. This difference results in coverages that are well below the theoretical 95% for the nonredundant sensor model.

To validate the recommendations of the previous section, the prediction interval (PI) was used for the redundant sensor model. The PI describes the uncertainty in the parameter estimates in terms of the measured values, whereas the confidence interval describes the uncertainty in terms of the expected or true value. The equation for the 95% prediction interval is given below:

$$U\left(\hat{\mathbf{x}}\right) = t_{n-p,\alpha/2}\sqrt{Var\left(\hat{\mathbf{x}}\right) + Bias^2 + \sigma_{\boldsymbol{\varepsilon}}^2} \quad , \tag{11.9}$$

Comparing Eq. (11.9) to Eq. (11.8), it is clear that the PI is, by definition, larger than the confidence interval and should be a valid uncertainty measure for redundant sensor models. The coverage in this case is calculated as the fraction of (not denoised) residuals contained in the PI centered about zero.

PIs are not used in nonredundant sensor models to cover the measured values; instead, confidence intervals are used to cover the residuals. As stated above, the confidence intervals result in smaller uncertainties because they do not contain the irreducable error.

The resulting uncertainties and coverages are presented in Table 11-2. The uncertainty is larger for the PI, and the mean coverage is near its theoretical 0.95.

Table 11-2. Confidence and prediction intervals and coverages of redundant sensor model

	Sensor 1	Sensor 2	Sensor 3	Mean
95% CI	0.2092	0.0225	0.4939	0.2419
CI coverage	0.4814	0.1951	0.8446	0.5070
95% PI	1.3143	0.2728	0.7154	0.7675
PI coverage	1.0000	0.9130	0.8580	0.9237

11.5 Correlation Analysis for Redundant Models

An additional concern may arise for some redundant models. Redundant models are usually comprised of fewer sensors and generally rely on the fact that all of the sensor inputs are highly correlated. However, under certain circumstances this high correlation is no longer present, causing many of the model assumptions to no longer be met, and the model accuracy and performance to degrade. When sensor data are highly steady state, as is the case with most nuclear power data, the independent random noise may become the dominating factor. Because the data are steady-state and the process is so stable, the only fluctuation between the redundant sensors is due to the noise. In this case, the correlation coefficients between the redundant sensors are low (often less than 0.5) because the independent noise is the prevalent component of each signal, causing each individual signal (or sensor) to be uncorrelated with the other signals. This low signal-to-noise ratio causes the model's performance to deteriorate because the model's inputs do not contain much information about the underlying process, and the model's predictions become less reliable. A way to combat this unconformity is to increase the sample size of the

data used to develop the model. With more observations, especially with observations gathered over a longer time period, it is likely that the dynamics of the process will overpower the noise, and the correlation between the sensors will increase. However, until this correlation is shown to improve, the model's predictions may not be trusted.

11.6 Recommendations

Because recent research with the redundant sensor models has indicated that the confidence interval may not be an appropriate uncertainty measure, alternatives must be considered. One such alternative is the PI that does not require the error to be smoothed and implements an additive relationship with the estimated noise variance, as opposed to the differential used by the confidence interval.

Additionally, before implementing a redundant model into an OLM system, a correlation analysis should be performed. As mentioned above, a correlation analysis will show if the model has a poor signal to noise ratio, which will degrade model performance. Poor correlations will usually be remedied by expanding the training data set.

12. CONCLUSIONS

This report investigates the effects of several assumptions and development parameters on auto-associative kernel regression models. The discussions included also provide some recommendations for identifying, preventing, or correcting these problems in applied OLM systems. The results of seven case studies have been summarized, which investigate the effects of model development and assumptions on model performance. Two case studies concern the effect of not meeting model assumptions: evaluating query data outside the training region and training with faulty data. Recommendations are given for identifying and correcting the problems caused by not meeting these important data assumptions. Both of these assumptions can be validated via inspection of the data. Data that may fall outside the training region can be identified through a two-step reliability metric, which first identifies query vectors outside the range of the training data and then characterizes the similarity to the training data of remaining query vectors. Faults within training data can be identified through either visual inspection or through analysis of the correlation coefficients if some prior knowledge of the training correlations is available. The third and fourth case studies investigate the effects of high noise levels on model performance and compare different methods of data denoising, respectively. The third case study outlines a method to determine the highest acceptable noise level for adequate OLM performance. For data sets with higher than acceptable noise levels, the effects of data denoising were investigated. The results of this study indicate that most well-applied data denoising methods produce equally acceptable results. A genetic algorithm optimization method was described for specifying the appropriate parameters for a specific data denoising technique. The remaining three case studies examine different features of model development by comparing vector selection methods, different amounts of memory vectors, and robust distance measures. Methodologies are outlined to determine the appropriate model development parameters for each of these cases. Finally, a section is included that highlights the special considerations needed for redundant sensor model architectures, including the use of prediction intervals instead of confidence intervals for estimating model uncertainty and the importance of correlation and signal to noise analysis for proper redundant model development.

Although this study is not an exhaustive review of the many issues in OLM system development, it provides a base set of considerations that should be examined and a method for testing these considerations with other model architectures. These common OLM data and modeling issues should be evaluated when developing and assessing an OLM system. Additionally, a method for identifying and correcting, or at least reducing the effects of each contingency, should be implemented before fielding an OLM system.

Page intentionally blank

13. REFERENCES

Atkeson, C. G., A. W. Moore, and S. Schaal (1997a), "Locally Weighted Learning," *Artificial Intelligence Review*, **11**, 11–73.

Atkeson, C. G., A. W. Moore, and S. Schaal (1997b), "Locally Weighted Learning for Control," *Artificial Intelligence R*eview, **11**, 75–113.

Cleveland, W. S., and C. Loader (1994a), *Computational Methods for Local Regression*, Technical Report 11, AT&T Bell Laboratories, Statistics Department, Murray Hill, NJ:

Cleveland, W. S., and C. Loader (1994b), *Smoothing by Local Regression: Principles and Methods*, Technical Report 95.3, AT&T Bell Laboratories, Statistics Department, Murray Hill, NJ:

Chung, C. Y., S. H. Leung, and Andrew Luk (1994), "An Evolutionary Search Algorithm for Adaptive IIR Equalizer," *Proc. ZEEE International Symposium on Circuits and Systems, London, UK, May 1994*, **2**, 53–56.

Diaz, I., A. B. Diez, and A. A. Cuadrado Vega (2001), "Complex Process Visualization Through Continuous Feature Maps Using Radial Basis Functions," *Proc. of the International Conference on Artificial Neural Networks, Vienna, Austria: August 21–25, 2001*.

EPRI (1993), TR-103436-V1, *Instrument Calibration and Monitoring Program, Volume 1: Basis for the Method*, EPRI, Palo Alto, CA.

EPRI (1995), *Monte Carlo Simulation and Uncertainty Analysis of the Instrument Calibration and Monitoring Program*, WO3785-02, EPRI, Palo Alto, CA.

EPRI (2000), *On-Line Monitoring of Instrument Channel Performance*, Topical Report 104965-R1, EPRI, Palo Alto, CA.

EPRI (2004), *On-Line Monitoring of Instrument Channel Performance Volume 1: Guidelines for Model Development and Implementation*, Final Report 1003361, Palo Alto, CA.

EPRI (2004), *On-Line Monitoring of Instrument Channel Performance Volume 2: Model Examples, Algorithm Descriptions, & Additional Results*, Topical Report 1003579, Palo Alto, CA.

Etter, D. M., M. J. Hicks, and K. H. Cho (1982), "Recursive Adaptive Filter Design using an Adaptive Genetic Algorithm," *Proc. IEEE Int. Conf on ASSP,* pp. 635–638.

Fan, J., and I. Gijbels (1996), *Local Polynomial Modeling and Its Applicati*ons, Chapman & Hall/CRC, New York.

Garvey, D. R., and J. W. Hines (2006), "Robust Distance Measures for On-Line Monitoring: Why Use Euclidean?", *7th International Conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science (FLINS), Genoa, Italy: August 29–31, 2006*.

Gross, K., S. Wegerich, R. Singer, and J. Mott (1998), "Industrial Process Surveillance System," U.S. Patent 5,764,509.

Hardle, W. (1989), *Applied Nonparametric Regression*, Cambridge University Press, New York.

Haupt, R. L., and S. E. Haupt (2004), *Practical Genetic Algorithms*, 2nd ed. Hoboken, John Wiley & Sons, Inc:

Hines, J. W., and B. Rasmussen (2005), "On-Line Sensor Calibration Monitoring Uncertainty Estimation," *Nuclear Technology*,**151**(3).

Hines, J.W. and E. Davis, "Lessons Learned From the U.S. Nuclear Power Plant On-Line Monitoring Programs," *Progress in Nuclear Energy*, **46**(3–4), 176–189.

Hines, J. W. and D. Garvey (2006), "Traditional and Robust Vector Selection Methods for Use with Similarity Based Models," *5th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies, Albuquerque, NM*.

Humenik, K. E. and K. C. Gross (1990), "Sequential Probability Ratio Tests for Reactor Signal Validation and Sensor Surveillance Applications," *Nucl. Sci. and Eng*., **105**, 383–390.

Leung S. H., C. Y. Chung, A. Luk and W. H. Lau (1996), "The Genetic Search Approach—A New Learning Algorithm for Adaptive IIR Filtering," *IEEE Signal Proc. Mag.*, Nov. 1996, pp. 38–46.

Mackenzie, M. and A. Kiet Tieu (2004), "Asymmetric Kernel Regression," *IEEE Transactions on Neural Networks*, **15**(2) (March 2004).

Miron, A. (2001), "A Wavelet Approach for Development and Application of a Stochastic Parameter Simulation System," Ph.D. Dissertation, University of Cincinnati, Cincinnati, OH.

Nadaraya E. A. (1964), "On Estimating Regression," *Theory of Probability and Its Applications*, **10**, 186–190 (1964).

Hines, J. W., and R. M. Seibert, (2006), *Technical Review of On-line Monitoring Techniques for Performance Assessment: Volume 1—State-of-the-Art*, NUREG/CR-6895, U.S. Nuclear Regulatory Commission, Washington, D.C.

NRC Project No. 669 (2000), "Safety Evaluation by the Office of Nuclear Reactor Regulation: Application of On-Line Performance Monitoring to Extend Calibration Intervals of Instrument Channel Calibrations Required by the Technical Specifications—EPRI Topical Report (TR) 104965 *On-Line Monitoring of Instrument Channel Performance*," U.S. Nuclear Regulatory Commission: Washington, D.C. (July, 2000).

Rasmussen, B., E. Davis, and J. W. Hines (2002), "Monte Carlo Analysis and Evaluation of the Instrumentation and Calibration Monitoring Program," *Proc. Maintenance and Reliability Conference (MARCON 2002), Knoxville, TN, May 7–10*.

Rasmussen, B. (2003), "Prediction Interval Estimation Techniques for Empirical Modeling Strategies and their Applications to Signal Validation Tasks," Ph.D. Dissertation, Nuclear Engineering Department, The University of Tennessee, Knoxville, TN.

Scott, D. W. (1992), *Multivariate Density Estimation*, Wiley, New York.

Terrell, T. J. (1988), *Introduction to Digital Filters*, Halsted Press/John Wiley & Sons, 2nd Ed.

Upadhyaya, B. R., O. Glockler, and F. P. Wolvaardt (1988), "An Integrated Approach for Signal Validation in Dynamic Systems," *Progress in Nuclear Energy*, **21**, 605–611.

Wand, M., and M. Jones (1995), *Kernel Smoothing*, Monographs on Statistics and Applied Probability, Chapman & Hall, London.

Watson, G. S. (1964), "Smooth Regression Analysis," *The Indian Journal of Statistics*, Series A, Vol. 26, pp. 359–372: 1964.

Wilson, D. R. and T. Martinez (2000), "Reduction Techniques for Exemplar-Based Learning Algorithms," *Machine Learning* **38**(3): 257–286.

Zavaljevski, N., and K. C. Gross (2000a), "Support Vector Machines for Nuclear Reactor State Estimation," *Proc. of the International Topical Meeting on Advances in Reactor Physics and Mathematics and Computation (PHYSOR 2000)*, Pittsburgh, Pennsylvania, USA.

Zavaljevski, N., and K. C. Gross (2000b), "Sensor Fault Detection in Nuclear Power Plants Using Multivariate State Estimation Technique and Support Vector Machines," *Proc. of the Third International Conference of the Yugoslav Nuclear Society (YUNSC 2000), Belgrade, Yugoslavia: October 2–5, 2000*.

Page intentionally blank

APPENDIX A

Because the results for each case study were so expansive, not all of them were necessarily displayed in the document body. Rather, for some case studies, only the significant results were shown. As such, this appendix shows the complete results for the case studies which only presented the condensed results.

Section A.1 of the appendix gives the uncondensed results for the baseline or "ideal" Pressurizer-Level model and RPS Loop A model. Section A.2 presents the results of the faulty training data case study. This includes the full results for both models trained with outliers, drifted data, and stuck data. Section A.3 provides the results for both models when they attempt to predict values for data that no longer fall within the training range. Section A.4 displays the results of the models using different vector selection methods. Section A.5 gives the results of the models developed with varying numbers of memory vectors. The final section of this appendix, A.6, presents the results of the models developed using the Robust Euclidean and the L-1 Norm distance metrics.

A.1 Baseline Model Results

Table A1-1. Baseline Pressurizer-Level model results

	Signal 1	**Signal 2**	**Signal 3**
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.055	0.038	0.102
Auto-sensitivity	0.466	0.712	0.434
Cross-sensitivity	0.301	0.274	0.355
EULM detectability (% of span)	0.200	0.248	0.357
SPRT detectability (% of span)	0.00112	0.00114	0.00108
Uncertainty			
Analytic (% of span)	0.107	0.071	0.203
Coverage	0.993	0.943	0.995
Monte Carlo (% of span)	0.121	0.092	0.209
Coverage	0.997	0.949	0.998

Table A1-2. Baseline RPS Loop A model results

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.261	0.246	0.290	0.229	0.919	0.120	0.027	0.037	0.046
Auto-sensitivity	0.420	0.451	0.620	0.714	0.160	0.082	0.405	0.370	0.433
Cross-sensitivity	0.120	0.119	0.098	0.116	0.079	0.046	0.130	0.125	0.150
EULM detectability (% of span)	1.440	1.480	2.340	3.130	2.210	0.255	0.073	0.107	0.146
SPRT detectability (% of span)	0.00276	0.0028	0.00308	0.00294	0.00166	0.00178	0.000449	0.000454	0.000454
Uncertainty									
Analytic (% of span)	0.840	0.818	0.897	0.899	1.840	0.235	0.043	0.067	0.083
Coverage	1.000	1.000	1.000	1.000	1.000	0.987	0.975	0.996	0.991
Monte Carlo (% of span)	0.550	0.520	0.554	0.544	1.860	0.299	0.075	0.095	0.104
Coverage	0.999	0.999	0.997	0.999	1.000	0.997	0.997	0.999	0.998

A.2 Faulted Training Data Results

Table A2-1. Results of Pressurizer-Level model trained with outliers

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.209	0.407	0.252
Correlation coefficients			
Signal 1	1.000	0.504	0.639
Signal 2	0.504	1.000	0.471
Signal 3	0.639	0.471	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.055	0.039	0.107
Auto-sensitivity	0.481	0.697	0.411
Cross-sensitivity	0.362	0.240	0.370
EULM detectability (% of span)	0.707	2.360	0.748
SPRT detectability (% of span)	0.0204	0.0428	0.0212
Uncertainty			
Analytic (% of span)	0.367	0.715	0.442
Coverage	1.000	1.000	1.000
Monte Carlo (% of span)	0.068	0.072	0.068
Coverage	0.749	0.952	0.258

Table A2-2. Results of RPS Loop A model trained with outliers

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.519	0.486	0.634	0.590	0.430	0.416	0.341	0.338	0.027
Correlation coefficients									
Signal 1	1.000	0.959	0.669	0.684	0.742	0.741	–0.096	–0.080	–0.269
Signal 2	0.959	1.000	0.677	0.690	0.753	0.747	–0.091	–0.070	–0.264
Signal 3	0.669	0.677	1.000	0.627	0.693	0.688	–0.122	–0.100	–0.314
Signal 4	0.684	0.690	0.627	1.000	0.708	0.701	–0.123	–0.107	–0.362
Signal 5	0.742	0.753	0.693	0.708	1.000	0.833	–0.103	–0.080	–0.278
Signal 6	0.741	0.747	0.688	0.701	0.833	1.000	–0.106	–0.089	–0.272
Signal 7	–0.096	–0.091	–0.122	–0.123	–0.103	–0.106	1.000	0.213	0.445
Signal 8	–0.080	–0.070	–0.100	–0.107	–0.080	–0.089	0.213	1.000	0.447
Signal 9	–0.269	–0.264	–0.314	–0.362	–0.278	–0.272	0.445	0.447	1.000
Model: aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics:									
Accuracy (% of span)	0.255	0.252	0.235	0.238	0.946	0.120	0.038	0.055	0.033
Auto-sensitivity	0.452	0.435	0.674	0.657	0.168	0.080	0.209	0.178	0.576
Cross-sensitivity	0.097	0.100	0.079	0.086	0.062	0.038	0.087	0.088	0.128
EULM detectability (% of span)	1.420	1.290	2.920	2.580	2.160	0.681	0.651	0.621	0.123
SPRT detectability (% of span)	1.490	1.370	2.410	1.910	2.510	2.270	1.600	1.700	0.233
Uncertainty:									
Analytic (% of span)	0.784	0.733	0.957	0.891	1.780	0.628	0.514	0.510	0.052
Coverage	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.977
Monte Carlo (% of span)	0.474	0.447	0.515	0.535	1.680	0.225	0.097	0.100	0.086
Coverage	0.999	1.000	0.997	1.000	1.000	0.973	0.997	0.985	0.999

Table A2-3. Results of Pressurizer-Level model trained with drifted data

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.637
Signal 2	0.987	1.000	0.638
Signal 3	0.637	0.638	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.032	0.030	0.078
Auto-sensitivity	0.539	0.538	0.438
Cross-sensitivity	0.261	0.305	0.294
EULM detectability (% of span)	0.122	0.114	0.274
SPRT detectability (% of span)	0.00289	0.00274	0.00336
Uncertainty			
Analytic (% of span)	0.056	0.053	0.153
Coverage	0.922	0.942	0.939
Monte Carlo (% of span)	0.060	0.060	0.162
Coverage	0.943	0.954	0.940

Table A2-4. Results of RPS-Loop A model trained with drifted data

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.324	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.288	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.273	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.324	0.288	0.273	1.000	0.316	0.404	–0.266	–0.338	–0.206
Signal 5	0.820	0.824	0.797	0.316	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.404	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.266	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.338	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.206	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.278	0.245	0.253	0.276	1.020	0.114	0.029	0.050	0.054
Auto-sensitivity	0.425	0.472	0.718	0.698	0.051	0.080	0.365	0.353	0.265
Cross-sensitivity	0.094	0.097	0.096	0.106	0.041	0.047	0.119	0.120	0.103
EULM detectability (% of span)	1.410	1.490	3.060	2.900	2.160	0.242	0.076	0.147	0.136
SPRT detectability (% of span)	0.00219	0.00217	0.00225	0.00249	0.00124	0.00127	0.000343	0.000339	0.00034
Uncertainty									
Analytic (% of span)	0.813	0.792	0.868	0.871	2.030	0.223	0.048	0.095	0.100
Coverage	1.000	1.000	1.000	0.996	1.000	0.979	0.985	0.997	0.997
Monte Carlo (% of span)	0.552	0.529	0.576	0.677	2.050	0.315	0.079	0.115	0.112
Coverage	0.998	0.999	0.998	0.994	1.000	0.999	0.998	0.999	0.999

Table A2-5. Results of Pressurizer-Level model trained with data with a signal stuck at the maximum

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.015
Correlation coefficients			
Signal 1	1.000	0.987	0.627
Signal 2	0.987	1.000	0.612
Signal 3	0.627	0.612	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	1	1	1
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.111	0.049	0.080
Auto-sensitivity	0.177	0.149	0.633
Cross-sensitivity	0.226	0.214	0.272
EULM detectability (% of span)	0.264	0.104	0.422
SPRT detectability (% of span)	0.00218	0.00226	0.00084
Uncertainty			
Analytic (% of span)	0.217	0.089	0.157
Coverage	1.000	0.967	0.993
Monte Carlo (% of span)	0.332	0.260	0.298
Coverage	1.000	1.000	1.000

Table A2-6. Results of Pressurizer-Level model trained with data with a signal stuck at the mean

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.020
Correlation coefficients			
Signal 1	1.000	0.987	0.974
Signal 2	0.987	1.000	0.960
Signal 3	0.974	0.960	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.061	0.033	0.095
Auto-sensitivity	0.655	0.554	0.519
Cross-sensitivity	0.356	0.306	0.393
EULM detectability (% of span)	0.342	0.134	0.391
SPRT detectability (% of span)	0.00111	0.00114	0.00106
Uncertainty			
Analytic (% of span)	0.118	0.060	0.188
Coverage	0.997	0.941	0.989
Monte Carlo (% of span)	0.138	0.100	0.201
Coverage	0.998	0.957	0.994

Table A2-7. Results of Pressurizer-Level model trained with data with a signal stuck at the minimum

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.016
Correlation coefficients			
Signal 1	1.000	0.987	0.857
Signal 2	0.987	1.000	0.877
Signal 3	0.857	0.877	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.069	0.051	0.132
Auto-sensitivity	0.506	0.538	0.234
Cross-sensitivity	0.440	0.413	0.596
EULM detectability (% of span)	0.273	0.211	0.348
SPRT detectability (% of span)	0.000798	0.000827	0.000632
Uncertainty			
Analytic (% of span)	0.135	0.097	0.264
Coverage	0.997	0.994	0.998
Monte Carlo (% of span)	0.188	0.161	0.300
Coverage	1.000	1.000	0.999

Table A2-8. Results of RPS Loop A model trained with data with a signal stuck at the maximum

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.019	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	0.291	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	0.281	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	0.292	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	0.293	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	0.242	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	0.210	–0.216	–0.297
Signal 7	0.291	0.281	0.292	0.293	0.242	0.210	1.000	–0.022	–0.116
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	–0.022	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	–0.116	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.269	0.293	0.301	0.335	0.874	0.151	0.048	0.056	0.060
Auto-sensitivity	0.403	0.408	0.612	0.707	0.183	0.211	0.019	0.340	0.366
Cross-sensitivity	0.120	0.129	0.119	0.140	0.110	0.080	0.036	0.142	0.112
EULM detectability (% of span)	1.460	1.440	2.400	3.190	2.160	0.375	0.094	0.160	0.175
SPRT detectability (% of span)	0.0035	0.00346	0.00375	0.00383	0.00219	0.00211	0.000908	0.000556	0.000558
Uncertainty									
Analytic (% of span)	0.877	0.854	0.936	0.939	1.750	0.297	0.091	0.105	0.111
Coverage	1.000	1.000	1.000	1.000	1.000	0.983	0.978	0.978	0.982
Monte Carlo (% of span)	0.634	0.648	0.679	0.680	1.790	0.440	0.131	0.143	0.134
Coverage	0.999	0.999	0.999	1.000	1.000	0.996	0.994	0.995	0.995

Table A2-9. Results of RPS Loop A model trained with data with a signal stuck at the mean

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.017	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.287	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.276	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.318	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.317	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.376	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.386	–0.216	–0.297
Signal 7	–0.287	–0.276	–0.318	–0.317	–0.376	–0.386	1.000	0.650	0.684
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.650	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.684	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.303	0.316	0.298	0.344	0.836	0.134	0.041	0.056	0.054
Auto-sensitivity	0.368	0.297	0.554	0.468	0.091	0.043	0.268	0.223	0.543
Cross-sensitivity	0.111	0.098	0.112	0.102	0.073	0.045	0.141	0.086	0.119
EULM detectability (% of span)	1.490	1.310	2.250	1.890	1.860	0.274	0.108	0.139	0.219
SPRT detectability (% of span)	0.00379	0.00378	0.00402	0.00436	0.0021	0.00228	0.000434	0.000601	0.000551
Uncertainty									
Analytic (% of span)	0.946	0.921	1.010	1.010	1.670	0.263	0.079	0.108	0.100
Coverage	1.000	0.999	1.000	1.000	1.000	0.985	0.957	0.978	0.972
Monte Carlo (% of span)	0.656	0.655	0.734	0.726	1.710	0.485	0.104	0.133	0.131
Coverage	0.999	0.997	0.999	0.998	1.000	0.998	0.995	0.999	0.999

Table A2-10. Results of RPS Loop A model trained with data with a signal stuck at the minimum

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.015	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.083	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.052	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.081	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.086	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.127	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.164	–0.216	–0.297
Signal 7	–0.083	–0.052	–0.081	–0.086	–0.127	–0.164	1.000	0.411	0.319
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.411	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.319	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.276	0.287	0.281	0.334	0.868	0.121	0.055	0.054	0.052
Auto-sensitivity	0.460	0.434	0.621	0.451	0.128	0.040	0.040	0.195	0.487
Cross-sensitivity	0.106	0.104	0.113	0.091	0.075	0.033	0.028	0.092	0.098
EULM detectability (% of span)	1.690	1.570	2.570	1.780	2.010	0.246	0.111	0.127	0.187
SPRT detectability (% of span)	0.00274	0.0026	0.00308	0.00314	0.00152	0.00161	0.000678	0.000498	0.000491
Uncertainty									
Analytic (% of span)	0.920	0.895	0.981	0.984	1.740	0.236	0.107	0.102	0.096
Coverage	1.000	1.000	1.000	1.000	1.000	0.978	0.986	0.974	0.968
Monte Carlo (% of span)	0.611	0.565	0.636	0.601	1.770	0.313	0.131	0.129	0.125
Coverage	0.999	0.999	0.998	0.996	1.000	0.998	0.994	0.993	0.996

Table A2-11. Results of Pressurizer-Level model trained with drifted data

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.637
Signal 2	0.987	1.000	0.638
Signal 3	0.637	0.638	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.032	0.030	0.078
Auto-sensitivity	0.539	0.538	0.438
Cross-sensitivity	0.261	0.305	0.294
EULM detectability (% of span)	0.122	0.114	0.274
SPRT detectability (% of span)	0.00289	0.00274	0.00336
Uncertainty			
Analytic (% of span)	0.056	0.053	0.153
Coverage	0.922	0.942	0.939
Monte Carlo (% of span)	0.060	0.060	0.162
Coverage	0.943	0.954	0.940

Table A2-12. Results of RPS Loop A model trained with drifted data

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.324	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.288	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.273	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.324	0.288	0.273	1.000	0.316	0.404	–0.266	–0.338	–0.206
Signal 5	0.820	0.824	0.797	0.316	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.404	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.266	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.338	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.206	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.278	0.245	0.253	0.276	1.020	0.114	0.029	0.050	0.054
Auto-sensitivity	0.425	0.472	0.718	0.698	0.051	0.080	0.365	0.353	0.265
Cross-sensitivity	0.094	0.097	0.096	0.106	0.041	0.047	0.119	0.120	0.103
EULM detectability (% of span)	1.410	1.490	3.060	2.900	2.160	0.242	0.076	0.147	0.136
SPRT detectability (% of span)	0.00219	0.00217	0.00225	0.00249	0.00124	0.00127	0.000343	0.000339	0.00034
Uncertainty									
Analytic (% of span)	0.813	0.792	0.868	0.871	2.030	0.223	0.048	0.095	0.100
Coverage	1.000	1.000	1.000	0.996	1.000	0.979	0.985	0.997	0.997
Monte Carlo (% of span)	0.552	0.529	0.576	0.677	2.050	0.315	0.079	0.115	0.112
Coverage	0.998	0.999	0.998	0.994	1.000	0.999	0.998	0.999	0.999

A.3 Model Results with Data Outside the Training Range

Table A3-1. Results of Pressurizer-Level model trained with query data outside the training region

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	15,000	15,000	15,000
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	12.4	12.2	12.2
Auto-sensitivity	0.003	0.004	0.011
Cross-sensitivity	0.007	0.006	0.003
EULM detectability (% of span)	30.2	29.8	29.8
SPRT detectability (% of span)	8.04E-06	8.22E-06	7.78E-06
Uncertainty			
Analytic (% of span)	24.7	24.5	24.4
Coverage	0.946	0.947	0.947
Monte Carlo (% of span)	24.7	24.5	24.4
Coverage	0.947	0.947	0.947

Table A3-2. Results of RPS Loop A model with query data outside of the training region

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	15,000	15,000	15,000	15,000	15,000	15,000	15,000	15,000	15,000
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	28.0	27.5	26.9	27.0	27.4	27.3	1.06	1.04	1.05
Auto-sensitivity	0.001	0.001	0.006	0.001	0.000	0.000	0.154	0.146	0.171
Cross-sensitivity	0.060	0.061	0.061	0.060	0.060	0.064	0.034	0.035	0.033
EULM detectability (% of span)	75.7	73.6	72.7	72.7	73.5	72.7	2.48	2.40	2.50
SPRT detectability (% of span)	0.000103	0.000104	0.000115	0.00011	6.19E-05	6.62E-05	1.67E-05	1.69E-05	1.69E-05
Uncertainty									
Analytic (% of span)	56.0	55.0	53.8	53.9	54.7	54.5	2.12	2.08	2.10
Coverage	0.933	0.933	0.933	0.933	0.933	0.933	0.880	0.878	0.886
Monte Carlo (% of span)	56.1	55.0	53.9	54.0	54.8	54.6	2.19	2.15	2.17
Coverage	0.933	0.933	0.933	0.933	0.933	0.933	0.894	0.887	0.902

A.4 Model Results for Vector Selection Methods

Table A4-1. Results of Pressurizer-Level model using min-max vector selection method

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	m	m	m
Metrics			
Accuracy (% of span)	0.001	0.001	0.001
Auto-sensitivity	0.479	0.820	0.487
Cross-sensitivity	0.291	0.297	0.379
EULM detectability (% of span)	0.203	0.424	0.320
SPRT detectability (% of span)	0.00041	0.000527	0.000432
Uncertainty			
Analytic (% of span)	0.106	0.076	0.164
Coverage	0.985	0.961	0.978
Monte Carlo (% of span)	0.389	0.350	0.474
Coverage	1.000	1.000	1.000

Table A4-2. Results of Pressurizer-Level model using sort-select vector selection method

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	s	s	s
Metrics			
Accuracy (% of span)	0.001	0.001	0.002
Auto-sensitivity	0.466	0.717	0.432
Cross-sensitivity	0.301	0.275	0.355
EULM detectability (% of span)	0.198	0.225	0.358
SPRT detectability (% of span)	0.000512	0.00052	0.000493
Uncertainty			
Analytic (% of span)	0.106	0.064	0.203
Coverage	0.995	0.947	0.994
Monte Carlo (% of span)	0.114	0.081	0.208
Coverage	0.996	0.960	0.996

Table A4-3. Results of RPS Loop A model using min-max vector selection method

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	m	m	m	m	m	m	m	m	m
Metrics									
Accuracy (% of span)	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
Auto–sensitivity	0.487	0.399	0.728	0.732	0.391	0.273	0.468	0.360	0.404
Cross–sensitivity	0.154	0.145	0.153	0.136	0.135	0.131	0.153	0.141	0.151
EULM detectability (% of span)	1.590	1.320	3.190	3.260	3.050	0.544	0.085	0.115	0.175
SPRT detectability (% of span)	0.00321	0.00316	0.00352	0.00373	0.00243	0.00252	0.00049	0.000508	0.000501
Uncertainty									
Analytic (% of span)	0.816	0.794	0.871	0.873	1.860	0.396	0.045	0.074	0.104
Coverage	1.000	1.000	1.000	0.999	1.000	0.997	0.954	0.996	0.999
Monte Carlo (% of span)	1.060	1.120	1.290	1.350	1.890	0.484	0.122	0.143	0.154
Coverage	1.000	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000

Table A4-4. Results of RPS Loop A model using sort-select vector selection method

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	s	s	s	s	s	s	s	s	s
Metrics									
Accuracy (% of span)	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
Auto–sensitivity	0.412	0.468	0.652	0.610	0.131	0.074	0.390	0.359	0.383
Cross–sensitivity	0.105	0.110	0.108	0.092	0.065	0.043	0.114	0.119	0.130
EULM detectability (% of span)	1.390	1.490	2.500	2.240	2.160	0.251	0.072	0.112	0.126
SPRT detectability (% of span)	0.00273	0.00261	0.0031	0.00307	0.0019	0.00186	0.00046	0.00049	0.000474
Uncertainty									
Analytic (% of span)	0.816	0.794	0.871	0.873	1.880	0.232	0.044	0.072	0.077
Coverage	1.000	1.000	1.000	0.999	1.000	0.988	0.970	0.992	0.987
Monte Carlo (% of span)	0.538	0.544	0.558	0.561	1.900	0.301	0.080	0.102	0.103
Coverage	0.999	1.000	0.999	0.996	1.000	0.998	0.998	1.000	0.997

A.5 Model Results for Number of Memory Vectors

Table A5-1. Results of Pressurizer-Level model using ten memory vectors

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	10	10	10
Optimal kernel width	0.5	0.5	0.5
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.004	0.004	0.003
Auto–sensitivity	0.171	0.068	0.419
Cross–sensitivity	0.136	0.116	0.176
EULM detectability (% of span)	0.586	0.585	0.663
SPRT detectability (% of span)	0.036	0.036	0.036
Uncertainty			
Analytic (% of span)	0.485	0.545	0.385
Coverage	0.942	0.943	0.943
Monte Carlo (% of span)	0.500	0.557	0.408
Coverage	0.943	0.944	0.944

Table A5-2. Results of Pressurizer-Level model using 25 memory vectors

	Signal 1	Signal 2	Signal 3
Data:			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	25	25	25
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.003	0.003	0.003
Auto–sensitivity	0.294	0.448	0.377
Cross–sensitivity	0.248	0.219	0.241
EULM detectability (% of span)	0.574	0.753	0.549
SPRT detectability (% of span)	0.060	0.061	0.060
Uncertainty			
Analytic (% of span)	0.405	0.416	0.342
Coverage	0.943	0.944	0.944
Monte Carlo (% of span)	0.435	0.459	0.384
Coverage	0.944	0.945	0.946

Table A5-3. Results of Pressurizer-Level model using 50 memory vectors

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	50	50	50
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.003	0.003	0.003
Auto–sensitivity	0.417	0.528	0.480
Cross–sensitivity	0.326	0.240	0.360
EULM detectability (% of span)	0.560	0.745	0.621
SPRT detectability (% of span)	0.00651	0.00677	0.00642
Uncertainty			
Analytic (% of span)	0.326	0.352	0.323
Coverage	0.962	0.962	0.945
Monte Carlo (% of span)	0.418	0.467	0.419
Coverage	0.966	0.966	0.951

Table A5-4. Results of Pressurizer-Level model using 100 memory vectors

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	100	100	100
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.001	0.001	0.002
Auto–sensitivity	0.454	0.705	0.393
Cross–sensitivity	0.321	0.296	0.381
EULM detectability (% of span)	0.222	0.344	0.410
SPRT detectability (% of span)	0.0013	0.00127	0.00131
Uncertainty			
Analytic (% of span)	0.121	0.101	0.248
Coverage	0.972	0.968	0.955
Monte Carlo (% of span)	0.164	0.158	0.272
Coverage	0.986	0.985	0.959

Table A5-5. Results of Pressurizer-Level model using 3750 memory vectors

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	3750	3750	3750
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.001	0.001	0.002
Auto–sensitivity	0.443	0.701	0.456
Cross–sensitivity	0.303	0.288	0.333
EULM detectability (% of span)	0.169	0.200	0.344
SPRT detectability (% of span)	0.00031	0.000314	0.000291
Uncertainty			
Analytic (% of span)	0.094	0.060	0.187
Coverage	0.989	0.944	0.996
Monte Carlo (% of span)	0.095	0.061	0.187
Coverage	0.990	0.944	0.996

Table A5-6. Results of Pressurizer-Level model using 7000 memory vectors

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	7500	7500	7500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.001	0.001	0.002
Auto–sensitivity	0.439	0.735	0.458
Cross–sensitivity	0.305	0.285	0.340
EULM detectability (% of span)	0.167	0.201	0.340
SPRT detectability (% of span)	0.000172	0.000181	0.000153
Uncertainty			
Analytic (% of span)	0.094	0.053	0.184
Coverage	0.990	0.944	0.996
Monte Carlo (% of span)	0.093	0.053	0.184
Coverage	0.989	0.944	0.996

Table A5-7. Results of Pressurizer-Level model using 15,000 memory vectors

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	15000	15000	15000
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.001	0.001	0.002
Auto–sensitivity	0.435	0.738	0.460
Cross–sensitivity	0.292	0.285	0.339
EULM detectability (% of span)	0.164	0.203	0.338
SPRT detectability (% of span)	0.000167	0.000178	0.000147
Uncertainty			
Analytic (% of span)	0.093	0.053	0.182
Coverage	0.988	0.944	0.998
Monte Carlo (% of span)	0.092	0.053	0.182
Coverage	0.987	0.944	0.998

Table A5-8. Results of RPS Loop A model using ten memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	10	10	10	10	10	10	10	10	10
Optimal kernel width	2	2	2	2	2	2	2	2	2
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.00017	0.00016	0.00032	0.00023	0.00123	0.00030	0.00009	0.00010	0.00008
Auto–sensitivity	0.103	0.072	1.320	0.539	0.016	0.003	0.092	0.106	0.014
Cross–sensitivity	0.082	0.036	0.247	0.213	0.042	0.004	0.097	0.104	0.022
EULM detectability (% of span)	1.270	1.200	–7.880	3.640	1.810	0.435	0.197	0.212	0.159
SPRT detectability (% of span)	0.0439	0.0423	0.0628	0.0736	0.0476	0.043	0.00818	0.00815	0.00852
Uncertainty									
Analytic (% of span)	1.140	1.110	2.490	1.680	1.780	0.434	0.179	0.190	0.157
Coverage	0.992	0.990	1.000	0.980	0.992	0.988	0.989	0.993	0.950
Monte Carlo (% of span)	0.890	1.000	2.910	1.780	1.810	0.459	0.266	0.301	0.227
Coverage	0.933	0.975	1.000	0.986	0.993	0.988	1.000	1.000	1.000

Table A5-9. Results of RPS Loop A model using 25 memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	25	25	25	25	25	25	25	25	25
Optimal kernel width	1	1	1	1	1	1	1	1	1
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.000112	0.000127	0.000117	0.000134	0.00145	0.000273	4.77E–05	4.97E–05	5.00E–05
Auto–sensitivity	0.261	0.156	0.245	0.094	0.017	0.013	0.309	0.355	0.237
Cross–sensitivity	0.090	0.075	0.076	0.068	0.023	0.022	0.130	0.135	0.117
EULM detectability (% of span)	1.140	0.968	1.190	0.991	2.140	0.402	0.123	0.140	0.114
SPRT detectability (% of span)	2.270	2.330	2.830	3.390	2.370	2.210	0.249	0.233	0.289
Uncertainty									
Analytic (% of span)	0.839	0.817	0.895	0.898	2.110	0.397	0.085	0.090	0.087
Coverage	0.991	0.988	0.991	0.988	0.995	0.988	0.990	0.980	0.992
Monte Carlo (% of span)	0.669	0.700	0.950	0.810	2.130	0.515	0.150	0.149	0.129
Coverage	0.987	0.978	0.992	0.982	0.995	0.989	0.995	0.996	0.995

Table A5-10. Results of RPS Loop A model using 50 memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	50	50	50	50	50	50	50	50	50
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	9.32E–05	8.40E–05	9.85E–05	0.000102	0.00143	0.000193	4.44E–05	5.24E–05	6.18E–05
Auto–sensitivity	0.300	0.412	0.559	0.524	0.065	0.069	0.330	0.343	0.253
Cross–sensitivity	0.147	0.168	0.171	0.136	0.068	0.061	0.138	0.147	0.140
EULM detectability (% of span)	1.410	1.640	2.390	2.220	2.220	0.298	0.119	0.147	0.152
SPRT detectability (% of span)	0.00576	0.00516	0.00678	0.00659	0.00319	0.00347	0.000968	0.00097	0.00102
Uncertainty									
Analytic (% of span)	0.988	0.962	1.050	1.060	2.070	0.278	0.079	0.097	0.114
Coverage	0.999	1.000	0.999	0.999	1.000	0.985	0.993	0.990	0.991
Monte Carlo (% of span)	0.804	0.787	0.856	0.839	2.100	0.416	0.139	0.151	0.146
Coverage	0.999	0.999	0.997	0.999	1.000	0.999	0.995	0.993	0.994

Table A5-11. Results of RPS Loop A model using 100 memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	100	100	100	100	100	100	100	100	100
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	9.72E–05	9.10E–05	7.62E–05	7.32E–05	0.00147	0.000213	3.90E–05	5.73E–05	5.45E–05
Auto–sensitivity	0.304	0.374	0.661	0.592	0.039	0.054	0.385	0.325	0.216
Cross–sensitivity	0.124	0.122	0.120	0.133	0.040	0.050	0.125	0.133	0.113
EULM detectability (% of span)	1.330	1.440	2.930	2.440	2.210	0.326	0.112	0.160	0.127
SPRT detectability (% of span)	0.00526	0.0053	0.00534	0.00535	0.00293	0.00307	0.000809	0.000844	0.000805
Uncertainty									
Analytic (% of span)	0.929	0.905	0.992	0.995	2.130	0.308	0.069	0.108	0.100
Coverage	0.999	1.000	0.999	0.999	1.000	0.988	0.990	0.993	0.992
Monte Carlo (% of span)	0.700	0.683	0.772	0.746	2.150	0.420	0.118	0.146	0.134
Coverage	0.999	0.999	0.998	0.997	1.000	0.993	0.993	0.994	0.994

Table A5-12. Results of RPS Loop A model using 10 memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	3750	3750	3750	3750	3750	3750	3750	3750	3750
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	4.43E–05	4.18E–05	5.14E–05	5.06E–05	0.0012	0.000154	2.27E–05	3.10E–05	3.50E–05
Auto–sensitivity	0.456	0.472	0.672	0.667	0.232	0.093	0.391	0.360	0.492
Cross–sensitivity	0.085	0.089	0.071	0.066	0.077	0.043	0.102	0.107	0.119
EULM detectability (% of span)	1.110	1.110	1.960	1.940	2.260	0.241	0.045	0.076	0.102
SPRT detectability (% of span)	0.00159	0.00155	0.00176	0.00176	0.00103	0.00104	0.000215	0.000218	0.000233
Uncertainty									
Analytic (% of span)	0.602	0.586	0.643	0.645	1.740	0.219	0.028	0.049	0.052
Coverage	1.000	1.000	1.000	1.000	1.000	0.979	0.850	0.976	0.891
Monte Carlo (% of span)	0.304	0.298	0.337	0.359	1.750	0.245	0.038	0.058	0.061
Coverage	0.995	0.997	0.996	0.997	1.000	0.991	0.963	0.997	0.956

Table A5-13. Results of RPS Loop A model using 7500 memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	7500	7500	7500	7500	7500	7500	7500	7500	7500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	4.40E–05	4.22E–05	4.64E–05	4.78E–05	0.00121	0.000149	2.11E–05	3.09E–05	3.45E–05
Auto–sensitivity	0.456	0.461	0.692	0.680	0.219	0.085	0.406	0.369	0.495
Cross–sensitivity	0.074	0.076	0.056	0.053	0.064	0.037	0.097	0.101	0.116
EULM detectability (% of span)	0.935	0.918	1.760	1.700	2.250	0.230	0.039	0.074	0.095
SPRT detectability (% of span)	0.00149	0.00143	0.00167	0.00165	0.00095	0.000951	0.000181	0.000179	0.000198
Uncertainty									
Analytic (% of span)	0.509	0.495	0.543	0.545	1.760	0.210	0.023	0.047	0.048
Coverage	1.000	1.000	1.000	1.000	1.000	0.975	0.796	0.966	0.866
Monte Carlo (% of span)	0.235	0.218	0.261	0.265	1.770	0.228	0.029	0.052	0.052
Coverage	0.979	0.985	0.991	0.988	1.000	0.985	0.890	0.989	0.892

Table A5-14. Results of RPS Loop A model using 10 memory vectors

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	15000	15000	15000	15000	15000	15000	15000	15000	15000
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	4.05E–05	3.87E–05	4.69E–05	4.43E–05	0.0012	0.000145	2.14E–05	3.05E–05	3.34E–05
Auto–sensitivity	0.459	0.468	0.675	0.691	0.207	0.088	0.400	0.351	0.510
Cross–sensitivity	0.064	0.067	0.042	0.043	0.059	0.031	0.091	0.092	0.108
EULM detectability (% of span)	0.760	0.754	1.350	1.430	2.190	0.222	0.032	0.068	0.086
SPRT detectability (% of span)	0.00143	0.00138	0.00161	0.00159	0.000913	0.000917	0.000172	0.000171	0.000188
Uncertainty									
Analytic (% of span)	0.412	0.401	0.439	0.441	1.740	0.203	0.019	0.044	0.042
Coverage	1.000	1.000	1.000	1.000	1.000	0.969	0.676	0.935	0.818
Monte Carlo (% of span)	0.170	0.153	0.189	0.195	1.740	0.212	0.021	0.046	0.043
Coverage	0.922	0.927	0.920	0.942	1.000	0.978	0.719	0.959	0.825

A.6 Results for Models Using Various Distance Calculation Methods

Table A6-1. Results for Pressurizer-Level model using L-1 norm distance metric

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.052	0.026	0.113
Auto–sensitivity	0.418	0.723	0.348
Cross–sensitivity	0.240	0.275	0.352
EULM detectability (% of span)	0.173	0.163	0.342
SPRT detectability (% of span)	0.00119	0.0012	0.00117
Uncertainty			
Analytic (% of span)	0.101	0.045	0.224
Coverage	0.901	0.948	0.958
Monte Carlo (% of span)	0.113	0.077	0.236
Coverage	0.947	0.987	0.966

Table A6-2. Results for Pressurizer-Level model using Robust Eucledian distance metric

	Signal 1	Signal 2	Signal 3
Data			
Signal noise estimate (% of span)	0.024	0.023	0.023
Correlation coefficients			
Signal 1	1.000	0.987	0.994
Signal 2	0.987	1.000	0.981
Signal 3	0.994	0.981	1.000
Model aakr			
Training range	30,000	30,000	30,000
Test range	22,501	22,501	22,501
Validation range	28,500	28,500	28,500
Data cleaning	no	no	no
Number of memory vectors	500	500	500
Optimal kernel width	0.2	0.2	0.2
Vector selection method	x	x	x
Metrics			
Accuracy (% of span)	0.043	0.019	0.168
Auto–sensitivity	0.218	0.264	0.068
Cross–sensitivity	0.099	0.075	0.245
EULM detectability (% of span)	0.103	0.045	0.357
SPRT detectability (% of span)	0.000571	0.000777	0.000573
Uncertainty			
Analytic (% of span)	0.081	0.033	0.334
Coverage	0.938	0.954	1.000
Monte Carlo (% of span)	0.081	0.038	0.335
Coverage	0.938	0.968	1.000

Table A6-3. Results for RPS Loop A model using L-1 norm metric

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	1	1	1	1	1	1	1	1	1
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.272	0.260	0.298	0.252	0.972	0.117	0.026	0.041	0.047
Auto–sensitivity	0.417	0.445	0.645	0.727	0.032	0.171	0.454	0.271	0.411
Cross–sensitivity	0.132	0.133	0.134	0.138	0.099	0.056	0.117	0.143	0.141
EULM detectability (% of span)	1.610	1.650	2.820	3.670	2.020	0.275	0.080	0.103	0.146
SPRT detectability (% of span)	0.00235	0.00236	0.0028	0.00269	0.00133	0.00135	0.000404	0.000405	0.000401
Uncertainty									
Analytic (% of span)	0.943	0.918	1.010	1.010	1.940	0.229	0.043	0.075	0.086
Coverage	1.000	1.000	1.000	1.000	1.000	0.989	0.981	0.996	0.983
Monte Carlo (% of span)	0.600	0.575	0.676	0.658	1.970	0.334	0.072	0.100	0.104
Coverage	1.000	1.000	0.998	1.000	1.000	0.997	0.998	1.000	0.997

Table A6-4. Results for RPS Loop A model using Robust Eucledian distance metric

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6	Signal 7	Signal 8	Signal 9
Data									
Signal noise estimate (% of span)	0.496	0.483	0.530	0.531	0.035	0.035	0.023	0.022	0.027
Correlation coefficients									
Signal 1	1.000	0.970	0.716	0.711	0.820	0.815	–0.212	–0.183	–0.273
Signal 2	0.970	1.000	0.716	0.711	0.824	0.815	–0.199	–0.161	–0.264
Signal 3	0.716	0.716	1.000	0.681	0.797	0.788	–0.280	–0.239	–0.328
Signal 4	0.711	0.711	0.681	1.000	0.797	0.789	–0.279	–0.242	–0.372
Signal 5	0.820	0.824	0.797	0.797	1.000	0.990	–0.244	–0.195	–0.302
Signal 6	0.815	0.815	0.788	0.789	0.990	1.000	–0.253	–0.216	–0.297
Signal 7	–0.212	–0.199	–0.280	–0.279	–0.244	–0.253	1.000	0.970	0.956
Signal 8	–0.183	–0.161	–0.239	–0.242	–0.195	–0.216	0.970	1.000	0.947
Signal 9	–0.273	–0.264	–0.328	–0.372	–0.302	–0.297	0.956	0.947	1.000
Model aakr									
Training range	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000	30,000
Test range	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500	22,500
Validation range	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500	28,500
Data cleaning	no	no	no	no	no	no	no	no	no
Number of memory vectors	500	500	500	500	500	500	500	500	500
Optimal kernel width	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Vector selection method	x	x	x	x	x	x	x	x	x
Metrics									
Accuracy (% of span)	0.234	0.232	0.317	0.290	1.050	0.124	0.022	0.041	0.045
Auto–sensitivity	0.393	0.378	0.584	0.636	0.047	0.140	0.421	0.442	0.342
Cross–sensitivity	0.086	0.092	0.099	0.091	0.070	0.050	0.104	0.120	0.099
EULM detectability (% of span)	1.040	0.989	1.620	1.860	2.220	0.279	0.050	0.133	0.117
SPRT detectability (% of span)	0.00229	0.00231	0.00407	0.00405	0.00127	0.00139	0.000316	0.000312	0.000315
Uncertainty									
Analytic (% of span)	0.635	0.618	0.677	0.680	2.090	0.241	0.029	0.074	0.077
Coverage	1.000	1.000	0.999	1.000	1.000	0.983	0.892	0.998	0.973
Monte Carlo (% of span)	0.458	0.453	0.608	0.579	2.110	0.301	0.057	0.092	0.089
Coverage	0.999	0.999	0.998	1.000	1.000	0.996	0.997	0.999	0.995