NASA-CR-191650

Na... University
Raleigh, N.C. 27695-7921

p-97

TELEROBOTIC CONTROL OF A MOBILE COORDINATED ROBOTIC SERVER

NAG-1-1283-2

INTERIM TECHNICAL REPORT

Executive Summary

This interim report is comprised primarily of results from the Master's Degree Thesis of Mr. Robert Stanley, a graduate student supervised by the principal investigator on this project. The goal of this effort is to develop advanced control methods for flexible space manipulator systems. As such, a fuzzy logic controller has been developed in which model structure as well as parameter constraints are not required for compensation. A general rule base is formulated using quantized linguistic terms.; it is then augmented to a traditional integral control. The resulting hybrid fuzzy controller stabilizes the structure over a broad range of uncertainties, including unknown initial conditions. An off-line tuning approach using phase portraits gives further insight into the algorithm. The approach was applied to a three-degree-of-freedom manipulator system - the prototype of the coordinated flexible manipulator system currently being designed and built at North Carolina State University.

(NASA-CR-191650) TELEROBOTIC CONTROL OF A MOBILE COORDINATED ROBOTIC SERVER Interim Technical Report (North Carolina State Univ.) 97 p

N93-16387

Unclas

G3/63 0137563

North Carolina State University
MARS MISSION RESEARCH CENTER

Box 7921
Raleigh, NC 27695-7921
(919) 515-5931

FAX 919-515-5934
ETATE@JUPITER1.MMRC.NCSU.EDU

January 7, 1993

NASA Center for Aerospace Information
P. O. Box 8757
Baltimore/Washington International Airport
Maryland 21240

Gentlemen:

Enclosed are two copies of the Interim Technical Report for research performed under NASA Cooperative Research Agreement NAG-1-1283 entitled "Telerobotic Control of a Mobile Coordinated Robotic Servicer."

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$\underline{(\ )}$	vector
$\theta(t)$	angular position
$\theta(k)$	angular position at time sample k
C_i	$\cos(\theta_i)$
C_{ij}	$\cos(\theta_i + \theta_j)$
$\theta_d(t)$	desired angular position
$\theta_d(k)$	desired angular position at time sample k
$\dot{(\ )}$	first time derivative
$\ddot{(\ )}$	second time derivative
$\cap$	intersection
$\underline{R}$	manipulator vector with nonlinear components
Δt	sample period
S_i	$\sin(\theta_i)$
S_{ij}	$\sin(\theta_i + \theta_j)$
$(\ )_{ij}$	the i,jth entry of a matrix
τ	torque
T_{max}	torque maximum
τ_{max}	torque maximum
τ_{FPD}	torque supplied by FPD controller
$\underline{T}$	torque vector
$\cup$	union
A	event

A/D	analog to digital
AND	intersection
B	universe of event
CE	change in error
CE(k)	change in error at k
CEA	change in error angle
DOF	degree of freedom
DR-106	robot under construction at MMRC/NCSU
E	error
e(k)	error at time sample k
e(k-1)	error at previous time sample
FLC	fuzzy logic control
FPD	fuzzy proportional derivative controller
FPID	fuzzy proportional integral derivative controller
G	proportional gain matrix
g	proportional gain
H	derivative gain matrix
h	derivative gain
I	control input
I	integral gain matrix
i	integral gain
k	discrete time sample
L	length
LN	large negative
LP	large positive
M	mass matrix

m .... mass

max .... maximum

min .... minimum

MN .... medium negative

MP .... medium positive

OR .... union

PD .... proportional derivative control

QCE .... quantized change in error

QCEA .... quantized change in error angle

QE .... quantized error

QI .... quantized input

rad .... radians

sec .... seconds

SN .... small negative

SP .... small positive

U .... universe of discourse

u .... membership value

X(n) .... n^{th} state

ZE .... zero

1. INTRODUCTION

Many future NASA missions require robotics to assist in the assembly, maintenance and servicing of spacecraft. Such scenarios may include one or more multi-linked manipulator arms which, because of their lightly damped characteristics, require vibration suppression as well as end point tracking in a somewhat uncertain environment. Due to the flexibility in the joints/links and the inherent vibration due to the mobility of the robotics system, adaptability to the environment and varying inertia is a requirement.

Several methodologies have been suggested for robot control based upon known tasks and environments. Classical proportional-integral-derivative (PID) control has been employed in industry for many years. The approach assumes complete knowledge of all pertinent system and environmental characteristics. It also requires tuning the PID gains to meet some performance specifications. When the system or environmental parameters change, the gains must be re-tuned accordingly. Thus, unknown disturbances or changing environmental conditions may result in performance degradation.

To address the issue of uncertainty or time-varying conditions, several adaptive control algorithms have been suggested. These include joint-space control [18] and global linearization [4] methods in which some nonlinear or discrete matrix polynomial equation set must be solved in order to construct the controller. While these methods may guarantee stability under certain restrictions, the computation time may limit their implementation for multi-linked robotics systems.

Fuzzy logic control offers an alternative approach in which the structure of the system model is not required for control design [25,2]. Fuzzy control algorithms have been applied to several process control and automotive systems [8,14,13] in which the time constants were somewhat large. The use of fuzzy logic for robotics systems has yielded some success [16,5] although issues such as time delays and initial conditions sometimes limit the applicability of these algorithms.

This thesis develops a fuzzy logic control algorithm which can be applied to systems with uncertainties. These uncertainties may include unknown initial conditions, and undetermined system dynamics. Unknown initial conditions may exist in space manipulator systems due to sensor inaccuracies.

The concepts of fuzzy logic control are presented in a progressive manor. First, an extensive development concentrating on the theories of fuzzy logic control is presented. Secondly, fuzzy logic control is applied to two simple systems; the first being a horizontal pendulum while the second example is a vertical pendulum. Then the algorithm is applied to a three degree-of-freedom robotic manipulator.

The horizontal pendulum provides an environment with which to develop a FLC that produces performance characteristics similar to traditional proportional derivative control. In the course of this development seven linguistic terms are presented and defined over a quantized Universe of Discourse. The membership function used to define the linguistic terms differs from classical methods. A novel means to quantize the change in error is presented. Utilization of the fully populated rule base results in a general fuzzy logic controller that accomplishes vibration suppression (Fuzzy-PD). By examination of the phase portraits an off-line tuning approach is considered

The application of Fuzzy-PD to the vertical pendulum provides the ground-work for the development of a fuzzy logic controller that not only accomplishes vibration suppression but also compensates for steady-state errors. In the course of this development, a capture method and an alternative rule base are provided to compensate for system biases. The ultimate result is the hybridization of the Fuzzy-PD controller with traditional integral control resulting in what is referred to as Fuzzy-PID.

At this point the unrestricted fuzzy logic controller (Fuzzy-PID) is applied to the highly coupled, second order, nonlinear dynamics associated with a three degree-of-freedom robotics manipulator. The results of such an application as compared to traditional PID control illustrate the performance of Fuzzy-PID.

Concurrent to this investigation, a coordinated teleoperated mobile manipulator system is being designed and fabricated. The system will contain two DR-106 six-degree of freedom manipulators with flexible joints/links supported by a mobile platform. Several algorithms such as the fuzzy logic controller developed in this thesis are being considered whereby the human operator inputs the desired trajectory and the controller tracks the desired trajectory while suppressing vibration and compensating for platform motion. Such performance measures are typical for in-space robotics operations.

2. BRIEF OVERVIEW OF FUZZY LOGIC CONTROL

Fuzzy logic control, is quite confusing when initially introduced. However, like many concepts in life, once a global understanding is obtained the confusion associated with the specifics diminishes. Therefore, before more complicated deliberation on the uses of FLC may be developed, it may be in the readers best interest to consider a brief overview of the fuzzy logic control algorithm (FLC). The block diagram of the fuzzy logic control algorithm is illustrated in Figure 2.1

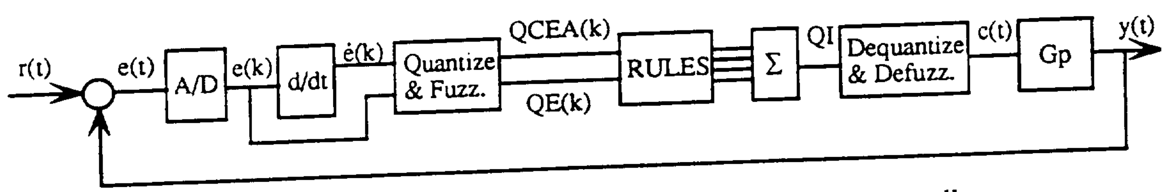


Figure 2.1: Simplified block diagram of fuzzy logic controller

Fuzzy logic control is a rule based controller. As the term "rule base" indicates, the fuzzy logic control algorithm is based on a number of rules which are accessed and processed in a specific fashion so as to provide the desired control input to a system. In order to construct such a controller, consideration must be given to the development of the rules and how they interact to form the control input.

Individual rules are constructed using qualitative terms in conjunction with IF . . . THEN statements. Some examples of common qualitative terms are big, small, large, hot, normal, fast, slow, etc. . .

A linguistic rule used in the process of balancing a stick may read: IF the stick is inclined moderately to the left AND is almost still THEN move the hand to the left quickly. In order for this rule to be useful a process must be implemented by which the linguistic terms "moderately", "almost still", and "quickly" are converted into some numeric value. Fuzzy set theory does just that.

Notice how the qualitative linguistic terms are vague in their meaning. This is a desired result because it closely resembles how humans think. The process provided by fuzzy set theory which enables a linguistic term to take on range of values is called the membership function. The membership function is defined over a domain referred to as the Universe of Discourse and assumes a value which ranges from 0 to 1. This value is referred to as the membership value. In general the membership value is a way to weigh how much of a particular linguistic term is present.

How does one determine what is "quick"? For example, if the term "quick" is defined to be 100 mi/hr. its associated membership value would be 1. Any variations, either positive or negative from the speed 100 mi/hr. would result in a membership value of less than 1. For example, 90 m/hr. may correspond to a membership value of .9. Therefore one may conclude that only 90% of the linguistic term "quick" is present.

At this point various rules using linguistic terms in conjunction with IF. . .THEN statements may be developed. However, because a large number of qualitative linguistic terms exist in the human language it is desirable to choose an appropriate number of linguistic terms and to define what region they are valid.

For control applications the linguistic terms tend to read as Large, Medium, Small, and Zero. By considering both positive and negative values of the linguistic terms listed one has seven distinct qualitative linguistic terms with which to construct rules. (Larger Positive, Medium Positive, Small Positive, Zero, Small Negative, Medium Negative, Large Negative). These linguistic terms must be defined over a region called the Universe of Discourse. It is common practice to define the qualitative terms over a quantized Universe of Discourse. By doing so the qualitative linguistic terms may be used to describe more than one state in a system. Take for example the rule: IF error is

Large Positive AND the rate of error change is Large Positive THEN the control input should be Large Negative. Even though the qualitative linguistic term defining error and error change is the same (Large Positive) the membership value associated with both of the states may differ.

With the concepts used to develop a rule presented and the foregone conclusion that a fuzzy logic controller requires more than one rule to accomplish any reasonable task, a brief discussion of the techniques used to process the rule base follows. The rule base is simply all of the rules created to perform a particular task. Given the states of one such particular task, it is likely that some of the rules will be appropriate to the conditions presented and others will not. The rules that are inappropriate are discarded by the use of the Logic Product and the others are combined using the Logic Sum. The result of this combination of rules is a weighted area. By finding the Center of Gravity of this area a single numeric value results. This value is the final fuzzy inference and is a quantized value. Dequantizing this fuzzy inference results in the final control input.

With this brief overview complete, the reader may proceed further where a more detailed description of fuzzy logic control and its applications to various dynamic systems is presented.

3. DEVELOPMENT OF THE FUZZY LOGIC CONTROL ALGORITHM

3.1 MEMBERSHIP FUNCTION

The basis for Fuzzy Logic Control (FLC) is the membership function, commonly referred to as the membership shape. A membership function is defined over a domain called the Universe of Discourse (U) and assumes a range from zero to one, referred to as the membership value (u). The universe includes all events that can take place in the context of a particular situation. Restated, the universe exists over the boundaries of a given situation.

The *membership value* (u) describes the probability of an event occurring, given a particular universe[15]. Probability is defined as the ratio of two numbers. The numerator represents the events in the universe on which interest is focused, and the denominator represents the universe of all possible events. Therefore, the numerator is a subset of the denominator. With this in mind, the probability of an event will range from zero to one indicating from non-membership to total-membership in the universe. Symbolically $u\langle A|B\rangle$, reads as follows: "the membership value (u) is the probability of event A given the universe of events B," where

$$u\langle A|B\rangle = \frac{A}{B} \qquad \text{Eq.(3.1)}$$

and

$$0 \le u\langle A|B\rangle \le 1 \qquad \text{Eq.(3.2)}$$

FLC is based on a set of heuristic rules. These rules use qualitative words which are defined mathematically in the form of membership functions. To illustrate this relationship between qualitative words and membership functions consider the following example. Suppose one was to look at the normal height of males. Say for instance that a reasonable normal height is 5 ft. 9 ins. This is not to say that people who are of height of 5 ft. 4 ins. or 6 ft. 2 ins. are not of normal height. Those are normal heights, but somehow the feeling of "normal height" is not as strong. However, at heights of less than 5 ft. 0 ins. or more than 6 ft. 6 ins. one could categorically say this is "short" or this is "tall". Therefore, as one moves from short to tall, the feeling of normal gradually rises and than gradually falls. If values from zero to one are assigned to this feeling of "normal height", the result could be the bell shaped curve shown in Figure 3.1.

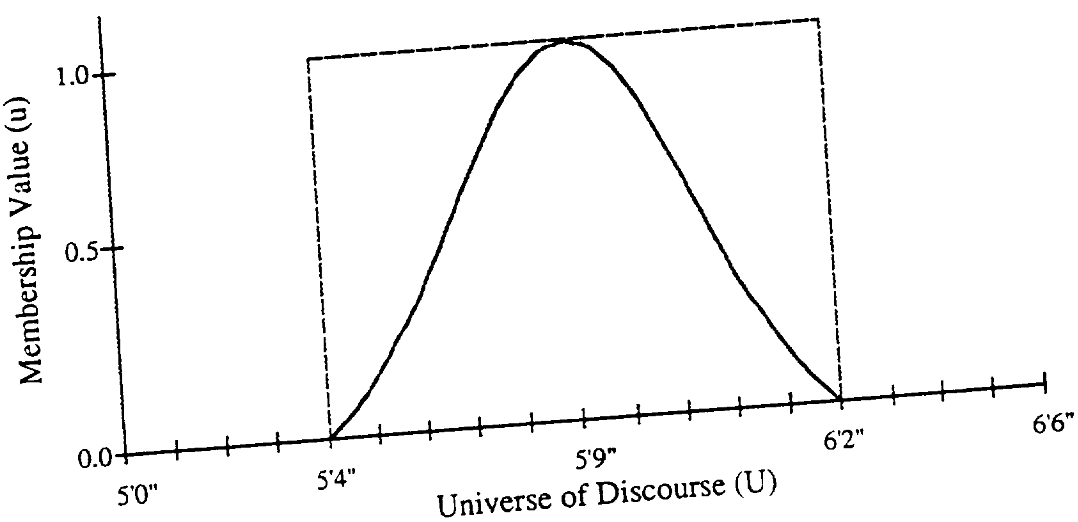


Figure 3.1: Membership function for people of normal height

This bell shaped curve is called the *membership function* and is defined over the Universe of Discourse (heights ranging between 5 ft. 4 ins. and 6 ft. 2 ins.) to take on values from zero to one, where one is the strongest feeling of "normal". With this concept of membership function defined, one now has the ability to quantitatively deal with inexact or ambiguous issues such as the qualitative words short, normal, tall.

The idea of a membership function differs from a binary approach. The dotted box in Figure 3.1 illustrates binary theory where there are only two membership values: zero and one. According to this, any person in the range of 5 ft. 4 in. to 6 ft. 2 in. takes on the membership value one and is of perfect normal height. Heights less than 5 ft. 4 ins. and greater than 6 ft. 2 ins. correspond to a zero membership value and are thus completely non-normal; clearly this is an unnatural situation.

It has been shown that the exact shape of the membership function is relatively arbitrary and may be chosen based on user preference[9]. Figure 3.2 illustrates some commonly chosen membership functions. Figure 3.3 illustrates how similar they are by superimposing them.

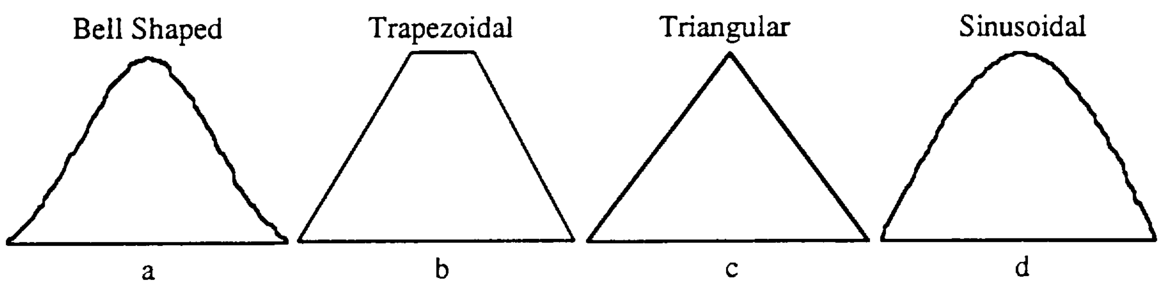


Figure 3.2: Commonly chosen membership functions

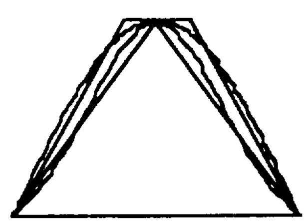


Figure 3.3: Similarities between some different membership functions

Due to the complexity of the bell shape (Figure 3.2a) and the piece-wise continuous behavior of the trapezoidal shape and triangular shape (Figure 3.2b& 3.2c), the smooth continuous and easily calculated sinusoidal shape has been selected as the membership function for this study (Figure 3.2d).

3.2 LINGUISTIC RULES

To develop a fuzzy controller, it is necessary to interpret linguistic rules that are based on experience so as to form a control surface that provides output values of the controller, corresponding to situations of interest[11]. The basis for the linguistic rule is the "IF. . .THEN" statement. One linguistic rule or "production rule" describes a portion of a particular problem or task in words. The antecedent blocks ("if" phrases) describe the states, and the consequent block ("then" phrase) describes how the controller should respond to the states.

For example, asking a first shift operator on an assembly line to describe a single portion of his/her task, a typical response may be:

IF the parts are running "far behind" and they "have been" for a period of time THEN I increase the line speed "alot".

This particular response is based on the operator's experience and is to be interpreted to produce a production rule. However, a complication arises when the same question is asked of the second shift line operator. The response may differ in the actual vocabulary used, but the premise would remain the same. Therefore, it is necessary to define a common or universal set of linguistic terms (common vocabulary) which may be used to specifically define the production rule. Refining the operator's response using specific linguistic terms results in a typical fuzzy linguistic rule:

*IF (the **error** is "large negative") AND (the **change in error** is "zero") THEN (the control **input** should be "large positive").*

Notice the change in the antecedents blocks and the consequence block. The term "parts . . . far behind" corresponds to error being *large negative*, while the term "have been for a period of time" corresponds to change in error being *zero*. Further, the term "increase . . . alot" corresponds to a control input *large positive*.

In general, in order to develop and interpret this rule, or any other fuzzy linguistic rule, the following concepts need to be addressed.

1. How are ***error*, *change in error*** and control ***input*** defined?
2. How are the qualitative linguistic terms "large positive", "zero", and "large negative" defined?

3.2.1 ERROR, CHANGE IN ERROR, AND CONTROL INPUT

Error is defined as the difference between the process output and the desired output:

$$e(k) = \theta(k) - \theta_d(k) \qquad \text{Eq.(3.3)}$$

where

$e(k) =$ error at time sample k
$\theta(k) =$ position at time sample k
$\theta_d(k) =$ desired position at time sample k

The *change in error* is the difference between the error from the current process output and the error from the last process output.

$$ce(k) = e(k) - e(k-1) \qquad \text{Eq.(3.4)}$$

where

$ce(k) =$ change in error at current sample
$e(k) =$ error at current sample
$e(k-1) =$ error at previous sample

All of the examples to follow are maneuvers of mechanical dynamic systems whose dependent variable is an angle $\theta(t)$ given in either radians or degrees. The *control input* is the input torque applied to the process.

3.2.2 QUALITATIVE LINGUISTIC TERMS

As illustrated in the line operator example, it is important to develop a set of qualitative linguistic terms to be used in the controller. In the same way the qualitative linguistic term "normal" was defined over the universe of heights ranging from 5 ft. 4 ins. to 6 ft. 2 ins., the linguistic terms for the FLC will span a quantized universe or domain defined from -6 to +6. These limits from -6 to +6 are not hardfast, rather they are chosen such that the individual membership functions begin and end on a whole number. As previously mentioned sinusoidal membership functions will be used to define the linguistic terms. Figure 3.4 illustrates seven such qualitative linguistic terms defined over a quantized Universe of Discourse ranging from -6 to +6 and their respective defining functions: large positive(LP), medium positive(MP), small positive(SP), zero(ZE), small negative(SN), medium negative(MN), and large negative(LN).

The purpose for defining the qualitative terms large-positive through large-negative on a quantized Universe of Discourse is to allow their universal use in defining error, change in error and the control input to the system. This may be accomplished by simply quantizing the values of error, change in error and the control input to the system to the values –6 to +6 on the Universe of Discourse (see Figure 3.5).

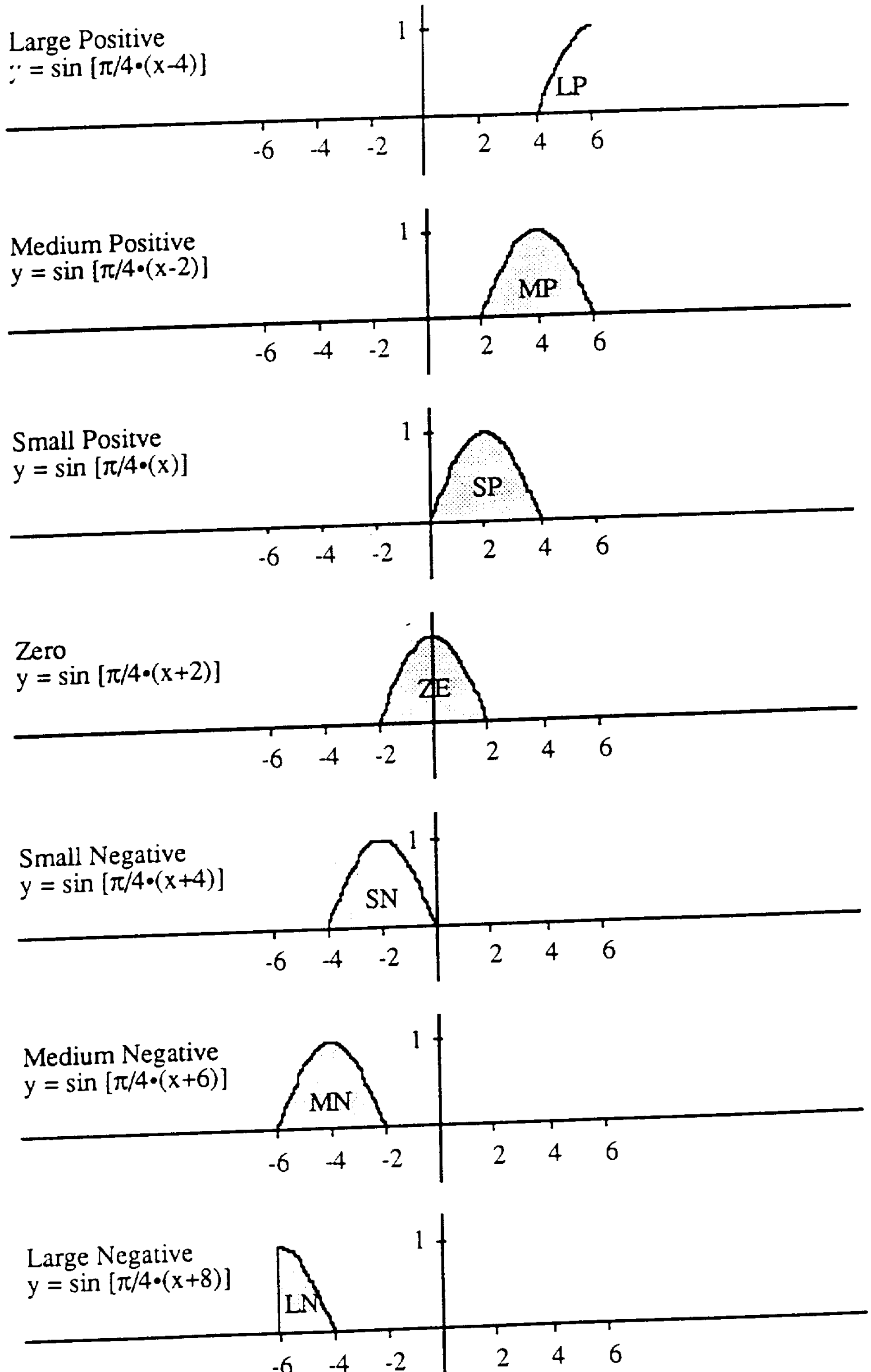


Figure 3.4: Linguistic quantized qualitative terms and their respective functions

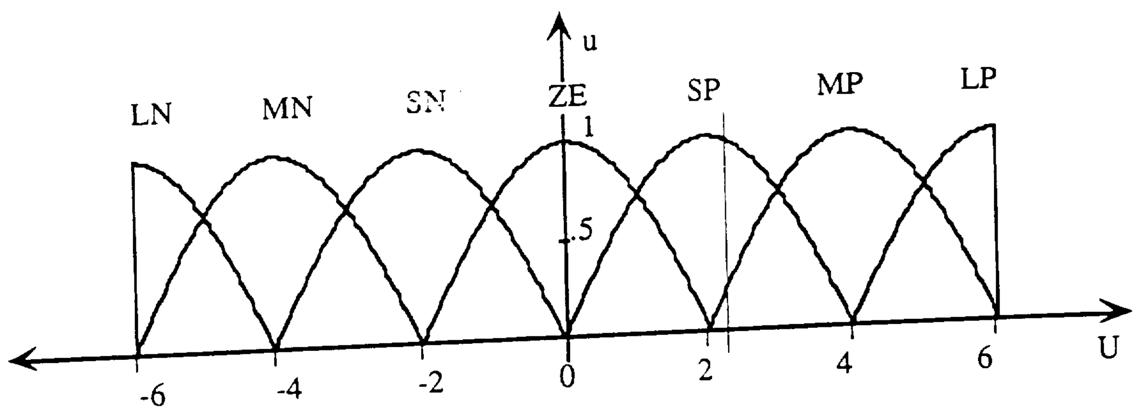


Figure 3.5: Qualitative linguistic terms defined on a quantized universe from -6 to +6

For example, suppose the measured error and calculated change in error in a particular system after A/D is 22 degrees and 33 degrees/sec, respectively. If the quantization function for error, equals one tenth of the measured erro., then the quantized error is 2.2. That is,

$$\text{quantized error} = \left(\frac{6}{60}\right) \times (\text{error}) \qquad \text{Eq.(3.5)}$$

Notice in Figure 3.5 that if a vertical line is drawn through the point 2.2, it intersects the membership functions SP and MP. Therefore, the error is a combination of a weight of the membership function Small Positive and a weight of the membership function Medium Positive.

If the quantization function for change in error is defined as

$$\text{quantized change in error} = \left(\frac{6}{90}\right) \times (\text{change in error}) \qquad \text{Eq.(3.6)}$$

then the quantized change in error is 2.2. Notice again in Figure 3.5 that this corresponds to some membership value of the linguistic term Small Positive and some membership value of Medium Positive. Collectively, error and change in error have been shown to posses the same quantized value and linguistic values SP and MP while retaining different actual values, thereby demonstrating the transcendental usefulness of the quantized Universe of Discourse.

It has been shown that the number of linguistic terms is arbitrary. As the number of linguistic terms increases, the resolution of the controller increases as a direct result of the induced ability to define each linguistic rule with more accuracy. In most FLC it is common practice to use only three to five linguistic terms. However, due to the complexity associated with the controller for robotics systems, seven linguistic terms are employed.

3.3 RULE BASE

In order to develop a fuzzy logic controller, a series of rules must be assembled. It is the assembly of production rules in which a repertoire of learned problem-solving actions (consequences) is associate with conditions (antecedents), to form condition-action pairs. Once a situation is recognized, the conditions constitute cues or indices for corresponding actions. This is how FLC attempts to model the heuristic problem solving approach of humans[15]. In the assembly line operator example, one production rule governing the case when the error is large negative and change in error is zero was developed. However, in order to handle other cases such as the error being medium positive and the change in error being large positive, one must develop other rules. Therefore, for each particular situation of interest, there exists a corresponding production rule. Combination of the production rules results in what is referred to as a "rule base". In order to assimilate the rule base, the concepts of *logic product*, *logic sum*, *center of gravity*, and the *quantization functions* must be developed. An illustrative example of an inverted pendulum is presented in order to mature these concepts.

3.3.1 LOGIC PRODUCT

Figure 3.6 illustrates seven rules which are commonly used for vibration suppression of an inverted pendulum.

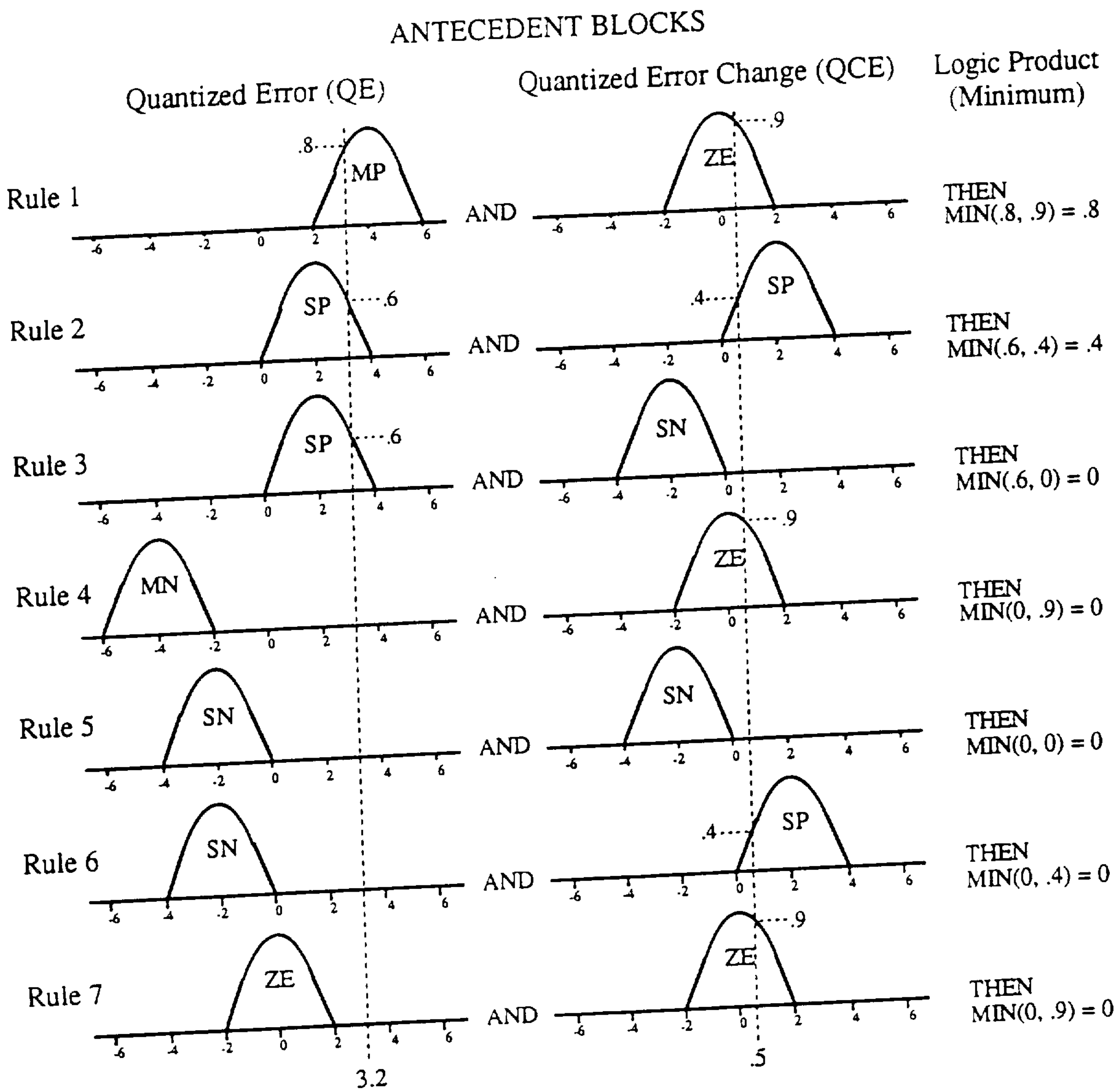


Figure 3.6: Seven rules used for an inverted pendulum

Logic product is the first of the concepts to be developed. The physical significance of taking the logic product is to discard any and all rules that are not relevant to a given pair of error and change in error values. In set theory, the logic product is the *AND* function. It can simply be defined as taking the minimum of the corresponding numeric entries of two sets. Mathematically, "the intersection of two sets, $A \cap B$, corresponds to the *AND* function and is define by

$$u(\text{A AND B}) = \min(u_a(x), u_b(x)) \qquad \text{Eq.(3.7)}$$

Figure 3.6, demonstrates how the antecedent block or the quantized error and quantized change in error are joined by the *AND* function, resulting in a logic product. The dashed vertical lines in the quantized error (QE) and quantized change in error (QCE) columns represent a given QE of 3.2 and QCE of .5. Examining the first row of Figure 3.6, which corresponds to Rule 1, one notices the quantized error membership value equals 0.8 (u=0.8) and the change in error membership value equals 0.9 (u=0.9). The minimum of these two values is 0.8, therefore u(.8 AND .9)=.8. The same procedure is performed for all of the rules as illustrated in Figure 3.6.

$$\{\min(.8,.9), \min(.6,.4), \min(.6,0), \min(0,.9), \min(0,0), \min(0,.4), \min(0,.9)\}$$

results in the logic product (a set).

$$\{.8,.4,0,0,0,0,0\}$$

The logic product is used because it provides the condition in which both error and change in error are satisfied. As shown in rule one (Fig.3.6) the membership value of u=0.8 satisfies both conditions; therefore it is transferred to the consequence block.

3.3.2 LOGIC SUM

The logic sum is an operator such that the contribution of each individual rule is combined to produce the final fuzzy inference. Mathematically stated, it is the OR function. The union of two sets, $A \cup B$, corresponds to the OR function and is defined by

$$u(\text{A OR B}) = \max(u_a(x), u_b(x)) \qquad \text{Eq.(3.8)}$$

The OR function is applied to the consequence block because even if an individual rule's influence or contribution is small, it should still be reflected in the resulting control input.

Using the previous values of QE and QCE one can view how the contributions of each rule are transferred to the consequence block.

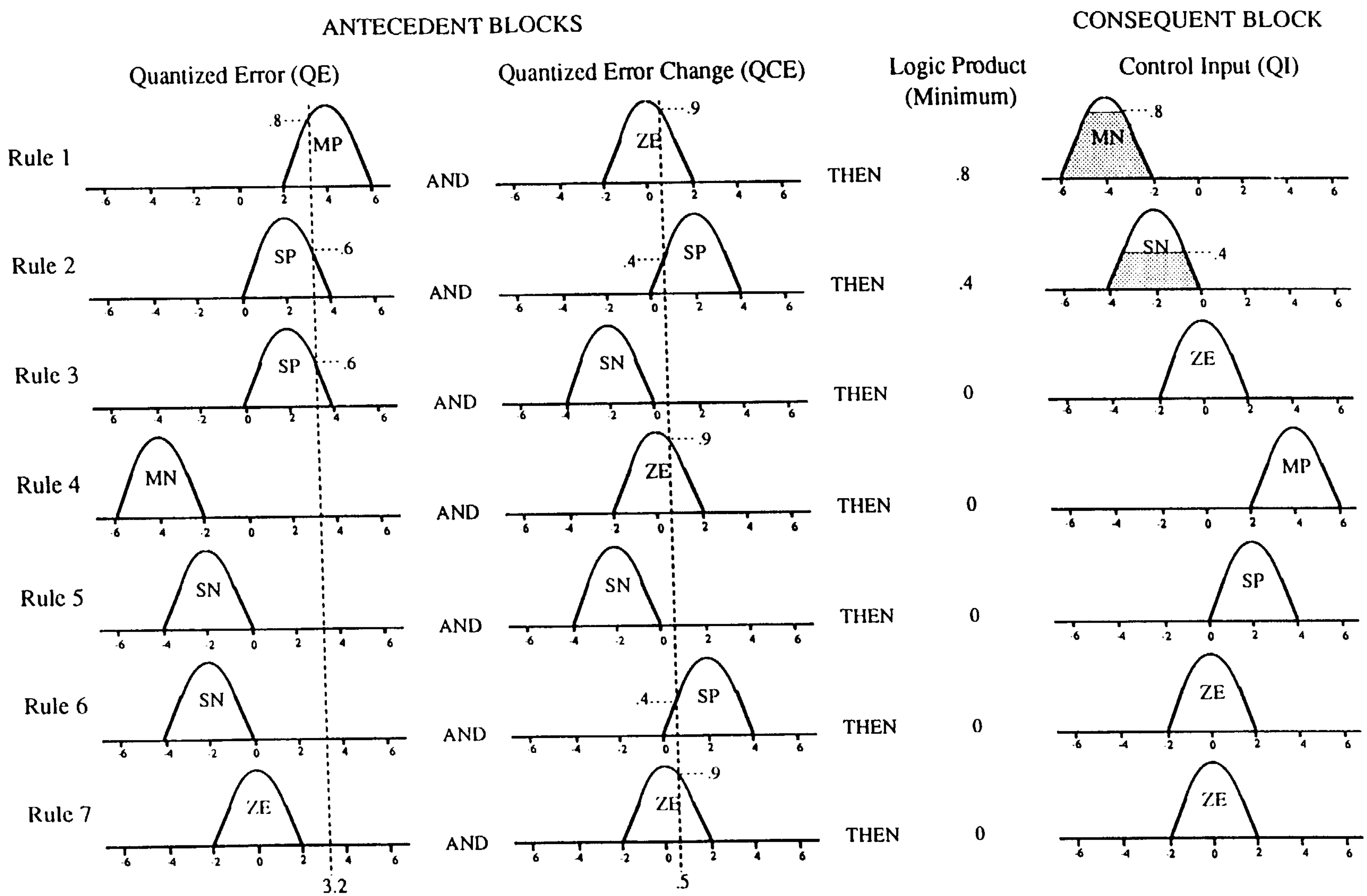


Figure 3.7: Logic sum

The logic sum is the OR function applied to the shaded area in the "Control Input" column shown in Figure 3.7. This shaded area is referred to as the conclusion of the fuzzy inference. Figure 3.8 illustrates the result of imposing all of the contributions on the Universe of Discourse.

The conclusion of the fuzzy inference is an area and cannot be used directly to produce a control command. Therefore, a conversion technique is needed to convert the fuzzy inference into a control quantity. The most common means by which to accomplish this goal is the use of the *center of gravity* method.

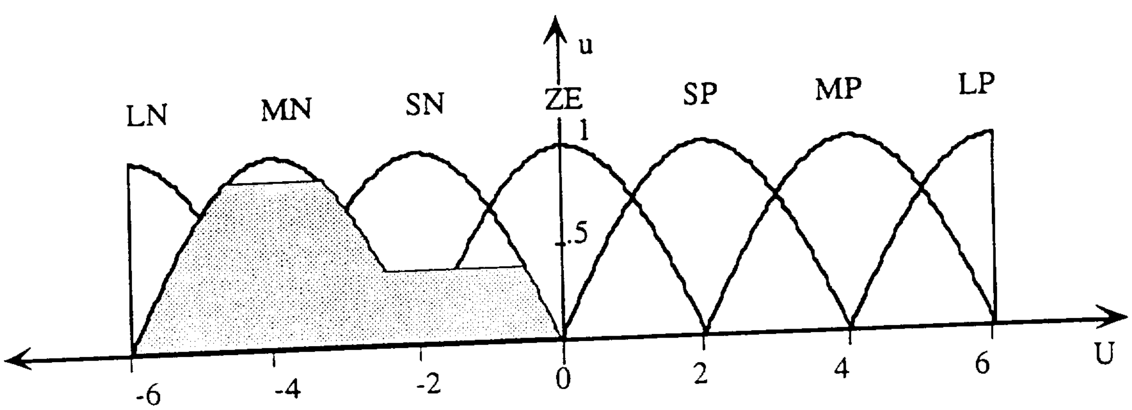


Figure 3.8: Final inference produced by the FLC

3.3.3 CENTER OF GRAVITY

The *center of gravity* method is the most commonly applied way of combining the individual consequences of each rule to get a specific control quantity that may be sent to the process under control. The shaded area in Figure 3.8 is the final inference produced by fuzzy controller. This area, however, cannot be used directly to control the output of the system. Therefore, the center of gravity of this area is taken. In general the equation for the center of gravity is

$$QI = \sum_{1}^{n}(u_n \times U_n) \Big/ \sum_{1}^{n} u_n \qquad \text{Eq.(3.9)}$$

where

$QI =$ Quantized Input

$u_n =$ membership value

$U_n =$ Universe of discourse

Referring to the final inference illustrated in Figure 3.8, and using Eq.(3.9)

$$QI = (.8 \times -4) + (.4 \times -2)/(.8 + .4) \qquad \text{Eq.(3.10)}$$

or

$$QI = -3.3 \qquad \text{Eq.(3.11)}$$

Figure 3.9 illustrates the position of the center of gravity.

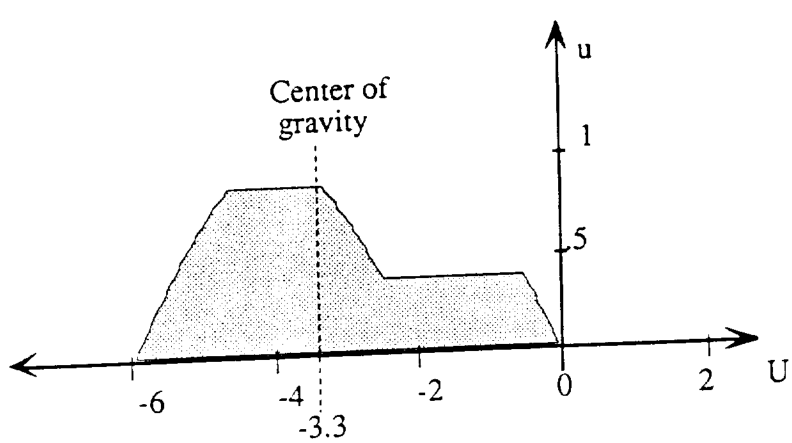


Figure 3.9: Center of gravity method

The quantized value of the input QI=-3.3 can now be directly related to a control input applied to the system by simply dequantizing it into an applied torque. For example, if the dequantizing function is

$$\text{Input} = \left(\frac{150}{6}\right) \times (\text{quantized input}) \qquad \text{Eq.(3.12)}$$

then the torque applied to the system

$$I = \left(\frac{150}{6}\right) \times (-3.3) \qquad \text{Eq.(3.13)}$$

is I=83.33 in-ounces.

This is the defuzzification operation. The method of defuzzification that employs center of gravity is known as the Mamdani method.

3.4 THE FUZZY CONTROL ALGORITHM

The overall general fuzzy control algorithm may now be summarized as follows(Fig. 2.1). First a pair of error and change in error values are measured and calculated respectively(Eqs. 3.3 & 3.4). These two states are then converted into quantized error and quantized change in error(Eqs. 3.5 & 3.6). These quantized values correspond to particular qualitative linguistic terms(Figs. 3.4 & 3.5). The linguistic terms are then applied to the antecedent block of the control rules. If both conditions in the antecedent block are met then a resulting consequence is registered(Eq. 3.7 & Fig. 3.6). All of the individual consequences are then combined by the use of the logic sum(Eq. 3.8 & Fig. 3.7). This results in a final fuzzy inference(Fig 3.8). This final inference may then be converted into a quantized input by application of the center of gravity method(Fig. 3.9 &

Eq. 3.9). The final step is to dequantize the quantized input to the control input to be applied to the system(Eq. 3.12).

The algorithm for FLC just presented is not specific, and it may be applied to a multitude of problems. The intent here, however, is to develop a FLC for robotics systems. Towards this end, two particular examples, a horizontal pendulum and a vertical pendulum, are supplied. These are given in the Chapters 4.

4. ILLUSTRATIVE EXAMPLES

Before applying the fuzzy logic control algorithm developed in Chapter 3 to a robotics example (the coupled nonlinear dynamics of a revolute three degree-of-freedom robot in this study), some physical insight into the behavior of the fuzzy logic controller is desirable. This may be obtained by first applying the fuzzy control algorithm to a simple second-order, linear system and comparing the response to a step input to that of a traditional PD controller. Once this is accomplished, the development of a FLC to handle the slightly more complicated dynamics of a tradition vertical pendulum will be considered.

4.1 THE HORIZONTAL PENDULUM

Consider a massless rod of length (L) in the horizontal plane with a concentrated mass (m) at the endpoint, and an input torque (τ) supplied by a motor. This system is referred to as a *horizontal pendulum*. The equation of motion for the system is.

$$\ddot{\theta} - \frac{\tau}{mL^2} = 0 \qquad \text{Eq.(4.1)}$$

where $\theta(t)$ is the dependent variable defining the angular position of the pendulum. When looking at the dynamics of this simple system notice that a traditional proportional derivative (PD) controller is adequate for vibration suppression. This is accomplished by defining the torque as:

$$\tau = -g(\theta - \theta_d) - h(\dot{\theta} - \dot{\theta}_d) \qquad \text{Eq.(4.2)}$$

where g and h are gains chosen to meet desired performance specifications. For this study the selection of the gains is based on the Theory of *Natural Control*[17]. Substituting Eq.(4.2) into Eq.(4.1) gives the overall closed loop system dynamics.

$$\ddot{\theta} + h(\dot{\theta} - \dot{\theta}_d) + g(\theta - \theta_d) = 0 \qquad \text{Eq.(4.3)}$$

Consider a specific fuzzy logic controller that provides the same performance characteristics as the traditional PD controller when applied to the horizontal pendulum. For clarification purposes, this fuzzy logic controller is referred to as "Fuzzy-PD" or "FPD". It is noticed in Chapter 3 that in order to develop a FLC, the following need to be defined:

(1) The number of qualitative linguistic terms used.

(2) The Universe of Discourse.

(3) Sign convention on error and change in error.

(4) The quantization functions.

(5) The number of rules in the rule base.

4.1.1 NUMBER OF LINGUISTIC TERMS, UNIVERSE OF DISCOURSE, AND SIGN CONVENTION

Equations 3.3-3.6, and 3.12 were presented in such a way that they are consistent with the Fuzzy-PD controller now under consideration. Therefore, the seven qualitative terms, and the quantized Universe of Discourse used in the previous Chapter will now be applied(Fig.3.4 & Fig. 3.5). It is interesting to note that the error, change in error and input to the system do not have to be quantized to the same seven qualitative linguistic values. For example, it may be decided that only three linguistic terms are needed for an accurate description of error (ex: Positive, Zero, Negative) but five linguistic terms may be needed for change in error (MN,SN,ZE,SP,MP). This is acceptable when developing a FLC for a particular application. For simplicity, this controller does quantize error, change in error and input to the same seven linguistic values ranging from large positive to large negative.

The sign convention for the error and the change in error are defined for this physical system is shown in Figure 4.1.

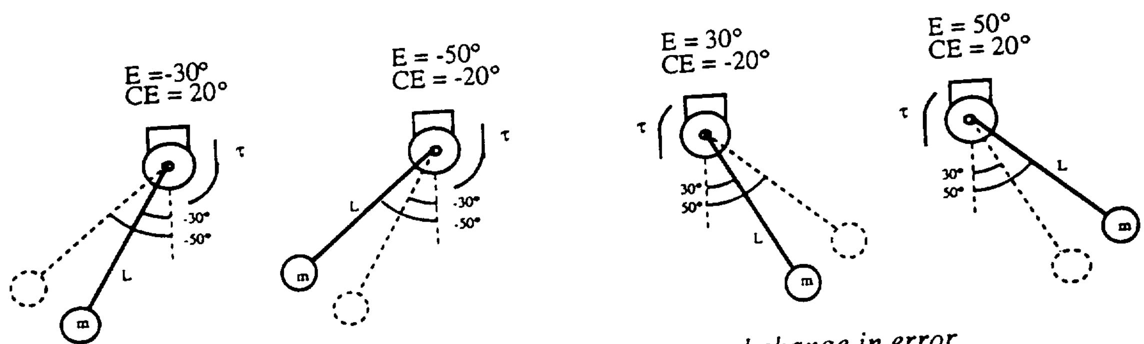


Figure 4.1: Sign convention for error and change in error

4.1.2 QUANTIZATION FUNCTIONS

In Chapter 3 the idea of developing rules based on *error*, *change in error* and *control input* was presented. This approach is now applied to the FPD controller. It was noted in Chapter 3 that quantization functions were required for all three parameters (E,CE,I). (see Eqs. 3.5, 3.6, & 3.12). Therefore, before the Fuzzy-PD rule base can be developed it is important to specifically define how the antecedent and consequence blocks are quantized.

4.1.2.1 QUANTIZED ERROR

In general it is desired to quantize the error and change in error such that the maximum and minimum quantized values correspond to the maximum and minimum actual values. For example, consider the quantization function for the *error* antecedent.

$$QE = \left(\frac{6}{\text{maximum expected error}} \right) \times E \qquad \text{Eq.(4.4)}$$

Keeping in mind that the ultimate intent for this thesis study is to develop a FLC to control a 3-DOF revolute manipulator, one notices that the DR-106(Fig.5.1) possesses physical limitations pertaining to the working space. The maximum working space for any of the three links is restricted to plus or minus 60 degrees. Therefore, the maximum expected error is set to 60 degrees.

$$QE = \left(\frac{6}{60} \right) \times E \qquad \text{Eq.(4.5)}$$

Figure 4.2 shows the error quantization function in Eq.(4.5). Notice that quantized error is clipped to either +6 or -6 if the error exceeds the expected limits.

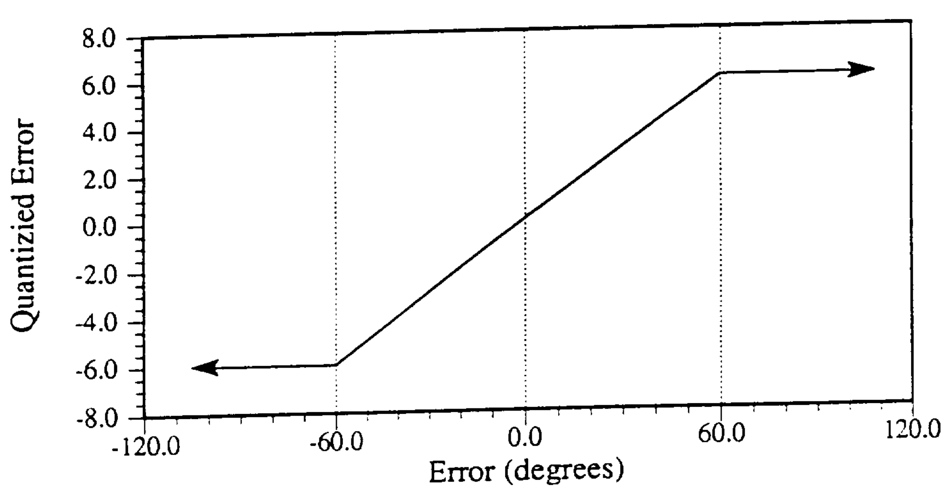


Figure 4.2: Quantized error as a function of error in degrees

4.1.2.2 QUANTIZED CHANGE IN ERROR

The quantization function for change in error is slightly more involved. The change in error is the first derivative of error or the slope of the error curve at a particular time. If the error changes rapidly with respect to time, the CE(slope) approaches infinity. As previously discussed, in order to use CE in the antecedent block it must be quantized. How does one quantize change in error values ranging from negative to positive infinity?

A novel solution to this problem is provided by taking the inverse tangent of the slope of the error with respect to the sampling period. This mapping operation provides a bounded domain for the change in error between negative 90 and positive 90 degrees. With this domain defined on a closed set, it is quite easy to parameterize the change in the error as a *change in error angle*(CEA).

$$\mathrm{CEA} = \tan^{-1}\left(\frac{e(k) - e(k-1)}{\Delta t}\right) \qquad \text{Eq.(4.6)}$$

where the $-90 \leq \mathrm{CEA} \leq +90$. Using the bounds of CEA and the same premise developed for quantizing error results in *quantized change in error angle*(QCEA) being defined as:

$$\mathrm{QCEA} = \left(\frac{6}{90}\right) \times \mathrm{CEA} \qquad \text{Eq.(4.7)}$$

The graphical representation of this quantization procedure is illustrated in Figure 4.3.

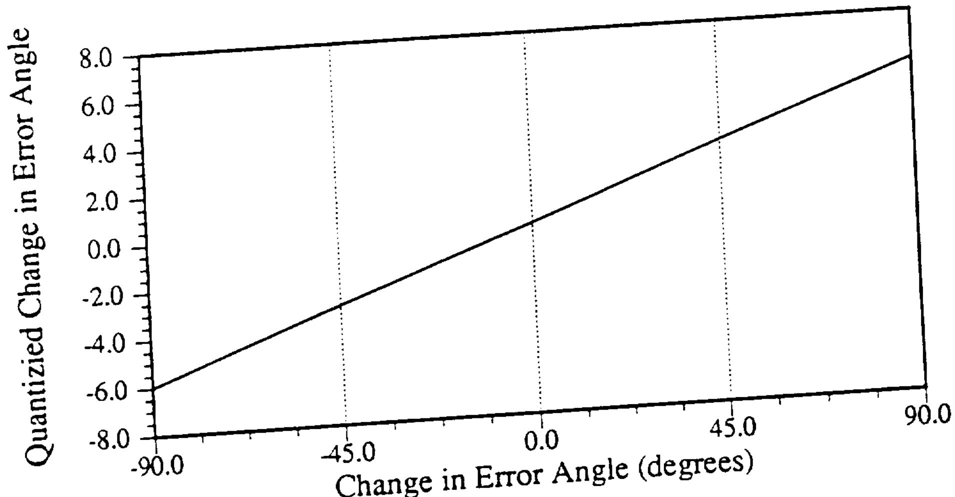


Figure 4.3: Quantized Change in Error Angle as a function of Change in Error Angle

4.1.2.3 DEQUANTIZED INPUT

With the antecedent quantization functions accounted for, the last quantization function to be selected is associated with the input to the system. This process is actually a dequantization process. Given the control input in quantized terms, it is dequantized to a torque value which is then applied to the system(Eq.4.8). For the example under consideration, the torque varies from +150 to −150 inch-ounces. This number was initially selected and adjusted according to the behavior of the step response. In general:

$$I = \left(\frac{\tau_{max}}{6}\right) \times QI \qquad \text{Eq.(4.8)}$$

and specifically:

$$I = \left(\frac{150}{6}\right) \times QI \qquad \text{Eq.(4.9)}$$

where

QI = quantized input

I= control input to the system (in-ounces)

This quantization procedure is illustrated in Figure 4.4.

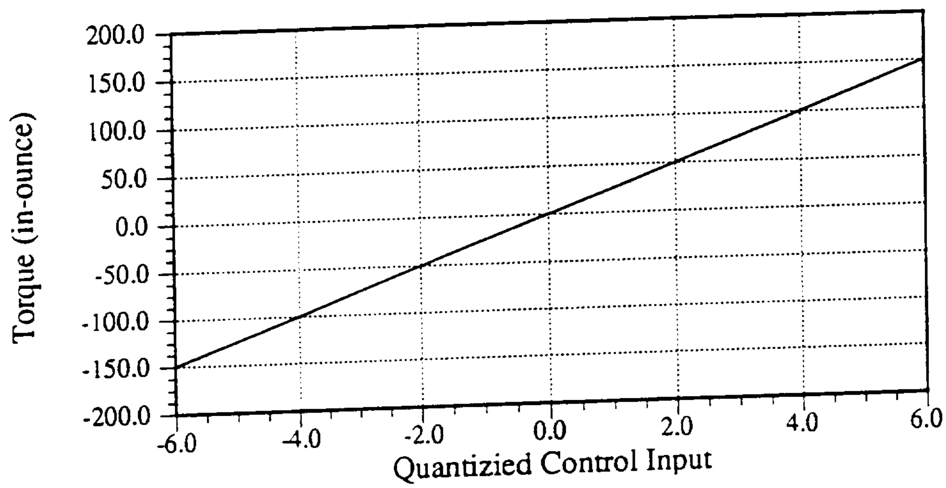


Figure 4.4: Torque as a function of quantized control input

4.1.3 NUMBER OF RULES

As a preliminary investigation leading up to the final Fuzzy-PD controller, seven rules were applied to the horizontal pendulum. This resulted in a poor performance associated with the controller's lack of ability to handle various wide ranges in initial conditions. To compensate for this poor performance, an increase in the rule base to 36 rules followed. This control surface also failed due to a lack of robustness.

Given the fact that there are only seven different values of QE and QCEA, one concludes that there are a total of 49 different possible combinations, corresponding to 49 distinct rules. Therefore, considering the poor performance of the 36 rules, it was decided to use a control surface fully populated with all 49 different rules. This final approach produced acceptable behavior and did not hinder the computational time associated with the fuzzy logic controller. This is due to the fact that, given any pair of error and change in error values, only *four* of the 49 rules are applied at that given instance.

The step responses of the horizontal pendulum controlled by both a tradition PD controller and the Fuzzy-PD controller are illustrated in Figure 4.5. Both systems possess a similar rise times but the Fuzzy-PD controller has considerably less overshoot. This is due to the fact that the controller is based on *humanistic* rules or the rule base as opposed to a mathematical function governed by the damping envelope.

Since the system is second order, the PD gains could be specifically chosen to produce an identical step response to that demonstrated by the Fuzzy-PD controller. However, in order to provide an objective test environment, the maximum torque that either controller could apply to the system was set to 15 in-ounces. With this in mind Fuzzy-PD out performed the traditional PD controller.

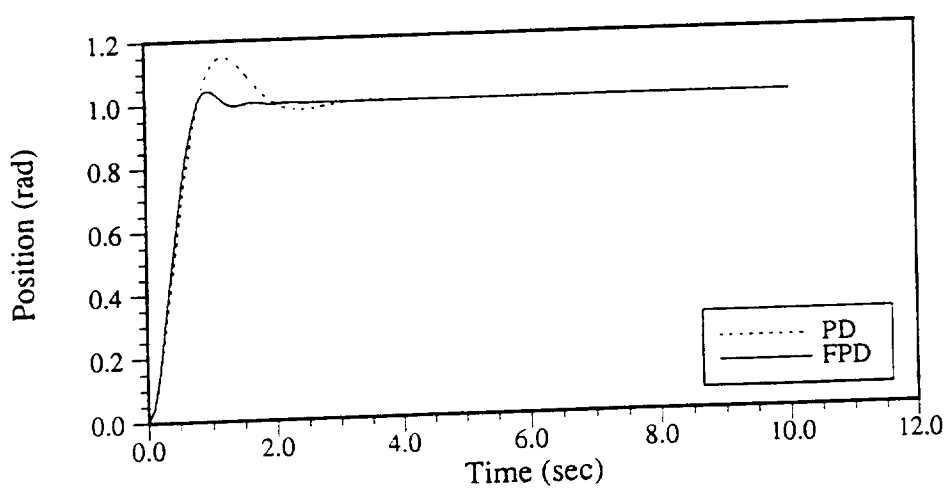


Figure 4.5: Step response of a traditional PD controller vs. Fuzzy-PD

With a working Fuzzy-PD controller now fully matured, consideration is given to how the individual rules can be formulated and how the Fuzzy-PD controller can be tuned.

4.1.4 POPULATION OF THE RULE BASE

The rule base was populated by simply allowing the quantized change in error angle to be set to zero and looking at how the system behaves as a function of only quantized error. For example, if the quantized error is "large positive" and the quantized change in error angle is "zero", then the quantized input to the system should be "large negative". This rule may be viewed graphically in Figure 4.6 where it is labeled as rule 4. The complement of rule 4 may be written as: If quantized error is "large negative" and the quantized change in error angle is "zero", then the quantized input to the system should be "large positive"(Fig. 4.6, rule 46). The intermediate rules may be found by interpolating between these two points. Extruding this slope, both in the positive and negative directions of the CEA axis results in an unacceptable controller surface. Therefore, the same technique may be applied to obtain the rules along the CEA axis. With both the error and change in error axes defined, one may then interpolate across all 49 different rules.

Looking at the quantized control input to the system as a function of quantized error and quantized change-in-error-angle, one may plot the surface as in Figure 4.6.

$$QI = \Im(QE, QCEA) \qquad \text{Eq.(4.10)}$$

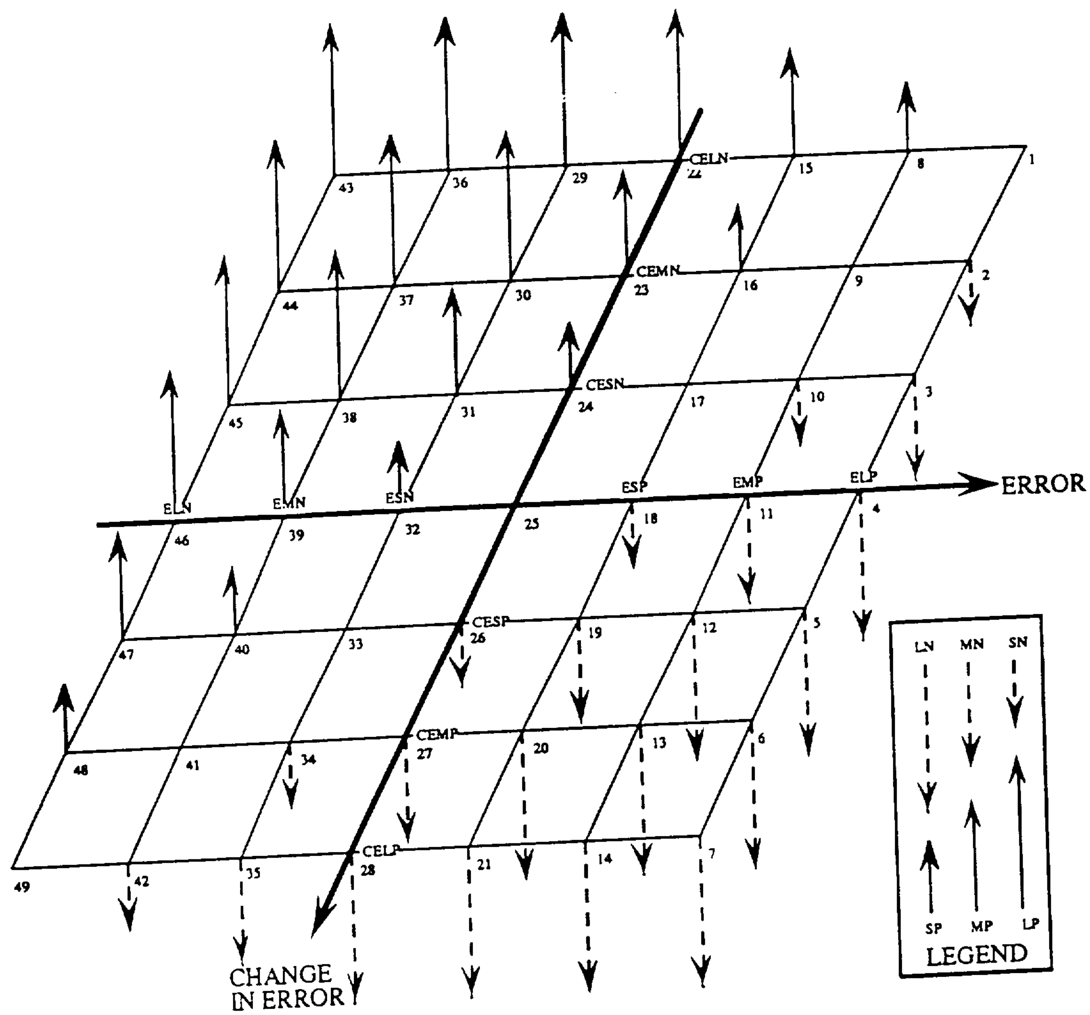


Figure 4.6: 49 rules used in the Fuzzy-Proportional-Derivative controller

Interpolating between the discrete values produces, a smooth fuzzy logic control surface.

4.1.5 TUNING

The second topic to be elaborated on before development of the more complicated Fuzzy-PID controller is a tuning scheme. There are numerous ways to tune a fuzzy controller. First, and most obvious, the rule base itself may be altered., Secondly, the shape of the membership function may be altered(sinusoidal, bell, trapezoid, etc. . .). Also, the overlap of the membership functions may be altered. And lastly, the quantizing schemes or functions may be also changed. Assuming that the rule base, the membership function, and the overlap of the membership function have been chosen appropriately, the only method of tuning seems to be to change the quantizing functions.

However, when trying to tune the quantizing functions, a problem arises. This problem is associated with the fact that the control input is a function of three quantized terms: the error, change in error angle, and the input. With no guidance governing the relationship between these three quantized values, it may be just as effective to randomly choose quantizing functions.

If two of the three quantization functions were constrained to certain limits, then there would remain only one independent variable to alter. Since, the error is bounded (between +60 and -60 degrees) and the change in error angle is bounded (the inverse tangent of the slope of the error curve lies between +90 and -90 degrees), the quantized control input-to-torque relationship remains the only quantization function that may be altered. Therefore, it stands to reason that the quantized input to the system is the only choice with which to tune the fuzzy controller.

As an off-line approach to tuning, the effect of varying the dequantization function for QI can be examined through a combination of the resulting step response in conjunction with its respective phase portrait. Figure 4.7 shows the system at eleven steps in time. From this, the quantized error and quantized CEA can be tabulated as in Table 4.1. A plot of the quantized error versus the quantized change in error angle may then be obtained; this is the phase portrait. Figure 4.8 illustrates the phase portrait for the step response illustrated in Figure 4.5 and 4.7.

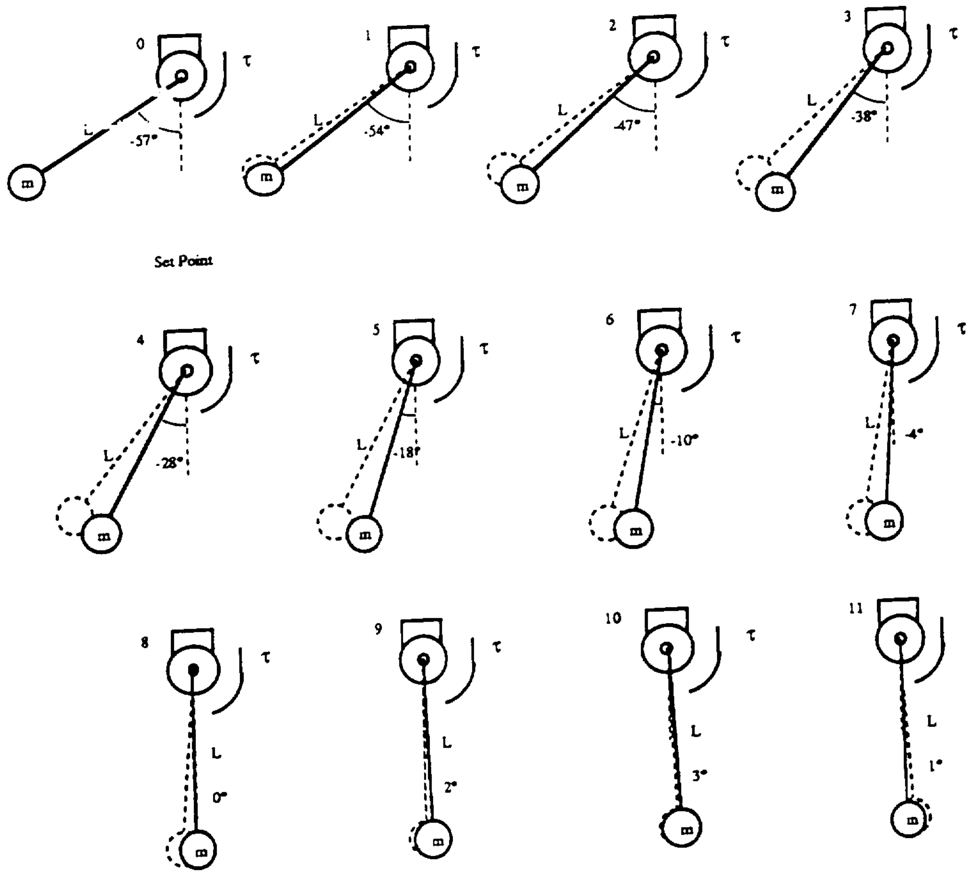


Figure 4.7: Pictorial representation of the Fuzzy-PD step response

Table 4.1: Control parameters for a unit step response

Time Step	E	CEA	QE	QCEA	QI
0	-57	0	-5.7	0	5.6
1	-54	24	-5.4	1.6	3.9
2	-47	50	-4.7	3.3	1.7
3	-38	58	-3.8	3.9	.0
4	-28	60	-2.8	4.1	-1.2
5	-18	59	-1.8	4.0	-2.0
6	-10	54	-1.0	3.6	-2.1
7	-4	47	-.4	3.2	-2.1
8	0	36	.3	2.5	-1.1
9	2	28	.3	1.3	-.0
10	3	17	.5	.3	.1
11	1	0	.5	-.4	0

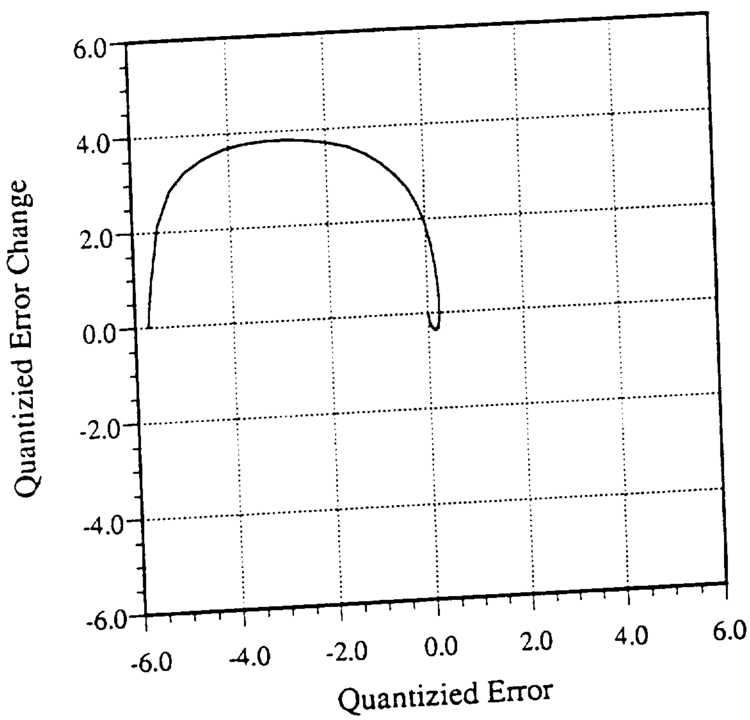


Figure 4.8: Phase portrait for the Fuzzy-PD step response

By varying the maximum torque (τ_{max}) in the dequantization function (see Eq. 4.8) between the values 2.5, 15, and 80 in-oz, one obtains three different time responses and three different phase portraits associated with these time responses. Figure 4.9(b) shows a phase portrait of an underdamped system when a maximum torque of 2.5 in-oz is applied. The corresponding time response of this system is given in Figure 4.9(a).

Figure 4.9(d) is a plot of a phase portrait when a maximum torque of 15 in. ounces is applied to the horizontal pendulum. The associated time response is illustrated in Figure 4.9(c). It may be noted that the phase portrait illustrated in Figure 4.9(d) corresponds to a quick rise time with a minimal amount of peak overshoot. Therefore, this phase portrait is the desired curve.

Figure 4.9(f) is a phase portrait for a maximum torque value of 80 in. ounces and Figure 4.9(e) is its corresponding time response. This particular system is over responsive. These three phase portraits and their corresponding time responses are critical to accomplishing tuning of the complicated dynamics associated with the three-link revolute manipulator.

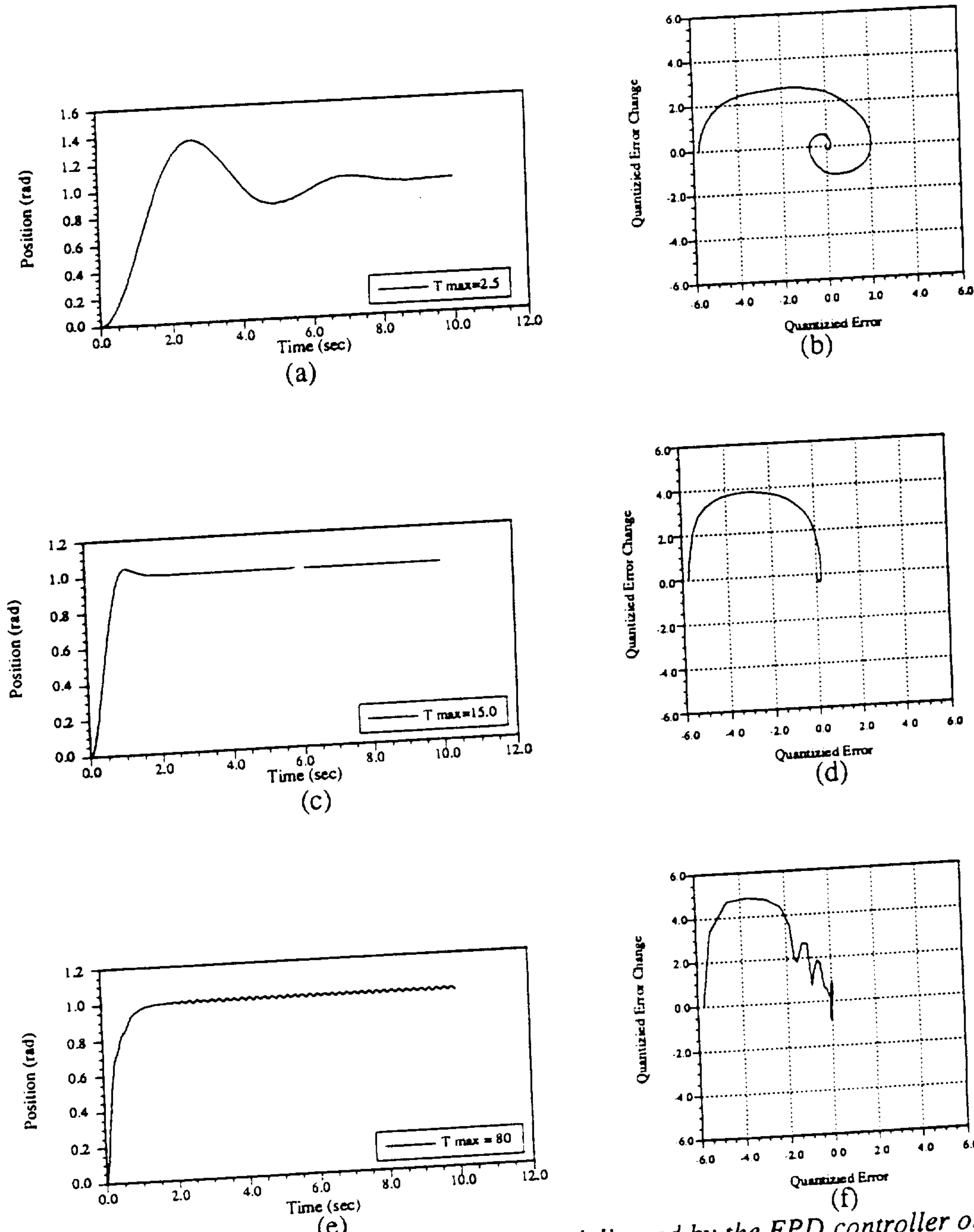


Figure 4.9: Effect of varying maximum torque delivered by the FPD controller on the step response and the associated phase portrait

4.1.6 VARYING INERTIA LOAD

Due to the varying task requirements associated with on orbit assemble issues it is of interest to investigate how the Fuzzy-PD controller compares to traditional PD control when the end-point mass varies. Figure 4.10 illustrates the step response of the horizontal pendulum when controlled by both Fuzzy-PD and traditional PD. Figure 4.10(a) illustrates the response when both controllers are tuned properly. Figures 4.10(b)-4.10(d) demonstrates that both controllers are approximately producing the same control response for a give situation. This is expected because the fuzzy rules discretely approximate the PD control surface.

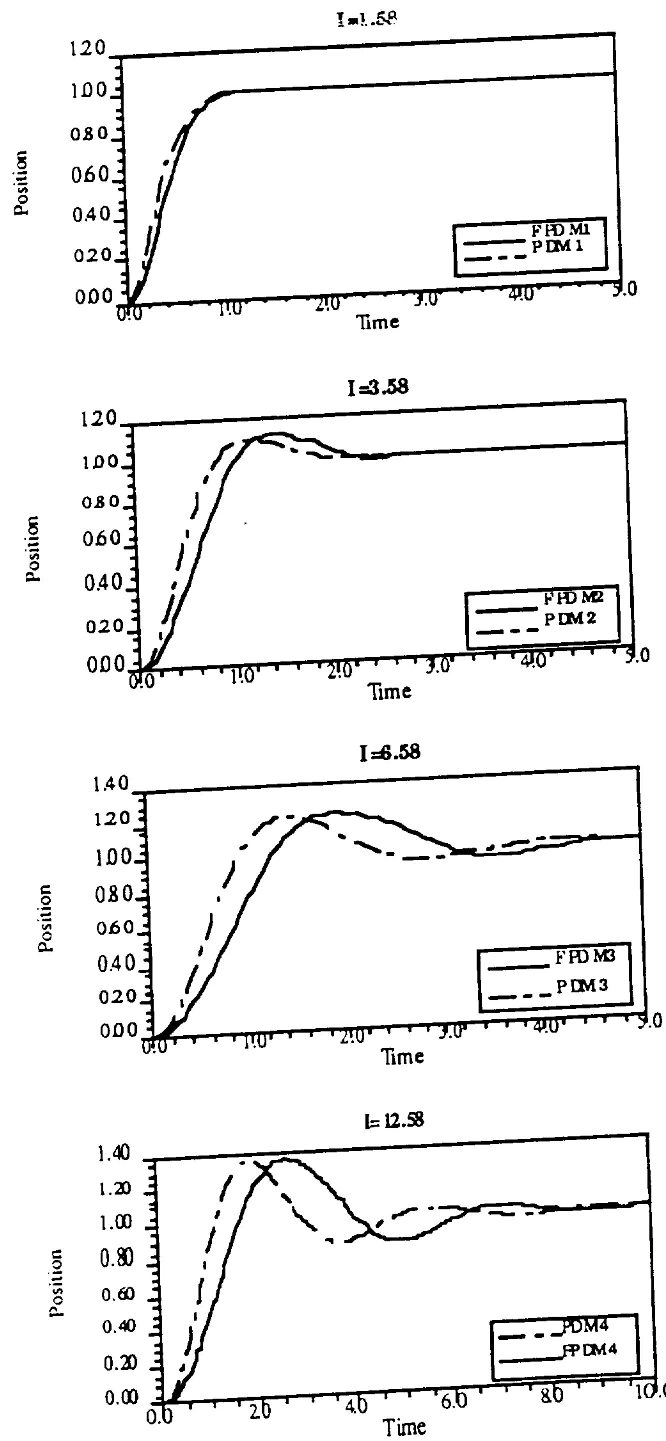


Figure 4.10 : Step response as end-point mass varies

4.1.7 TIME DELAYS

Another issue of particular interest is that of time delays. By tuning Fuzzy-PD and PD to the same performance characteristics and then introducing time delays a comparison of the robustness of each controller may be made. Figures' 4.11 and 4.12 illustrate the performance degradation associated with increasing the delay time. Figure 4.11(a) is the step response without any time delays. Figure 4.11(b) demonstrates that both controllers behave approximately the same with a time delay of one sample period (the sample period used here was .05 sec.). At a time delay of two sampling periods the Fuzzy-PD controller becomes marginally stable while the PD controller performs with slight indifference(Fig. 4.11(c)). Figure 4.12 illustrates that it is not until a five sampling period time delay that the PD controller becomes marginally stable. Figure 4.12(c) illustrates that as the time delay increase past 4 the system remains marginally stable with an increase in amplitude and a slower frequency.

The results of introducing time delays indicate that Fuzzy-PD control does not perform as well as traditional PD control. This is also an expected result due to the fact that the traditional PD controller is continuous while the Fuzzy-PD controller possesses a discrete number of rules.

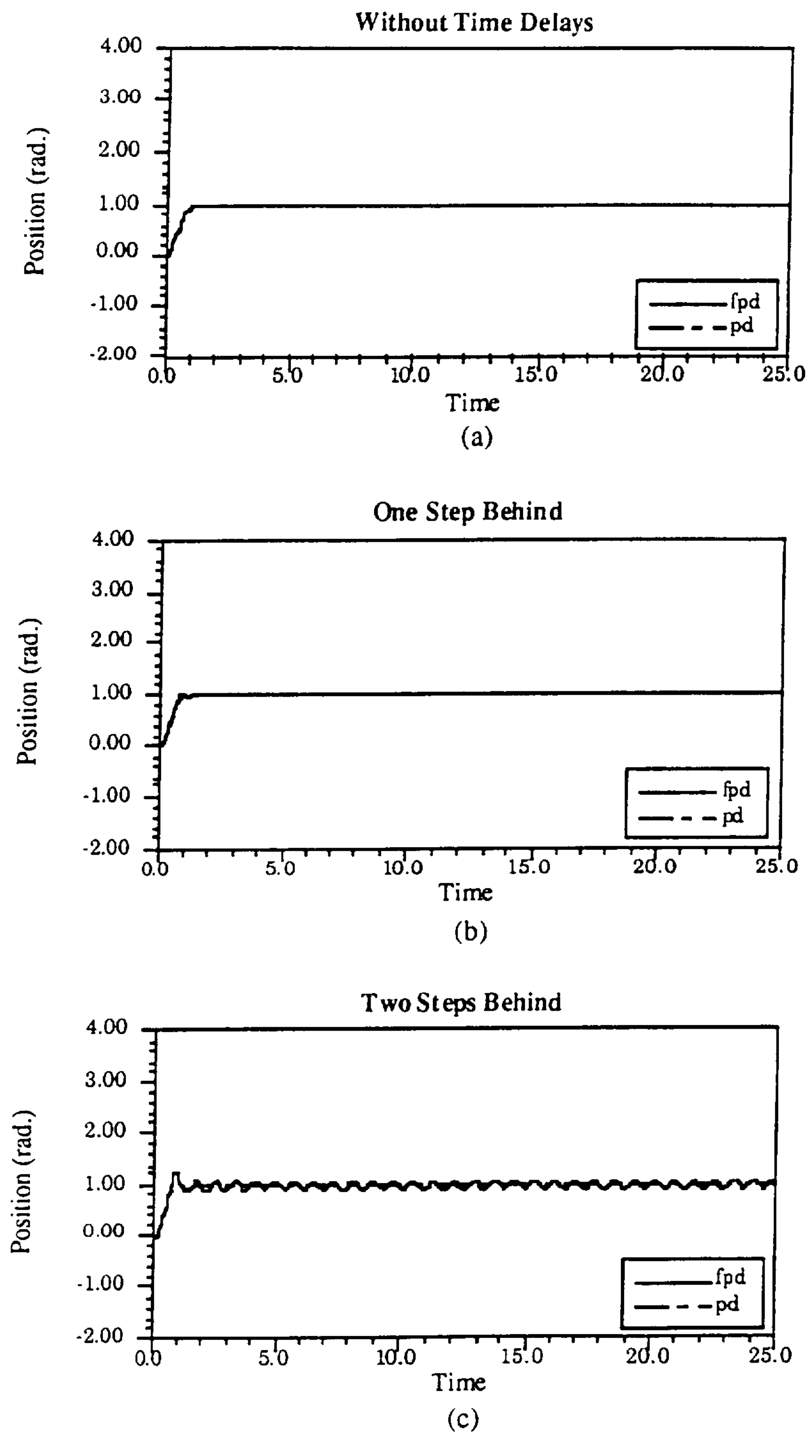


Figure 4.11: Step response as the time delay increases.

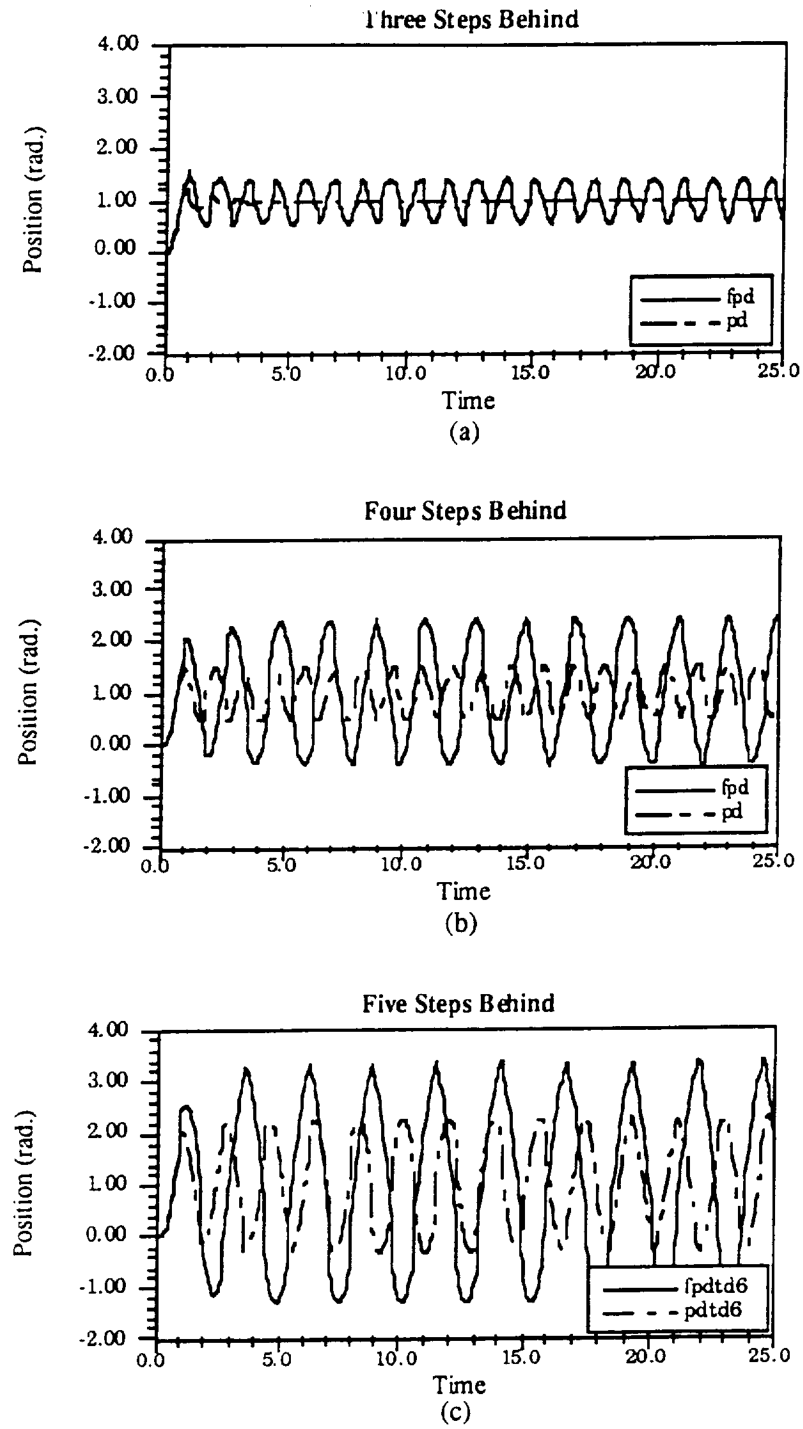


Figure 4.12: Step response as the time delay increases.

4.2 THE VERTICAL PENDULUM

As a second example, consider applying the Fuzzy-PD controller just developed to a second-order nonlinear system with a constant system bias. In particular, consider a traditional vertical pendulum. This pendulum possesses two equilibrium points. However, it does not operate about either one, thereby maintaining a constant system bias due to gravity. The dynamics associated with the pendulum are:

$$\ddot{\theta} + \frac{g}{L}\sin\theta = \tau \qquad \text{Eq.(4.11)}$$

As in the first illustration, to implement traditional proportional integral derivative control, consider applying the Theory of Natural Control[17], where,

$$\tau = -g(\theta - \theta_d) - h(\dot{\theta} - \dot{\theta}_d) - i\int_0^t (\theta - \theta_d)ds \qquad \text{Eq.(4.12)}$$

Linearizing Eq.(4.11) , substituting equation (4.12) into the linearized version of Eq.(4.11) and solving for the desired gains g, h, and i results in a PID controller which may then be applied to the nonlinear dynamics of Eq.(4.11).

NOTE: The term Fuzzy-PID represents the fuzzy controller that when applied to a system behaves like a tuned traditional PID controller.

4.2.1 APPLICATION OF FUZZY-PD ON THE VERTICAL PENDULUM

The first attempt at controlling this system using fuzzy logic is to apply the fuzzy controller developed for the horizontal pendulum to the dynamics of the vertical pendulum. The results of such an approach is illustrated in Figure 4.13 (the curve labeled "course"). The desired position in this case is one and the position obtained is approximately 0.425; therefore, there is a steady-state error.

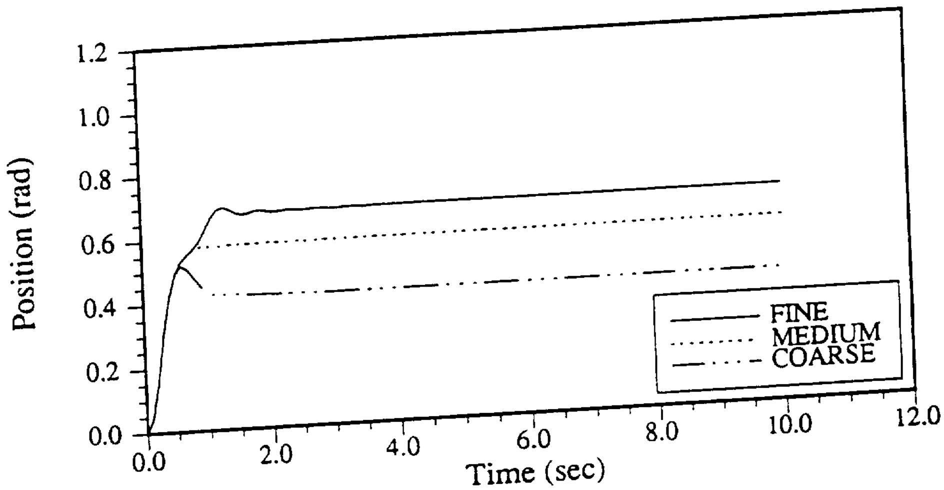


Figure 4.13: Application of the Fuzzy-PD controller to a traditional vertical pendulum

4.2.2 CAPTURE METHOD

In order to compensate for this steady-state error, a capture method was implemented. This capture method consists of redefining the quantized error function every time the error of the system falls within certain limits.

For example, if the error was originally quantized to a maximum value of plus or minus 60 degrees (coarse), when the error lies between ±30° the function could be redefined to have bounds of ±30°; this would be the medium quantization function. The same could be done for a fine quantization function by changing the bounds to ±15°. Figure 4.13 illustrates how this method succeeds in reducing the steady state error but never accomplishes the ultimate goal of no steady state error.

4.2.3 AN ALTERNATIVE RULE BASE FOR FPID

Due to the inability of the Fuzzy-PD controller to deal with the steady-state error, a second method using an altered rule base was developed. Figure 4.14 illustrates the new rule base.

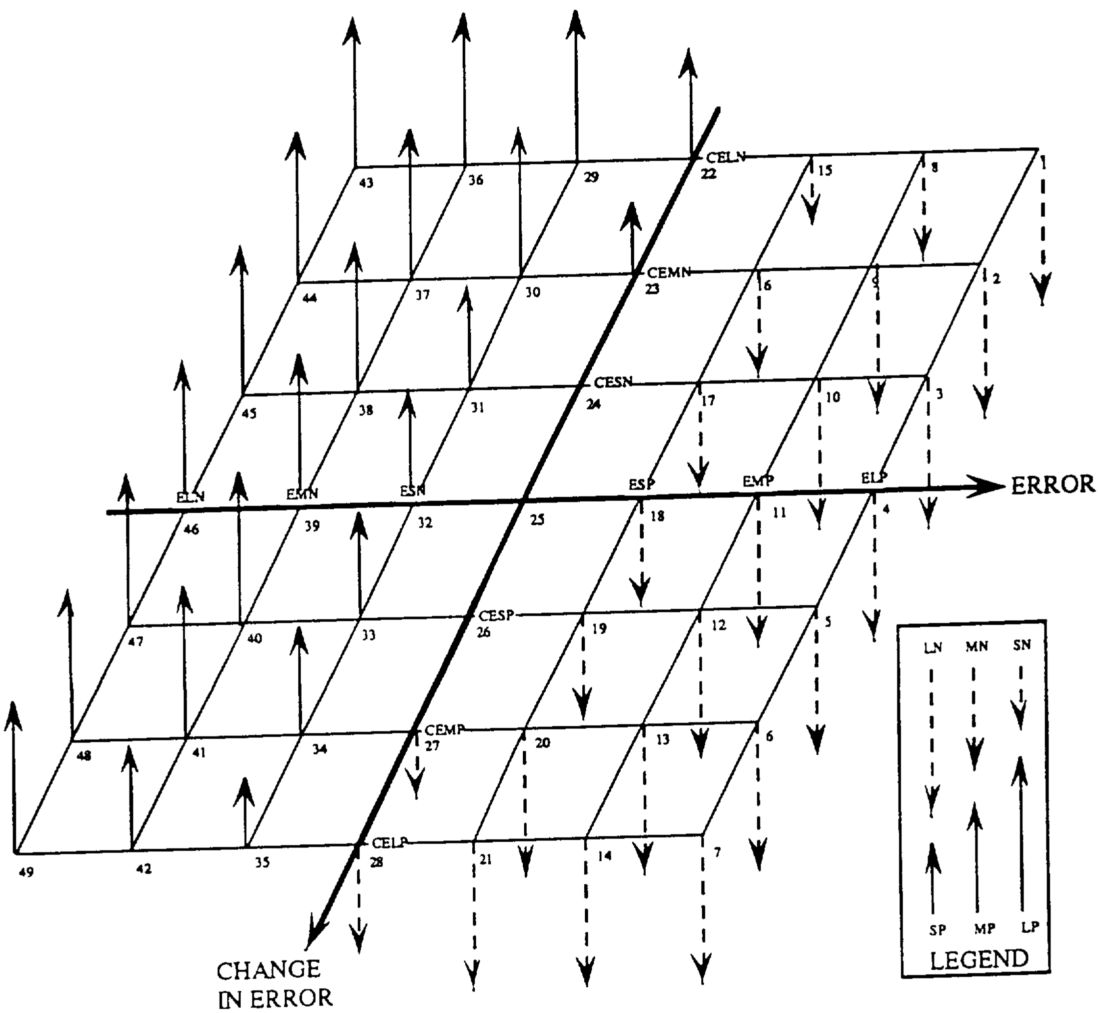


Figure 4.14: A candidate rule base for a Fuzzy-PID controller

The motivation for defining this new rule base came from observing the phase portraits of the time responses illustrated in Figure 4.13. It was noticed in these phase portraits that the portion of the original 49 rule base that compensated for overshoot was producing a steady-state error in the time response.

Therefore, the rules that compensated for overshoot were eliminated; thus the rule base illustrated in Figure 4.14 was assembled. The time response of the vertical pendulum to the new rules is illustrated in Figure 4.15.

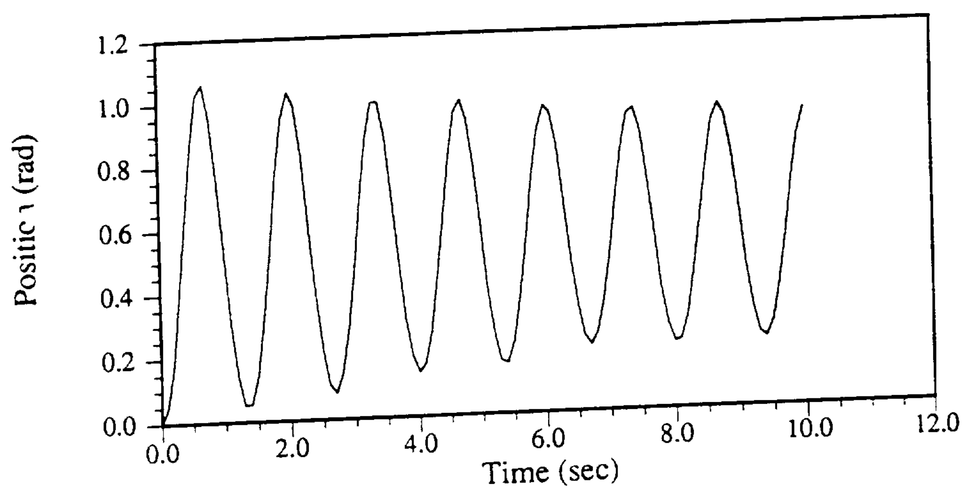


Figure 4.15: Step response of Fuzzy-PID controller with an alternative rule base

It is noticed that this system has a quick rise time; however once the system reaches its desired position, the control input produces no torque. This results in a rapid drop in the vertical pendulum back towards it equilibrium position. This cycle repeats itself ultimately resulting in a marginally stable system. Therefore, the method of altering the rule base to compensate for system biases fails and another method is needed.

4.2.4 HYBRID FUZZY-PD AND TRADITIONAL INTEGRAL CONTROL

In order to compensate for steady-state errors, a fuzzy integral control parameter may be augmented to the existing Fuzzy-PD controller. Due to the complexity associated with defining the necessary 343 rules, a hybrid controller was designed. This hybrid controller uses fuzzy proportional-derivative control (FPD) and classical integral control to produce a fuzzy-proportional-integral-derivative controller (FPID).

$$\tau = \tau_{FPD} + i\int_0^t (\theta - \theta_d)\, ds \qquad \text{Eq.(4.13)}$$

The Fuzzy-PD controller handles the vibration suppression as demonstrated earlier and the integral term compensates for any system bias.

The time responses of the vertical pendulum the Fuzzy-PID controller and traditional PID control are shown in Figure 4.16.

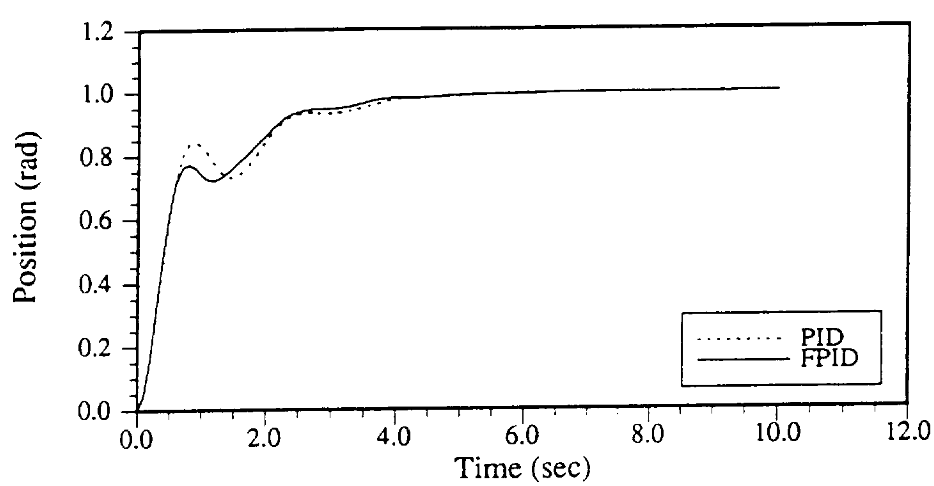


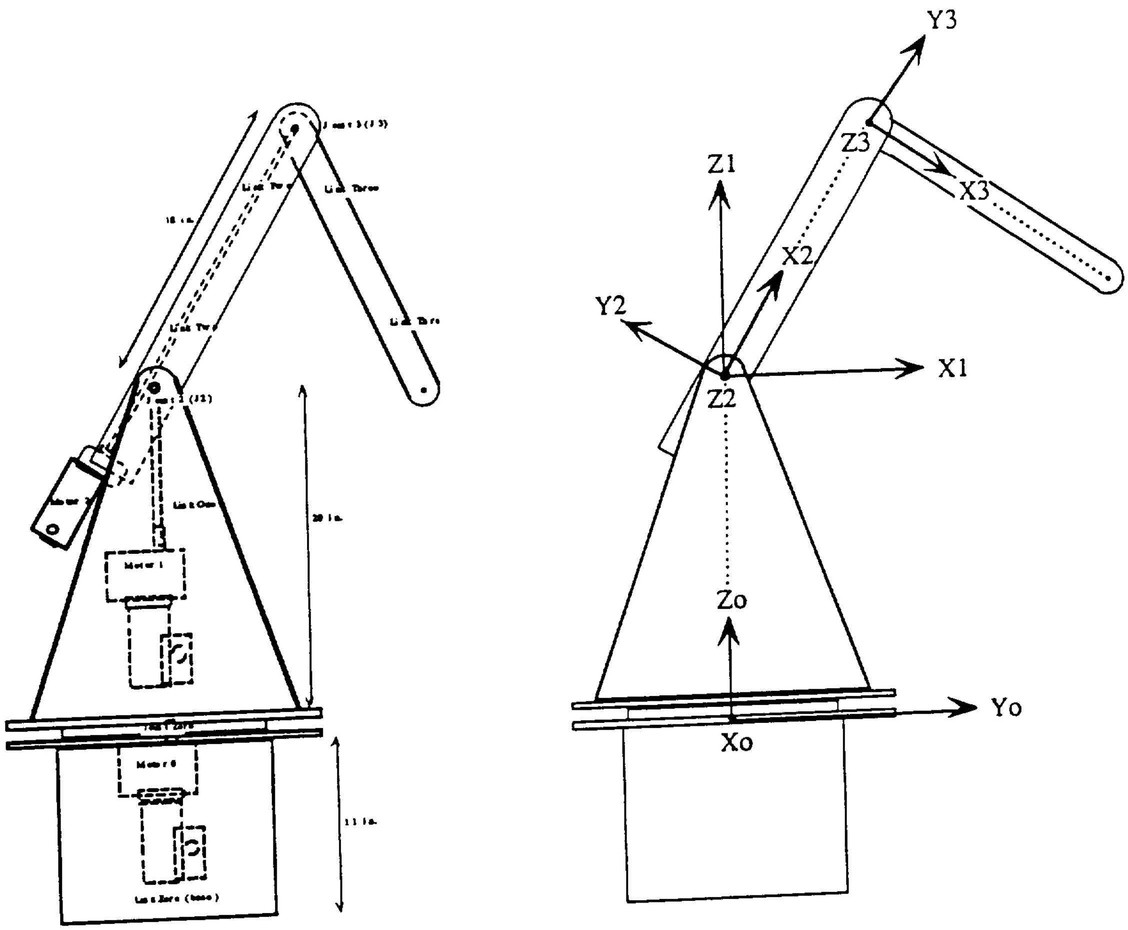
Figure 4.16: Step response of traditional PID and Fuzzy-PID

This hybrid Fuzzy-PID controller not only suppresses vibration but also compensates for steady-state error. The time response of the Fuzzy-PID controller compared to that of a tuned traditional PID controller is hardly distinguishable. With the hybrid Fuzzy-PID controller now developed, the approach can be applied to robotics systems. The dynamics of the 3-link manipulator are investigated in the next chapter.

5. APPLICATION OF FUZZY-PID TO A 3-DOF MANIPULATOR

With the hybrid fuzzy controller developed, its applicability to a robotics system can be investigated. A telerobotic flexible manipulator system is currently being developed at the Mars Mission Research Center in order to investigate several control algorithms for real-time implementation. A model of one of the robotic manipulator arms is selected for preliminary studies of the fuzzy control algorithm.

Figure 5.1(a) is a drawing illustrating the (DR-106) three-degree-of-freedom revolute manipulator being constructed at North Carolina State University. Figure 5.1(b) illustrates the coordinate axis definition.



(a) (b)

Figure 5.1: (a) DR-106 manipulator under construction at MMRC / NCSU
(b) Coordinate axes

5.1 DYNAMICS

The nonlinear dynamics for the three-degree-of-freedom revolute manipulator can be developed using the Lagrangian approach[22]. This results in

$$\begin{bmatrix} 26+8C_{23}^2+29C_2^2+24C_2C_{23} & 0 & 0 \\ 0 & 43+24C_3 & 8+12C_3 \\ 0 & 8+12C_3 & 8 \end{bmatrix}\begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3\end{bmatrix} =$$

$$\begin{bmatrix} (16C_{23}S_{23}+24C_2S_{23})\dot{\theta}_1(\dot{\theta}_2+\dot{\theta}_3)+(24S_2C_{23}+58S_2C_2)\dot{\theta}_1\dot{\theta}_2 \\ -(8C_{23}S_{23}+12S_2C_{23}+12C_2S_{23}+29C_2S_2)\dot{\theta}_1^2+24S_3\dot{\theta}_2\dot{\theta}_3+12S_3\dot{\theta}_3^2-20gC_2-6gC_{23} \\ -(8C_{23}S_{23}+12C_2S_{23})\dot{\theta}_1^2-12S_3\dot{\theta}_2^2-6gC_{23} \end{bmatrix}+$$

$$\begin{bmatrix}\tau_1 \\ \tau_2 \\ \tau_3\end{bmatrix} \qquad \text{Eq.(5.1)}$$

Eq.(5.1) may be written more compactly as

$$M\ddot{\underline{\theta}} = \underline{R}+\underline{T} \qquad \text{Eq.(5.2)}$$

where

M = mass matrix
$\ddot{\underline{\theta}}$ = acceleration vector
$\underline{R}$ = vector containing nonlinear terms
$\underline{T}$ = torque vector

Premultipling Eq.(5.2) by M^{-1}

$$\ddot{\underline{\theta}} = M^{-1}\underline{R}+M^{-1}\underline{T} \qquad \text{Eq.(5.3)}$$

Equation (5.3) may now be written in state space form and numerically integrated to find $\theta(t)$.

To apply proportional-integral-derivative control to the system, the appropriate relationship between the torque vector and the position vector must be found. Once again the methods developed in Natural Control Theory [17] will be utilized. First, make a linear approximation by simply dropping the nonlinear terms in Eq. (5.2). This results in

$$M\ddot{\underline{\theta}} = \underline{T} \qquad \text{Eq.(5.4)}$$

Secondly, assume

$$\underline{T} = -G\dot{\underline{\theta}} - H\underline{\theta} - I\int_0^t \underline{\theta}dt \qquad \text{Eq(5.5)}$$

where G, H, and I are control gain matrices. Letting

$$\begin{aligned} G &= gM \\ H &= hM \\ I &= iM \end{aligned} \qquad \text{Eqs.(5.6)}$$

Substitution of Equations (5.6) into Eq.(5.5) results in the following torque vector:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = -\begin{bmatrix} gm_{11}x(2) + hm_{11}x(3) \\ g[m_{22}x(5) + m_{23}x(8)] + h[m_{22}x(6) + m_{23}x(9)] + i[m_{22}x(4) + m_{23}x(7)] \\ g[m_{23}x(5) + m_{33}x(8)] + h[m_{23}x(6) + m_{33}x(9)] + i[m_{23}x(4) + m_{33}x(7)] \end{bmatrix} \qquad \text{Eq.(5.7)}$$

where $x(i)$ is the i^{th} state defined as:

$$\begin{aligned} x(1) &= \int_0^t \theta_1 - \theta_{1d}ds \\ x(2) &= \theta_1 - \theta_{1d} \\ x(3) &= \dot{\theta}_1 \end{aligned} \qquad \text{Eq.(5.8)}$$

$$
\begin{aligned}
x(4) &= \int_0^t \theta_2 - \theta_{2d} ds \\
x(5) &= \theta_2 - \theta_{2d} \\
x(6) &= \dot{\theta}_2
\end{aligned}
\qquad \text{Eq.(5.9)}
$$

$$
\begin{aligned}
x(7) &= \int_0^t \theta_3 - \theta_{3d} ds \\
x(8) &= \theta_3 - \theta_{3d} \\
x(9) &= \dot{\theta}_3
\end{aligned}
\qquad \text{Eq.(5.10)}
$$

Figure 5.2 illustrates how the dynamics of the system behaves. With no torque applied the links vibrate freely. Figure 5.3 provides a graphical representation of how links 2 and 3 behave when the links are released from a horizontal position and allowed to move freely. Gravity is acti: g in this system and the two links oscillate about their respective equilibrium points.

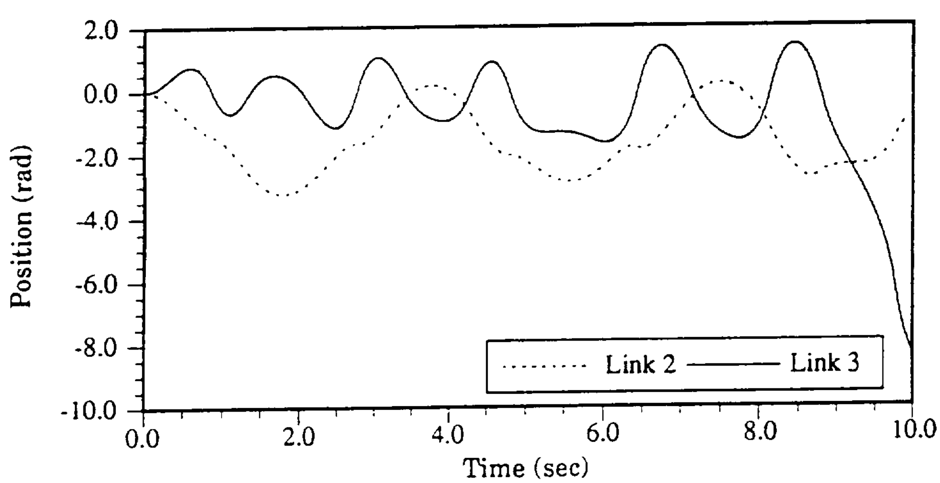


Figure 5.2: Free vibration of links 2 and 3

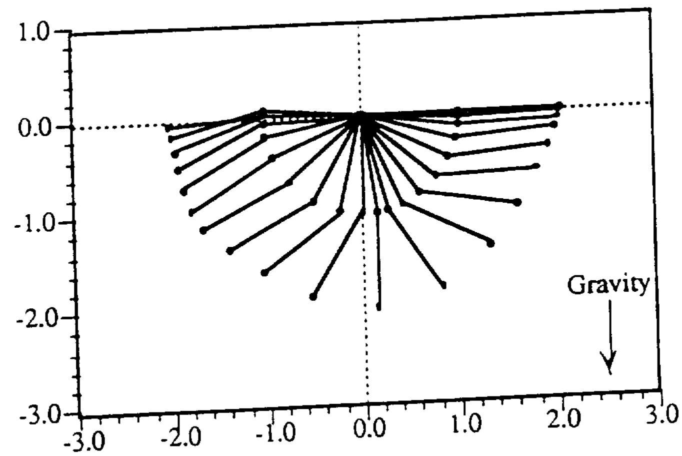


Figure 5.3: Graphical representation of the free vibration of links 2 and 3

5.2 TRADITIONAL PID VS. FUZZY-PID

The fuzzy-PID hybrid controller applied to the vertical pendulum will now be applied to this manipulator. Each link of the manipulator is independently controlled by a separate fuzzy logic control algorithm. Each of the three links are subjected to a unit step forcing function; links 1 and 2 have positive unit steps and link 3 has a negative unit step. Figure 5.4 shows the response of the first link using a FPID hybrid and a traditional PID controller. The step response shown in this figure is the robot's maneuver in the horizontal plane of rotation.

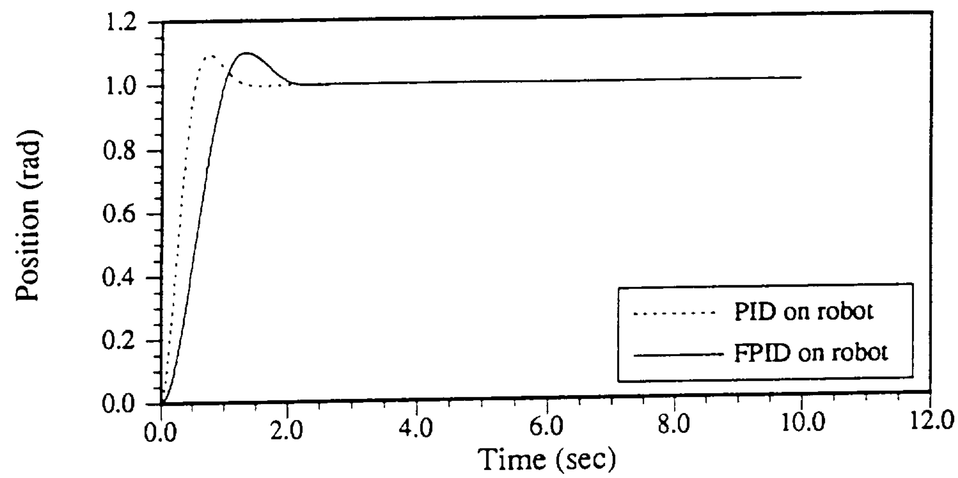


Figure 5.4: Step response of link one

Figure 5.5 illustrates the response of the second link of the manipulator. This graph illustrates a slower rise time for the fuzzy-PID controller as compared to the tradition PID. However, the fuzzy-PID controller has considerably less overshoot then PID.

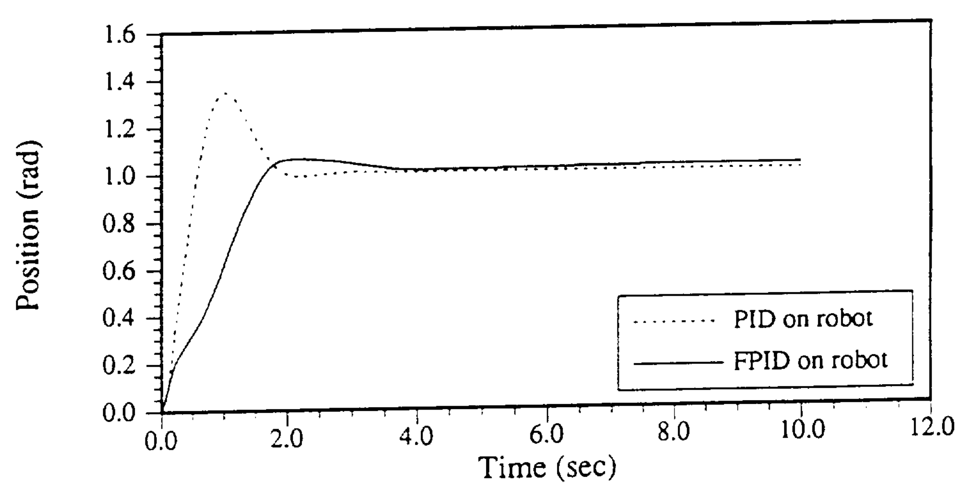


Figure 5.5: Step response of link two

Figure 5.6 shows the response of the robot's link 3. This graph is the most dramatic of the three links in terms of the difference between PID and FPID. The fuzzy logic controller not only produces a quicker rise time but also exhibits hardly any overshoot as compared to the traditional PID controller.

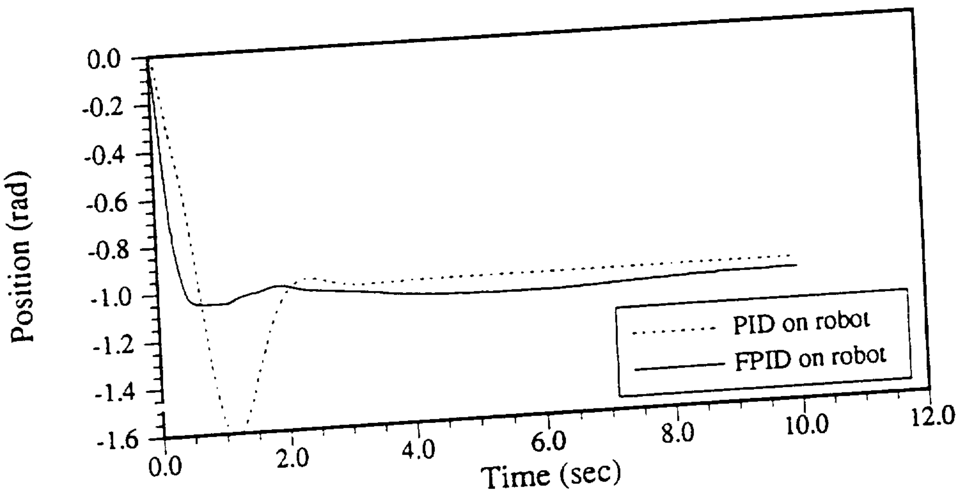


Figure 5.6: Negative step response of link three

The phase portraits associated with these step maneuvers are Figures 5.7, 5.8 and 5.9, respectively. Collectively these figures illustrate that the fuzzy-PID control provides a better or equivalent time response than classical PID control.

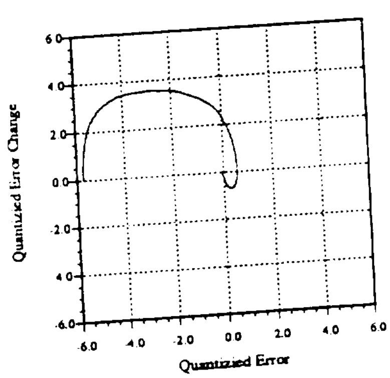


Figure 5.7: Phase portrait of link one

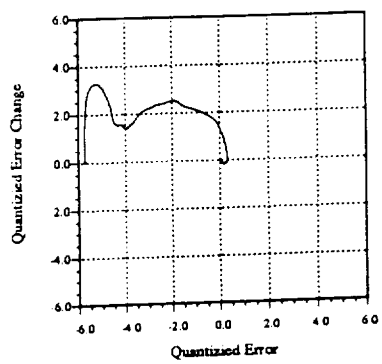


Figure5.8: Phase portrait of link two

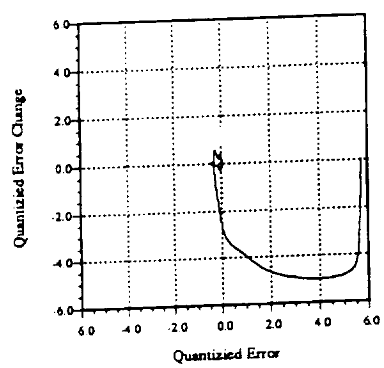


Figure 5.9: Phase portrait of link three

6. CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

This thesis presented a fuzzy-PD controller made up of 49 rules with a sinusoidal membership function. This general fuzzy-PD controller was then augmented with traditional integral control to produce a fuzzy-PID controller. The fuzzy-PID controller was then used to control nonlinear robotics models. Fuzzy-PID showed promise when compared to traditional PID due to PID's requirement of a model and the complexity associated with developing the gains. Because robotics models, including space robotics systems, contain uncertainties, exact model-based controllers are difficult to implement; hence the fuzzy approach may be more appealing.

Precise response characteristics using fuzzy controllers may be difficult, however. In [5] Cela and Hamam present some stability issues associated with fuzzy control systems. Although exact tuning of such systems may be difficult, this study has shown that this process is less difficult than PID gain tuning. Phase portraits provide a feasible off-line tuning approach.

The use of a fully populated quantized error and quantized change in error angle rule base provides a more effective controller while not hindering the processing time of the controller. Further, utilization of the inverse target function provides a novel means with which to bound the quantized change in error angle term.

Several issues may be considered as future activities, in order to extend the technique to space robotics systems and in particular, the ground testbed being constructed at North Carolina State University. First, fuzzy logic control may be constructed in such a fashion so as to tune a traditional PID controller. This approach would result in an adaptive controller that could compensate for varying inertia loads, and time delays; ultimately resulting in an ideal candidate for space applications.

Secondly, other calibration techniques including on-line methods are being considered. These methods include requantizing other variables besides the control input. Such extensions will be investigated and compared to other control methods being developed and implemented for space robotics systems.

7. REFERENCES

1. Batur, C. and Kasparian, V., "Adaptive Expert Control," *Journal of Control,* Vol 54, No. 4, October 1991, pp. 867-881.

2. Chang, C. H., "Tuning Fuzzy Logic Controllers via Input and Output Mapping Factors", M.S. Thesis, University of Oklahoma, 1989.

3. Chen,Y. Y. and Jang, J. S., "Imitation of State Feedback Controllers by Fuzzy Linguistic Control Rules," *IEEE*, Vol 3, December 1990 pp. 1545,1546.

4. Craig, J. J., Hsu, P.,.and Sastry, S., "Adaptive Control of Mechanical Manipulators", *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, San Francisco, 1986.

5. "Fuzzy Controller Robots and Its Practical Applications (Special Session)", IEEE/RSJ Int'l Conf. on Intelligent Robot. and Systems, Raleigh, 1992.

6. Huang, L. J. and Tomizuka, M., "A Self-Paced Fuzzy Tracking Controller for Two-Dimensional Motion Control," *IEEE Transactions on System, Man, and Cybernetics*, Vol. 20, No. 5, September/October 1990, pp. 1115-1123.

7. Jager, R. and Verbruggen, P. M., "Real-Time Fuzzy Expert Control," *IEE,* Vol. 2, No. 332, 1991, pp. 966-970.

8. King, P. J., and Mamdani, E. H., "The Application of Fuzzy Control Systems to Industrial Processes," *Automatica*, Vol. 13, pp. 235-242, 1977.

9. Kouatli, I. and Jones, B. "An Improved Design Procedure For Fuzzy Control Systems," *International Journal of Machine Tools and Manufacture*, Vol. 31, No. 1, January 1991, pp. 107-122.

10. Kwok, D. P. and Tam, C. K., "Analysis and Design of Fuzzy PID Control Systems," *IEE*, Vol. 2, No. 332, 1991, pp. 956-960.

11. Li, Y. F. and Lau, C. C. "Development of Fuzzy Algorithms for Servo Systems," *IEEE Control Systems Magazine*, April 1989, pp. 65-72.

12. Linkend, D. A. and Abbod, M. F., "Self-organizing Fuzzy Logic Control for Real-Time Processes," *IEE*, Vol. 2, No. 332, 1991, pp. 971-976.

13. Mamdani, E. H. and Assilian, S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *Int. J. Man-Machine Studies*, Vol. 7, pp. 1-13, 1975.

14. Murakami, S. and Maeda, M., "Automobile Speed Control System Using a Fuzzy Logic Controller", *Industrial Applications of Fuzzy Control* (M. Sugeno, Ed.), pp. 105-123, North Holland, 1985.

15. Rubinstein M. F., "Tools for Thinking and Problem Solving", Prentice-Hall, Inc., New Jersey, 1986.

16. Scharf, E. M., Mandic, N. J., and Mamdani, E. H., "A Self-organizing Algorithm for the Control of a Robot Arm", *ISMM Conf. on Mini and Microcomputers and Their Applications*, San Antonio, 1983.

17. Silverberg, L. and Morton, M. "On the Nature of Natural Control," *Journal of Vibration, Stress, and Reliability in Design*, Vol. 111, October 1989, pp. 412-422.

18. Slotine, S. J. and Li, W., "On the Adaptive Control of Robotic Manipulators", *Int. J. of Robotics Research*, Vol. 6, No. 3, pp. 49-59, 1987.

19. Smith, S. M. and Comer, D. J. "Automated Calibration of a Fuzzy Logic Controller Using a Cell State Space Algorithm," *IEEE International Conference on Systems Man and Cybernetics*, August 1991, pp.18-28.

20. Sutton, R. and Jess, I. M., "Real-Time Application of a Self-Organizing Autopilot to Warship Yaw Control," *IEE*, Vol. 2, No. 332, 1991, pp 827-832.

21. Tzafestas, S. and Papanikolopoulos, N.P., "Incremental Fuzzy Expert PID Control," *IEEE Transactions on Industrial Electronics*, Vol. 37, No. 5, October 1990, pp. 365-371.

22. Wolovich, W. A., "Robotics: Basic Analysis and Design", CBS College Publishing, New York, 1987, pp. 288-297.

23. Yamaguchi, T. ,Tanabe, M., Kuriyama, K., and Mita, T., "Fuzzy Adaptive Control With an Associative Memory System," *IEE*, Vol. 2, No. 332, 1991, pp. 944-949.

24. Ying,H., Siler, W., and Buckley, J. J., "Fuzzy Control Theory: A Nonlinear Case," *The Journal of International Federation of Automatic Control*, Vol. 26, No. 3, May 1990, pp. 513-520.

25. Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE* Trans. on Systems, Man and Cybernetics, Vol. SMC-3, pp. 28-44, 1975.

26. Zhang, B. S. and Edmunds, J.M., "On Fuzzy Logic Controllers," *IEE*, Vol. 2, No. 332, 1991, pp. 961-965.

8. APPENDICES

8.1 PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROL ON 3-DOF MANIPULATOR

```
c234567
      PROGRAM PID ON ROBOT
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: To apply PID control to the nonlinear dynamics of    c
c        a three link microbot.                                     c
c                                                                   c
c     AUTHOR:  Robert J. Stanley II                                 c
c                                                                   c
c     DATE:  8/5/92                                                 c
c                                                                   c
c     VARIABLES:                                                    c
c        T#:  Torque applied to respective links                    c
c        Td#:  The desired angle of each link                       c
c        NEQ: Number of equations                                   c
c        NSTEP:  Number of times runga-kutta subroutine is called   c
c        DT:  Time interval delta T in rad/sec                      c
c        TIME:  Independent variable                                c
c        X(I):  Dependent variable                                  c
c        F(I):  State equations                                     c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      INTEGER COUNT,NEQ,NSTEP
      REAL*8 X(10),DT,TIME,T1,T2,T3,Td1,Td2,Td3
      COMMON /MYCOMM/ Td1,Td2,Td3
c      OPEN (8,FILE='pidr.dat',STATUS='unknown')
c      OPEN (9,FILE='pidrT.dat',STATUS='unknown')
      OPEN (8,FILE='PIDR1G.dat',STATUS='unknown')
      OPEN (9,FILE='PIDR2G.dat',STATUS='unknown')
15    FORMAT(1X,F6.2,1X,F8.4,1X,F10.4,1X,F8.4,1X,F10.4,1X,F8.4,1X
     +,F10.4)
25    FORMAT(1X,F6.2,1X,F10.4,1X,F10.4,1X,F10.4)

      NEQ=9
      DT=.01D0
      NSTEP=1000
      TIME=0.0D0
      X(1)=0.0
      X(2)=-1.0
      X(3)=0.0
      X(4)=0.0
      X(5)=-1.0
      X(6)=0.0
      X(7)=0.0
      X(8)=1.0
      X(9)=0.0
```

```
      Td1=1.0
      Td2=1.0
      Td3=-1.0

      CALL TORQUE(X,T1,T2,T3)
      WRITE(8,15)TIME,X(2)+Td1,T1,X(5)+Td2,T2,X(8)+Td3,T3
      WRITE(9,25)TIME,X(2)+Td1,X(5)+Td2,X(8)+Td3
      DO 100 COUNT=1,NSTEP
         CALL RUNGA(X,DT,NEQ,TIME)
         CALL TORQUE(X,T1,T2,T3)
         WRITE(8,15)TIME,X(2)+Td1,T1,X(5)+Td2,T2,X(8)+Td3,T3
         WRITE(9,25)TIME,X(2)+Td1,X(5)+Td2,X(8)+Td3
100   CONTINUE
      CLOSE(8)
      END

      SUBROUTINE TORQUE(X,T1,T2,T3)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: Given the position calculate the torque using PID         c
c        control.                                                        c
c                                                                        c
c     AUTHOR: Robert J. Stanley II                                       c
c                                                                        c
c     DATE: 8/5/92                                                       c
c                                                                        c
c     VARIABLES:                                                         c
c        X: State vector                                                 c
c        Td1: Theta Desired One                                          c
c        Td2: Theta Desired Two                                          c
c        Td3: Theta Desired Three                                        c
c        T1: Torque applied to link one                                  c
c        T2: Torque applied to link Two                                  c
c        T3: Torque applied to link Three                                c
c        M##: The respective elements of the Mass Matrix                 c
c        H#: The derivative gain                                         c
c        G#: The portional gain                                          c
c        I#: The integral gain                                           c
c        ALPHA: The exponential decay rate                               c
c        BETA: The operating frequency                                   c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      REAL*8 X(10),Td1,Td2,Td3
      REAL*8 T1,T2,T3,M11,M22,M33,M23,C23,S23,C2,C3,S2
      REAL*8 S3,C23S23,C2S23,S2C23,S2C2,C2S2
      REAL*8 H1,H2,H3,G1,G2,G3,I1,I2,I3,ALPHA,BETA
      COMMON /MYCOMM/ Td1,Td2,Td3

      ALPHA=2.46051702
      BETA=3.14
```

```
      G1=3*ALPHA**2+BETA**2
      H1=3*ALPHA
      I1=0.0
      G2=3*ALPHA**2+BETA**2
      H2=3*ALPHA
      I2=ALPHA*(ALPHA**2+BETA**2)
      G3=3*ALPHA**2+BETA**2
      H3=3*ALPHA
      I3=ALPHA*(ALPHA**2+BETA**2)

      C23=COS(X(5)+Td2+X(8)+Td3)
      S23=SIN(X(5)+Td2+X(8)+Td3)
      C2=COS(X(5)+Td2)
      C3=COS(X(8)+Td3)
      S2=SIN(X(5)+Td2)
      S3=SIN(X(8)+Td3)
      C23S23=C23*S23
      C2S23=C2*S23
      S2C23=S2*C23
      S2C2=S2*C2
      C2S2=C2*S2

      M11=26.+8.*C23**2+29.*C2**2+24.*C2*C23
      M22=43.+24.*C3
      M23=8.+12.*C3
      M33=8.

      T1=-1*(G1*M11*X(2)+H1*M11*X(3)+I1*M11*X(1)) T2=-
      1*(G2*(M22*X(5)+M23*X(8))+H2*(M22*X(6)+M23*X(9))
     ++I2*(M22*X(4)+M23*X(7))) T3=-
      1*(G3*(M23*X(5)+M33*X(8))+H3*(M23*X(6)+M33*X(9))
     ++I3*(M23*X(4)+M33*X(7)))
      RETURN
      END

      SUBROUTINE RIGHT(R1,R2,R3,X,M11,M22,M33,M23,DET)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: To calculate the respective mass matrix entries and      c
c          the respective nonlinear contributions R#.                   c
c                                                                       c
c     AUTHOR:  Robert J. Stanley II                                     c
c                                                                       c
c     DATE:  8/5/92                                                     c
c                                                                       c
c     VARIABLES:                                                        c
c          R#:  The nonlinear terms of link # respectively              c
c          F:  State space                                              c
c          G:  Gravity                                                  c
c          DET:  The determinate of the mass matrix divided by M11      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      REAL*8 X(10),F(10),TIME,Td1,Td2,Td3
      REAL*8 T1,T2,T3,R1,R2,R3,M11,M22,M33,M23,G,C23,S23,C2,C3,S2
      REAL*8 S3,C23S23,C2S23,S2C23,S2C2,C2S2,DET
      COMMON /MYCOMM/ Td1,Td2,Td3

      G=9.8
      C23=COS(X(5)+Td2+X(8)+Td3)
      S23=SIN(X(5)+Td2+X(8)+Td3)
      C2=COS(X(5)+Td2)
      C3=COS(X(8)+Td3)
      S2=SIN(X(5)+Td2)
      S3=SIN(X(8)+Td3)
      C23S23=C23*S23
      C2S23=C2*S23
      S2C23=S2*C23
      S2C2=S2*C2
      C2S2=C2*S2

      M11=26.+8.*C23**2+29.*C2**2+24.*C2*C23
      M22=43.+24.*C3
      M23=8.+12.*C3
      M33=8.
      DET=M22*M33-M23*M23

      CALL TORQUE(X,T1,T2,T3)
      R1=(16.*C23S23+24.*C2S23)*X(3)*(X(6)+X(9))
     ++(24.*S2C23+58.*S2C2)*X(3)*X(6)+T1

        R2=-(8.*C23S23+12.*S2C23+12.*C2S23+29.*C2S2)*X(3)**2
       ++24.*S3*X(6)*X(9)+12.*S3*X(9)**2-20.*G*C2-6.*G*C23+T2

      R3=-(8.*C23S23+12.*C2S23)*X(3)**2-12.*S3*X(6)**2-6.*G*C23+T3
      RETURN
      END


      SUBROUTINE STATE(F,X,TIME)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: To compute the present state of the dynamic system.       c
c                                                                         c
c     AUTHOR:  Robert J. Stanley II                                       c
c                                                                         c
c     DATE:  8/5/92                                                       c
c                                                                         c
c     VARIABLES:                                                          c
c         All variables already defined.                                  c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      REAL*8 X(10),F(10),TIME,Td1,Td2,Td3
      REAL*8 R1,R2,R3,M11,M22,M33,M23,DET
      COMMON /MYCOMM/ Td1,Td2,Td3
```

```
      TIME=TIME*1.0

      CALL RIGHT(R1,R2,R3,X,M11,M22,M33,M23,DET)
      F(1)=X(2)
      F(2)=X(3)
      F(3)=R1/M11
      F(4)=X(5)
      F(5)=X(6)
      F(6)=(R2*M33/DET)-(R3*M23/DET)
      F(7)=X(8)
      F(8)=X(9)
      F(9)=-(R2*M23/DET)+(R3*M22/DET)

      RETURN
      END

      SUBROUTINE RUNGA(X,DT,NEQ,TIME)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: Use a Runge Kutta routine to compute the next state      c
c          vector                                                        c
c                                                                        c
c     AUTHOR:  Robert J. Stanley II                                      c
c                                                                        c
c     DATE:  8/5/92                                                      c
c                                                                        c
c     VARIABLES:                                                         c
c          G#:  Variable gains                                          c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        REAL*8 X(10),Y(10),F(10),DT,TIME,G1(10),G2(10),G3(10),G4(10)
      INTEGER I,NEQ

      DO 1 I=1,NEQ
1         Y(I)=X(I)

      CALL STATE(F,Y,TIME)
      DO 2 I=1,NEQ
2         G1(I)=DT*F(I)

      TIME=TIME+DT/2.0D0
      DO 3 I=1,NEQ
3         Y(I)=X(I)+G1(I)/2.0D0

      CALL STATE(F,Y,TIME)
      DO 4 I=1,NEQ
          G2(I)=DT*F(I)
4         Y(I)=X(I)+G2(I)/2.0D0
      CALL STATE(F,Y,TIME)
      DO 5 I=1,NEQ
          G3(I)=DT*F(I)
```

```
5       Y(I)=X(I)+G3(I)
      TIME=TIME+DT/2.0D0
      CALL STATE(F,Y,TIME)
      DO 6 I=1,NEQ
6       G4(I)=DT*F(I)
      DO 7 I=1,NEQ
7       X(I)=X(I)+(G1(I)+2.0D0*(G2(I)+G3(I))+G4(I))/6.0D0
      RETURN
      END
```

8.2 FUZZY LOGIC CONTROL ON 3-DOF MANIPULATOR

```
c234567
      PROGRAM FUZZY ON ROBOT
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE:  To apply a hybrid of Fuzzy Logic and traditional    c
c         integral feedback to a three link microbot. (Highly        c
c         Non-linear Coupled Second Order Differential Equations)    c
c                                                                    c
c     AUTHOR: Robert J. Stanley II                                   c
c                                                                    c
c     DATE: 8/4/92                                                   c
c                                                                    c
c     VARIABLES:                                                     c
c         COUNT: Holds the value of the present Runge-Kutta iteration c
c         TORQUE: The input to the system                            c
c         INERTIA: The inertia of the system                         c
c         Wn: The natural frequency of the system                    c
c         Td: The desired position  (Theata Desired)                 c
c         TRIGGER: Zero on the first pass and One afterwards         c
c         NEQ: Number of equations                                   c
c         NSTEP:  Number of times runga-kutta subroutine is called   c
c         DT:  Time interval delta T in rad/sec                      c
c         TIME:  Independent variable                                c
c         X(I):  Dependent variable                                  c
c         F(I):  State equations                                     c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      INTEGER COUNT,NEQ,NSTEP
      REAL*8 X(10),DT,TIME,C2,C3,C23,M11,M22,M23,M33
      REAL*8 T1,T2,T3,Td1,Td2,Td3,I2,I3,ALPHA,BETA
      INTEGER TRIG1,TRIG2,TRIG3,L1,L2,L3
      COMMON /MYCOMM/ T1,T2,T3,Td1,Td2,Td3
      OPEN (8,FILE='fpidr11.dat',STATUS='unknown')
      OPEN (12,FILE='fpidr11T.dat',STATUS='unknown')
      OPEN (9,FILE='fpidr111.dat',STATUS='unknown')
      OPEN (10,FILE='fpidr112.dat',STATUS='unknown')
      OPEN (11,FILE='fpidr113.dat',STATUS='unknown')
15    FORMAT(1X,F6.2,1X,F8.4,1X,F10.4,1X,F8.4,1X,F10.4,1X,F8.4,1X
     +,F10.4)
25    FORMAT(1X,F6.2,1X,F10.4,1X,F10.4,1X,F10.4)

      ALPHA=.46051702
      BETA=3.14

      I2=ALPHA*(ALPHA**2+BETA**2)
      I3=ALPHA*(ALPHA**2+BETA**2)
      L1=1
      L2=2
```

```
      L3=3
      NEO=9
      DT=.01D0
      NSTEP=1000
      TIME=0.0D0
      X(1)=0.0
      X(2)=-1.0
      X(3)=0.0
      X(4)=0.0
      X(5)=-1.0
      X(6)=0.0
      X(7)=0.0
      X(8)=1.0
      X(9)=0.0
      Td1=1.0
      Td2=1.0
      Td3=-1.0
      TRIG1=0
      TRIG2=0
      TRIG3=0
      CALL FUZZY_LOGIC(X,T1,TRIG1,L1)
      CALL FUZZY_LOGIC(X,T2,TRIG2 L2)
      CALL FUZZY_LOGIC(X,T3,TRIG3,L3)
      TRIG1=1
      TRIG2=1
      TRIG3=1
      C23=COS(X(5)+Td2+X(8)+Td3)
      C2=COS(X(5)+Td2)
      C3=COS(X(8)+Td3)
      M11=26.+8.*C23**2+29.*C2**2+24.*C2*C23
      M22=43.+24.*C3
      M23=8.+12.*C3
      M33=8.
c Agument the torque produced by the Fuzzy controller with that of c the traditional
Integral feedback
      T1=T1
      T2=T2-I2*(M22*X(4)+M23*X(7))
      T3=T3-I3*(M23*X(4)+M33*X(7))
c      T1=0.0
c      T2=0.0
c      T3=0.0
      WRITE(8,15)TIME,X(2)+Td1,T1,X(5)+Td2,T2,X(8)+Td3,T3
      WRITE(12,25)TIME,X(2)+Td1,X(5)+Td2,X(8)+Td3
      DO 100 COUNT=1,NSTEP
          CALL RUNGA(X,DT,NEQ,TIME)
          CALL FUZZY_LOGIC(X,T1,TRIG1,L1)
          CALL FUZZY_LOGIC(X,T2,TRIG2,L2)
          CALL FUZZY_LOGIC(X,T3,TRIG3,L3)
          C23=COS(X(5)+Td2+X(8)+Td3)
          C2=COS(X(5)+Td2)
          C3=COS(X(8)+Td3)
```

```
            M11=26.+8.*C23**2+29.*C2**2+24.*C2*C23
            M22=43.+24.*C3
            M23=8.+12.*C3
            M33=8.
c Agument the torque produced by the Fuzzy controller with that of c the traditional
Integral feedback
            T1=T1
            T2=T2-I2*(M22*X(4)+M23*X(7))
            T3=T3-I3*(M23*X(4)+M33*X(7))
c           T1=0.0
c           T2=0.0
c           T3=0.0
            WRITE(8,15)TIME,X(2)+Td1,T1,X(5)+Td2,T2,X(8)+Td3,T3
            WRITE(12,25)TIME,X(2)+Td1,X(5)+Td2,X(8)+Td3
 100  CONTINUE
         CLOSE(8)
         CLOSE(9)
         CLOSE(10)
         CLOSE(11)
         END

         SUBROUTINE RIGHT(R1,R2,R3,X,M11,M22,M33,M23,DET)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c        PURPOSE: To calculate the respective mass matrix entries and   c
c            the respective nonlinear contributions R#.                c
c                                                                      c
c        AUTHOR:  Robert J. Stanley II                                 c
c                                                                      c
c        DATE:  8/5/92                                                 c
c                                                                      c
c        VARIABLES:                                                    c
c            R#:  The nonlinear terms of link # respectively           c
c            F:  State space                                           c
c            G:  Gravity                                               c
c            DET:  The determinate of the mass matrix divided by M11   c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

         REAL*8 X(10),Td1,Td2,Td3
         REAL*8 T1,T2,T3,R1,R2,R3,M11,M22,M33,M23,G,C23,S23,C2,C3,S2
         REAL*8 S3,C23S23,C2S23,S2C23,S2C2,C2S2,DET
         COMMON /MYCOMM/ T1,T2,T3,Td1,Td2,Td3

         G=9.8
         C23=COS(X(5)+Td2+X(8)+Td3)
         S23=SIN(X(5)+Td2+X(8)+Td3)
         C2=COS(X(5)+Td2)
         C3=COS(X(8)+Td3)
         S2=SIN(X(5)+Td2)
         S3=SIN(X(8)+Td3)
         C23S23=C23*S23
         C2S23=C2*S23
```

```
      S2C23=S2*C23
      S2C2=S2*C2
      C2S2=C2*S2

      M11=26.+8.*C23**2+29.*C2**2+24.*C2*C23
      M22=43.+24.*C3
      M23=8.+12.*C3
      M33=8.
      DET=M22*M33-M23*M23

      R1=(16.*C23S23+24.*C2S23)*X(3)*(X(6)+X(9))
     ++(24.*S2C23+58.*S2C2)*X(3)*X(6)+T1

      R2=-(8.*C23S23+12.*S2C23+12.*C2S23+29.*C2S2)*X(3)**2
     ++24.*S3*X(6)*X(9)+12.*S3*X(9)**2-20.*G*C2-6.*G*C23+T2

      R3=-(8.*C23S23+12.*C2S23)*X(3)**2-12.*S3*X(6)**2-6.*G*C23+T3
      RETURN
      END


      SUBROUTINE STATE(F,X,TIME)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: Define the dynamics in a state space form for use in  c
c       Runge Kutta Subroutine.                                      c
c                                                                    c
c     AUTHOR: Robert J. Stanley II                                   c
c                                                                    c
c     DATE: 8/4/92                                                   c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      REAL*8 X(10),F(10),TIME,Td1,Td2,Td3,T1,T2,T3
      REAL*8 R1,R2,R3,M11,M22,M33,M23,DET
      COMMON /MYCOMM/ T1,T2,T3,Td1,Td2,Td3

      TIME=TIME*1.0

      CALL RIGHT(R1,R2,R3,X,M11,M22,M33,M23,DET)
      F(1)=X(2)
      F(2)=X(3)
      F(3)=R1/M11
      F(4)=X(5)
      F(5)=X(6)
      F(6)=(R2*M33/DET)-(R3*M23/DET)
      F(7)=X(8)
      F(8)=X(9)
      F(9)=-(R2*M23/DET)+(R3*M22/DET)

      RETURN
      END
```

```
      SUBROUTINE RUNGA(X,DT,NEQ,TIME)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: Use a Runge Kutta algorithm to numerical solve the    c
c          state equations given is subroutine STATE.                c
c                                                                    c
c     AUTHOR: Robert J. Stanley II                                   c
c                                                                    c
c     DATE: 8/4/92                                                   c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      REAL*8 X(10),Y(10),F(10),DT,TIME,G1(10),G2(10),G3(10),G4(10)
      INTEGER I,NEQ

      DO 1 I=1,NEQ
1        Y(I)=X(I)

      CALL STATE(F,Y,TIME)
      DO 2 I=1,NEQ
2        G1(I)=DT*F(I)

      TIME=TIME+DT/2.0D0
      DO 3 I=1,NEQ
3        Y(I)=X(I)+G1(I)/2.0D0

      CALL STATE(F,Y,TIME)
      DO 4 I=1,NEQ
         G2(I)=DT*F(I)
4        Y(I)=X(I)+G2(I)/2.0D0

      CALL STATE(F,Y,TIME)
      DO 5 I=1,NEQ
         G3(I)=DT*F(I)
5        Y(I)=X(I)+G3(I)

      TIME=TIME+DT/2.0D0
      CALL STATE(F,Y,TIME)
      DO 6 I=1,NEQ
6        G4(I)=DT*F(I)

      DO 7 I=1,NEQ
7        X(I)=X(I)+(G1(I)+2.0D0*(G2(I)+G3(I))+G4(I))/6.0D0
      RETURN
      END

      END

      SUBROUTINE FUZZY_LOGIC(X,TORQUE,TRIGGER,LINK)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c     PURPOSE: Given a position calculate a torque required to       c
```

```
c       drive the error to zero using Fuzzy Logic Control.              c
c                                                                       c
c       AUTHOR: Robert J. Stanley II                                    c
c                                                                       c
c       DATE: 8/4/92                                                    c
c                                                                       c
c       VARIABLES:                                                      c
c       E: Error                                                        c
c       CE: Change in Error                                             c
c       CEA: Change is Error Angle                                      c
c       LASTE: The last error                                           c
c       SET_PT: Set point desired                                       c
c       PI: 3.14                                                        c
c       QE: Quantizied value of the error                               c
c       QEC: Quantizied value of the Error Change                       c
c       u: Membership function value                                    c
c       UU: Universe of discourse value                                 c
c       NUM: NUMerator of the input value                               c
c       DEN: DENomenator of the input value                             c
c       Ye: Temp variable for the Error membership function             c
c       Yec: Temp variable for Change in Error membership func.         c
c       INPUT: The quantizied input to the plant                        c
c       TORQUE: The actual input to the plant                           c
c       N: Number of rules                                              c
c       I: Count variable                                               c
c       GRID: Tells output which grid is being utilized                 c
c       FINE: Boolean for the quantizied table                          c
c       MEDIUM: Boolean for the quantizied table                        c
c       COARSE: Boolean for the quantizied table                        c
c       GRID: Indicates which quantizied table is being accessed        c
c       ELP: Linguistic value Error Large Positive                      c
c       EMP: Linguistic value Error Medium Positive                     c
c       ESP: Linguistic value Error Small Positive                      c
c       EZE: Linguistic value Error Zero                                c
c       ESN: Linguistic value Error Small Negative                      c
c       EMN: Linguistic value Error Medium Negative                     c
c       ELN: Linguistic value Error Large Negative                      c
c       CELP: Linguistic value Change Error Large Positive              c
c       CEMP: Linguistic value Change Error Medium Positive             c
c       CESP: Linguistic value Change Error Small Positive              c
c       CEZE: Linguistic value Change Error Zero                        c
c       CESN: Linguistic value Change Error Small Negative              c
c       CEMN: Linguistic value Change Error Medium Negative             c
c       CELN: Linguistic value Change Error Large Negative              c
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      REAL*8 E,CE,LASTE3,PI,QE,QEC,u(50),UU(50),NUM,DEN
      REAL*8 Ye,Yec,INPUT,TORQUE,CEA,QECA,LASTE1,LASTE2
      REAL*8 EMAX,TOR_MAX,X(10)
      INTEGER N,I,TRIGGER,GRID,LINK
```

```
      LOGICAL FINE,MEDIUM,COARSE,ELP,EMP,ESP,EZE,ESN,EMN,ELN
      LOGICAL CELP,CEMP,CESP,CEZE,CESN,CEMN,CELN

16    FORMAT(1X,F10.4,1X,F10.4,1X,F10.4,1X,F10.4,1X,F10.4)

      ELP=.FALSE.
      EMP=.FALSE.
      ESP=.FALSE.
      EZE=.FALSE.
      ESN=.FALSE.
      EMN=.FALSE.
      ELN=.FALSE.
      CELP=.FALSE.
      CEMP=.FALSE.
      CESP=.FALSE.
      CEZE=.FALSE.
      CESN=.FALSE.
      CEMN=.FALSE.
      CELN=.FALSE.
      FINE=.FALSE.
      MEDIUM=.FALSE.
      COARSE=.FALSE.

      N=49
      PI=3.14
      IF (LINK.EQ.1) THEN
          E=X(2)
          IF (TRIGGER.EQ.0) THEN
              CEA=0.0
          ELSE
              CEA=ATAN2(E-LASTE1,.01)
          END IF
          LASTE1=E
          EMAX=60.0
          TOR_MAX=500.0
      END IF
      IF (LINK.EQ.2) THEN
          E=X(5)
          IF (TRIGGER.EQ.0) THEN
              CEA=0.0
          ELSE
              CEA=ATAN2(E-LASTE2,.01)
          END IF
          LASTE2=E
          EMAX=60.0
          TOR_MAX=500.0
      END IF
      IF (LINK.EQ.3) THEN
          E=X(8)
          IF (TRIGGER.EQ.0) THEN
              CEA=0.0
```

```
            ELSE
                CEA=ATAN2(E-LASTE3,.01)
            END IF
            LASTE3=E
            EMAX=60.0
            TOR_MAX=150.0
        END IF

c Change error and change in error from radians to degrees. E=(180/3.14)*E
        CEA=(180/3.14)*CEA
c Determine which quantizied table is to be used and find the c corresponding
quantizied values of error and error change.
c          IF ((E.LT.25.0).AND.(E.GT.-25.0)) THEN
c               FINE=.TRUE.
c               QE=E*(6/25.0)
c               QECA=CEA*(6/90.)
c          ELSE
c               IF ((E.LT.33.0).AND.(E.GT.-33.0)) THEN
c                    MEDIUM=.TRUE.
c                    QE=E*(6/33.)
c                    QECA=CEA*(6/90.)
c               ELSE
c                    COARSE=.TRUE.
c                    QE=E*(6/60.)
c                    QECA=CEA*(6/90.)
c               END IF
c          END IF
           COARSE=.TRUE.
           QE=E*(6/EMAX)
           QECA=CEA*(6/90.)

c Deterimine which grid is being used
           IF (COARSE) THEN
                GRID=1
           END IF
           IF (MEDIUM) THEN
                GRID=2
           END IF
           IF (FINE) THEN
                GRID=3
           ENDIF

c With the Quantizied Error determine which c
linguistic values are applicable.
           IF (QE.GE.6.0) THEN
                QE=6.0
                ELP=.TRUE.
           END IF
           IF ((QE.GE.4.0).AND.(QE.LT.6.0)) THEN ELP=.TRUE.
                EMP=.TRUE.
           END IF
```

```
IF ((QE.GE.2.0).AND.(QE.LE.4.0)) THEN EMP=.TRUE.
   ESP=.TRUE.
END IF
IF ((QE.GE.0.0).AND.(QE.LE.2.0)) THEN ESP=.TRUE.
   EZE=.TRUE.
END IF
  IF ((QE.GE.-2.0).AND.(QE.LE.0.0)) THEN
    EZE=.TRUE.
    ESN=.TRUE.
END IF
  IF ((QE.GE.-4.0).AND.(QE.LE.-2.0)) THEN
    ESN=.TRUE.
    EMN=.TRUE.
END IF
  IF ((QE.GE.-6.0).AND.(QE.LE.-4.0)) THEN
    EMN=.TRUE.
    ELN=.TRUE.
END IF
IF (QE.LE.-6.0) THEN
    QE=-6.0
    ELN=.TRUE.
END IF

c With the Quantizied Error Change determine which
c linguistic values are applicable.
      IF (QECA.GE.6.0) THEN
          QECA=6.0
          CELP=.TRUE.
      END IF
      IF ((QECA.GE.4.0).AND.(QECA.LT.6.0)) THEN
          CELP=.TRUE.
          CEMP=.TRUE.
      END IF
      IF ((QECA.GE.2.0).AND.(QECA.LE.4.0)) THEN
          CEMP=.TRUE.
          CESP=.TRUE.
      END IF
      IF ((QECA.GE.0.0).AND.(QECA.LE.2.0)) THEN
          CESP=.TRUE.
          CEZE=.TRUE.
      END IF
      IF ((QECA.GE.-2.0).AND.(QECA.LE.0.0)) THEN
          CEZE=.TRUE.
          CESN=.TRUE.
      END IF
      IF ((QECA.GE.-4.0).AND.(QECA.LE.-2.0)) THEN
          CESN=.TRUE.
          CEMN=.TRUE.
      END IF
      IF ((QECA.GE.-6.0).AND.(QECA.LE.-4.0)) THEN
          CEMN=.TRUE.
```

```
            CELN=.TRUE.
        END IF
        IF (QECA.LE.-6.0) THEN
            QECA=-6.0
            CELN=.TRUE.
        END IF

c Initialize the membership function value (u) and the
c universe of discourse value (U) to zero before the rules c are applied.
        DO 250 I=1,N
            u(I)=0.0
            UU(I)=0.0
250  CONTINUE

c Rule one if Error is Large Positive and the Change in Error is c Large Negative
then contribution is Zero.
        IF (ELP.AND.CELN) THEN
            Ye=SIN(PI/4*(QE-4.0))
            Yec=SIN(PI/4*(QECA+8.0))
            u(1)=MIN(Ye,Yec)
            UU(1)=0.0
        END IF
c Rule two if Error is Large Positive and the Change in Error is c Medium
Negative then contribution is Small Negative.
        IF (ELP.AND.CEMN) THEN
            Ye=SIN(PI/4*(QE-4.0))
            Yec=SIN(PI/4*(QECA+6.0))
            u(2)=MIN(Ye,Yec)
            UU(2)=-2.0
        END IF
c Rule three if Error is Large Positive and the Change in Error is c Small Negative
then contribution is Medium Negative.
        IF (ELP.AND.CESN) THEN
            Ye=SIN(PI/4*(QE-4.0))
            Yec=SIN(PI/4*(QECA+4.0))
            u(3)=MIN(Ye,Yec)
            UU(3)=-4.0
        END IF
c Rule four if Error is Large Positive and the Change in Error is c Zero then
contribution is Large Negative.
        IF (ELP.AND.CEZE) THEN
            Ye=SIN(PI/4*(QE-4.0))
            Yec=SIN(PI/4*(QECA+2.0))
            u(4)=MIN(Ye,Yec)
            UU(4)=-6.0
        END IF
c Rule five if Error is Large Positive and the Change in Error is c Small Positive
then contribution is Large Negative.
        IF (ELP.AND.CESP) THEN
            Ye=SIN(PI/4*(QE-4.0))
```

```
            Yec=SIN(PI/4*(QECA))
            u(5)=MIN(Ye,Yec)
            UU(5)=-6.0
        END IF
c Rule six if Error is Large Positive and the Change in Error is c Medium Positive
then contribution is Large Negative.
        IF (ELP.AND.CEMP) THEN
            Ye=SIN(PI/4*(QE-4.0))
            Yec=SIN(PI/4*(QECA-2.0))
            u(6)=MIN(Ye,Yec)
            UU(6)=-6.0
        END IF
c Rule seven if Error is Large Positive and the Change in Error is c Large Positive
then contribution is Large Negative.
        IF (ELP.AND.CELP) THEN
            Ye=SIN(PI/4*(QE-4.0))
            Yec=SIN(PI/4*(QECA-4.0))
            u(7)=MIN(Ye,Yec)
            UU(7)=-6.0
        END IF
c Rule eight if Error is Medium Positive and the Change in Error is c Large Negative
then contribution is Small Positive.
        IF (EMP.AND.CELN) THEN
            Ye=SIN(PI/4*(QE-2.0))
            Yec=SIN(PI/4*(QECA+8.0))
            u(8)=MIN(Ye,Yec)
            UU(8)=2.0
        END IF
c Rule nine if Error is Medium Positive and the Change in Error is c Medium
Negative then contribution is Zero.
        IF (EMP.AND.CEMN) THEN
            Ye=SIN(PI/4*(QE-2.0))
            Yec=SIN(PI/4*(QECA+6.0))
            u(9)=MIN(Ye,Yec)
            UU(9)=0.0
        END IF
c Rule ten if Error is Medium Positive and the Change in Error is c Small Negative
then contribution is Small Negative.
        IF (EMP.AND.CESN) THEN
            Ye=SIN(PI/4*(QE-2.0))
            Yec=SIN(PI/4*(QECA+4.0))
            u(10)=MIN(Ye,Yec)
            UU(10)=-2.0
        END IF
c Rule eleven if Error is Medium Positive and the Change in Error is c Zero then
contribution is Medium Negative.
        IF (EMP.AND.CEZE) THEN
            Ye=SIN(PI/4*(QE-2.0))
            Yec=SIN(PI/4*(QECA+2.0))
            u(11)=MIN(Ye,Yec)
            UU(11)=-4.0
```

```
      END IF
c Rule twelve if Error is Medium Positive and the Change in Error is c Small Positive
then contribution is Large Negative.
      IF (EMP.AND.CESP) THEN
         Ye=SIN(PI/4*(QE-2.0))
         Yec=SIN(PI/4*(QECA))
         u(12)=MIN(Ye,Yec)
         UU(12)=-6.0
      END IF
c Rule thirteen if Error is Medium Positive and the Change in Error is c Medium Positive
then contribution is Large Negative.
      IF (EMP.AND.CEMP) THEN
         Ye=SIN(PI/4*(QE-2.0))
         Yec=SIN(PI/4*(QECA-2.0))
         u(13)=MIN(Ye,Yec)
         UU(13)=-6.0
      END IF
c Rule fourteen if Error is Medium Positive and the Change in Error is c Large Positive
then contribution is Large Negative.
      IF (EMP.AND.CELP) THEN
         Ye=SIN(PI/4*(QE-2.0))
         Yec=SIN(PI/4*(QECA-4.0))
         u(14)=MIN(Ye,Yec)
         UU(14)=-6.0
      END IF
c Rule fifteen if Error is Small Positive and the Change in Error is c Large Negative
then contribution is Medium Positive.
      IF (ESP.AND.CELN) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA+8.0))
         u(15)=MIN(Ye,Yec)
         UU(15)=4.0
      END IF
c Rule sixteen if Error is Small Positive and the Change in Error is c Medium Negative
then contribution is Small Positive.
      IF (ESP.AND.CEMN) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA+6.0))
         u(16)=MIN(Ye,Yec)
         UU(16)=2.0
      END IF
c Rule seventeen if Error is Small Positive and the Change in Error is c Small Negative
then contribution is Zero.
      IF (ESP.AND.CESN) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA+4.0))
         u(17)=MIN(Ye,Yec)
         UU(17)=-2.0
      END IF
c Rule eighteen if Error is Small Positive and the Change in Error is c Zero then
contribution is Small Negative.
```

```
      IF (ESP.AND.CEZE) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA+2.0))
         u(18)=MIN(Ye,Yec)
         UU(18)=-2.0
      END IF
c Rule nineteen if Error is Small Positive and the Change in Error is c Small Positive then
contribution is Medium Negative.
      IF (ESP.AND.CESP) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA))
         u(19)=MIN(Ye,Yec)
         UU(19)=-4.0
      END IF
c Rule twenty if Error is Small Positive and the Change in Error is c Medium Positive
then contribution is Large Negative.
      IF (ESP.AND.CEMP) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA-2.0))
         u(20)=MIN(Ye,Yec)
         UU(20)=-6.0
      END IF
c Rule twenty one if Error is Small Positive and the Change in Error is c Large Positive then
contribution is Large Negative.
      IF (ESP.AND.CELP) THEN
         Ye=SIN(PI/4*(QE))
         Yec=SIN(PI/4*(QECA-4.0))
         u(21)=MIN(Ye,Yec)
         UU(21)=-6.0
      END IF
c Rule twenty two if Error is Zero and the Change in Error is c Large
Negative then contribution is Large Positive.
      IF (EZE.AND.CELN) THEN
         Ye=SIN(PI/4*(QE+2.0))
         Yec=SIN(PI/4*(QECA+8.0))
         u(22)=MIN(Ye,Yec)
         UU(22)=6.0
      END IF
c Rule twenty three if Error is Zero and the Change in Error is c Medium
Negative then contribution is Medium Positive.
      IF (EZE.AND.CEMN) THEN
         Ye=SIN(PI/4*(QE+2.0))
         Yec=SIN(PI/4*(QECA+6.0))
         u(23)=MIN(Ye,Yec)
         UU(23)=4.0
      END IF
c Rule twenty four if Error is Zero and the Change in Error is c Small Negative
then contribution is Small Positive.
      IF (EZE.AND.CESN) THEN
         Ye=SIN(PI/4*(QE+2.0))
         Yec=SIN(PI/4*(QECA+4.0))
```

```
        u(24)=MIN(Ye,Yec)
        UU(24)=2.0
    END IF
c Rule twenty five if Error is Zero and the Change in Error is c Zero then
contribution is Zero.
    IF (EZE.AND.CEZE) THEN
        Ye=SIN(PI/4*(QE+2.0))
        Yec=SIN(PI/4*(QECA+2.0))
        u(25)=MIN(Ye,Yec)
        UU(25)=0.0
    END IF
c Rule Twenty six if Error is Zero and the Change in Error is c Small Positive
then contribution is Small Negative.
    IF (EZE.AND.CESP) THEN
        Ye=SIN(PI/4*(QE+2.0))
        Yec=SIN(PI/4*(QECA))
        u(26)=MIN(Ye,Yec)
        UU(26)=-2.0
    END IF
c Rule Twenty seven if Error is Zero and the Change in Error is c Medium
Positive then contribution is Medium Negative.
    IF (EZE.AND.CEMP) THEN
        Ye=SIN(PI/4*(QE+2.0))
        Yec=SIN(PI/4*(QECA-2.0))
        u(27)=MIN(Ye,Yec)
        UU(27)=-4.0
    END IF
c Rule Twenty eight if Error is Zero and the Change in Error is c Large Positive
then contribution is Large Negative.
    IF (EZE.AND.CELP) THEN
        Ye=SIN(PI/4*(QE+2.0))
        Yec=SIN(PI/4*(QECA-4.0))
        u(28)=MIN(Ye,Yec)
        UU(28)=-6.0
    END IF
c Rule Twenty nine if Error is Small Negative and the Change in c Error is
Large Negative then contribution is Large Positive.
    IF (ESN.AND.CELN) THEN
        Ye=SIN(PI/4*(QE+4.0))
        Yec=SIN(PI/4*(QECA+8.0))
        u(29)=MIN(Ye,Yec)
        UU(29)=6.0
    END IF
c Rule Thirty if Error is Small Negative and the Change in
c Error is Medium Negative then contribution is Large Positive. IF
    (ESN.AND.CEMN) THEN
        Ye=SIN(PI/4*(QE+4.0))
        Yec=SIN(PI/4*(QECA+6.0))
        u(30)=MIN(Ye,Yec)
        UU(30)=6.0
    END IF
```

```
c Rule Thirty one if Error is Small Negative and the Change in c Error is Small
Negative then contribution is Medium Positive.
        IF (ESN.AND.CESN) THEN
            Ye=SIN(PI/4*(QE+4.0))
            Yec=SIN(PI/4*(QECA+4.0))
            u(31)=MIN(Ye,Yec)
            UU(31)=4.0
        END IF
c Rule Thirty two if Error is Small Negative and the Change in c Error is Zero
then contribution is Small Positive.
        IF (ESN.AND.CEZE) THEN
            Ye=SIN(PI/4*(QE+4.0))
            Yec=SIN(PI/4*(QECA+2.0))
            u(32)=MIN(Ye,Yec)
            UU(32)=2.0
        END IF
c Rule Thirty three if Error is Small Negative and the Change in c Error is Small
Positive then contribution is Small Positive.
        IF (ESN.AND.CESP) THEN
            Ye=SIN(PI/4*(QE+4.0))
            Yec=SIN(PI/4*(QECA))
            u(33)=MIN(Ye,Yec)
            UU(33)=2.0
        END IF
c Rule Thirty four if Error is Small Negative and the Change in c Error is
Medium Positive then contribution is Small Negative.
        IF (ESN.AND.CEMP) THEN
            Ye=SIN(PI/4*(QE+4.0))
            Yec=SIN(PI/4*(QECA-2.0))
            u(34)=MIN(Ye,Yec)
            UU(34)=-2.0
        END IF
c Rule Thirty five if Error is Small Negative and the Change in c Error is Large
Positive then contribution is Medium Negative.
        IF (ESN.AND.CELP) THEN
            Ye=SIN(PI/4*(QE+4.0))
            Yec=SIN(PI/4*(QECA-4.0))
            u(35)=MIN(Ye,Yec)
            UU(35)=-4.0
        END IF
c Rule Thirty six if Error is Medium Negative and the Change in c Error is
Large Negative then contribution is Large Positive.
        IF (EMN.AND.CELN) THEN
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA+8.0))
            u(36)=MIN(Ye,Yec)
            UU(36)=6.0
        END IF
c Rule Thirty seven if Error is Medium Negative and the Change in c Error is
Medium Negative then contribution is Large Positive.
        IF (EMN.AND.CEMN) THEN
```

```
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA+6.0))
            u(37)=MIN(Ye,Yec)
            UU(37)=6.0
         END IF
c Rule Thirty eight if Error is Medium Negative and the Change in c Error is Small
Negative then contribution is Large Positive.
         IF (EMN.AND.CESN) THEN
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA+4.0))
            u(38)=MIN(Ye,Yec)
            UU(38)=6.0
         END IF
c Rule Thirty nine if Error is Medium Negative and the Change in c Error is Zero
then contribution is Medium Positive.
         IF (EMN.AND.CEZE) THEN
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA+2.0))
            u(39)=MIN(Ye,Yec)
            UU(39)=4.0
         END IF
c Rule fourty if Error is Medium Negative and the Change in c Error is Small
Positive then contribution is Small Positive.
         IF (EMN.AND.CESP) THEN
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA))
            u(40)=MIN(Ye,Yec)
            UU(40)=2.0
         END IF
c Rule fourty one if Error is Medium Negative and the Change in c Error is
Medium Positive then contribution is Zero.
         IF (EMN.AND.CEMP) THEN
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA-2.0))
            u(41)=MIN(Ye,Yec)
            UU(41)=0.0
         END IF
c Rule fourty two if Error is Medium Negative and the Change in c Error is
Large Positive then contribution is Small Negative.
         IF (EMN.AND.CELP) THEN
            Ye=SIN(PI/4*(QE+6.0))
            Yec=SIN(PI/4*(QECA-4.0))
            u(42)=MIN(Ye,Yec)
            UU(42)=-2.0
         END IF
c Rule fourty three if Error is Large Negative and the Change in c Error is Large
Negative then contribution is Large Positive.
         IF (ELN.AND.CELN) THEN
            Ye=SIN(PI/4*(QE+8.0))
            Yec=SIN(PI/4*(QECA+8.0))
            u(43)=MIN(Ye,Yec)
```

```
          UU(43)=6.0
      END IF
c Rule fourty four if Error is Large Negative and the Change in c Error is
Medium Negative then contribution is Large Positive.
      IF (ELN.AND.CEMN) THEN
          Ye=SIN(PI/4*(QE+8.0))
          Yec=SIN(PI/4*(QECA+6.0))
          u(44)=MIN(Ye,Yec)
          UU(44)=6.0
      END IF
c Rule fourty five if Error is Large Negative and the Change in c Error is Small
Negative then contribution is Large Positive.
      IF (ELN.AND.CESN) THEN
          Ye=SIN(PI/4*(QE+8.0))
          Yec=SIN(PI/4*(QECA+4))
          u(45)=MIN(Ye,Yec)
          UU(45)=6.0
      END IF
c Rule fourty six if Error is Large Negative and the Change in c Error is Zero
then contribution is Large Positive.
      IF (ELN.AND.CEZE) THEN
          Ye=SIN(PI/4*(QE+8.0))
          Yec=SIN(PI/4*(QECA+2))
          u(46)=MIN(Ye,Yec)
          UU(46)=6.0
      END IF
c Rule fourty seven if Error is Large Negative and the Change in c Error is Small
Positive then contribution is Medium Positive.
      IF (ELN.AND.CESP) THEN
          Ye=SIN(PI/4*(QE+8.0))
          Yec=SIN(PI/4*(QECA))
          u(47)=MIN(Ye,Yec)
          UU(47)=4.0
      END IF
c Rule fourty eight if Error is Large Negative and the Change in c Error is
Medium Positive then contribution is Small Positive.
      IF (ELN.AND.CEMP) THEN
          Ye=SIN(PI/4*(QE+8.0))
          Yec=SIN(PI/4*(QECA-2.0))
          u(48)=MIN(Ye,Yec)
          UU(48)=2.0
      END IF
c Rule fourty nine if Error is Large Negative and the Change in c Error is Large
Positive then contribution is Zero.
      IF (ELN.AND.CELP) THEN
          Ye=SIN(PI/4*(QE+8.0))
          Yec=SIN(PI/4*(QECA-4.0))
          u(49)=MIN(Ye,Yec)
          UU(49)=0.0
      END IF
```

```
c Initialize the NUMerator and DENomenator to zero so that only c contributions
occurring on this pass will be considered.
      NUM=0.0
      DEN=0.0
c Calculate the NUMerator and the DENomenator of the control input c by means of
the center of gravity method.
      DO 300 I=1,N
         NUM=NUM+u(I)*UU(I)
         DEN=DEN+u(I)
300   CONTINUE
c Setting the DENomentator to 1.0 prevents division by zero and c does not
effect the value of the control input.
      IF (DEN.LT.0001)THEN
         DEN=1.0
      END IF
      INPUT=NUM/DEN
      IF (LINK.EQ.1) THEN
         WRITE(9,16)E,CEA,QE,QECA,
      INPUT END IF
      IF (LINK.EQ.2) THEN
         WRITE(10,16)E,CEA,QE,QEC
         A,INPUT
      END IF
      IF (LINK.EQ.3) THEN
         WRITE(11,16)E,CEA,QE,QEC
         A,INPUT
      END IF

c Using the correct quantizied table convert the input into c a torque to be
sent to the plant.
      IF (COARSE) THEN
         TORQUE=INPUT*(TOR_MAX/6.0)
      END IF
      IF (MEDIUM) THEN
         TORQUE=INPUT*(15./6.0)
      END IF
      IF (FINE) THEN
         TORQUE=INPUT*(17./6.0)
      END IF
      RETURN
      END
```