Summer 2000

Design of Adaptive Sliding Mode Fuzzy Control for Robot Manipulator Based on Extended Kalman Filter

Abdelrahman Aledhaibi
Old Dominion University

DESIGN OF ADAPTIVE SLIDING MODE FUZZY CONTROL FOR ROBOT MANIPULATOR BASED ON EXTENDED KALMAN FILTER

by

Abdelrahman Aledhaibi
M.S. December 1995, Murray State University
B.S. May 1992, Southern Illinois University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPY

MECHANICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 2000

Approved by:

Jen-Kuang Huang (Director)

Sabastian Bawab (Member)

Keith M. Williamson (Member)

Cheng Y. Lin (Member)

ABSTRACT

DESIGN OF ADAPTIVE SLIDING MODE FUZZY CONTROL
FOR ROBOT MANIPULATOR BASED ON EXTENDED KALMAN FILTER

Abdelrahman M. Aledhaibi
Old Dominion University, 2000
Director: Dr. Jen-Kuang Huang

In this work, a new adaptive motion control scheme for robust performance control of robot manipulators is presented. The proposed scheme is designed by combining the fuzzy logic control with the sliding mode control based on extended Kalman filter. Fuzzy logic controllers have been used successfully in many applications and were shown to be superior to the classical controllers for some nonlinear systems. Sliding mode control is a powerful approach for controlling nonlinear and uncertain systems. It is a robust control method and can be applied in the presence of model uncertainties and parameter disturbances, provided that the bounds of these uncertainties and disturbances are known. We have designed a new adaptive Sliding Mode Fuzzy Control (SMFC) method that requires only position measurements. These measurements and the input torques are used in an extended Kalman filter (EKF) to estimate the inertial parameters of the full nonlinear robot model as well as the joint positions and velocities. These estimates are used by the SMFC to generate the input torques. The combination of the EKF and the SMFC is shown to result in a stable adaptive control scheme called trajectory-tracking adaptive robot with extended Kalman (TAREK) method. The theory behind TAREK method provides clear guidelines on the selection of the design parameters for the controller. The proposed controller is applied to a two-link robot

manipulator. Computer simulations show the robust performance of the proposed scheme.

In the memory of my father

This dissertation is dedicated to
my mother Monirah, my wife Norah,
my daughter Shymaa, and my son Tarek.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jen-Kuang Huang, for his thoughtful guidance, care and motivation. The members of my doctoral committee, Dr. Sebastian Bawab, Dr. Keith Williamson and Dr. Cheng Lin are especially appreciated for all the help they gave me.

This dissertation would not have been possible without my mother's and my wife's love, understanding and support. I would also like to thank my daughter, Shymaa, and my son, Tarek, for their patience and encouragement. A special thanks to my brothers and sisters for their sincere care and advice.

In addition, I would like to thank Hussein Alahmadi, Ahmed Alfihaed and all members of Saudi Student Association in Norfolk for their encouragement and support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

Table

LIST OF FIGURES

Page

CHAPTER I

Introduction

The use of robotics manipulators has had an immeasurable effect on industry and manufacturing world-wide. Robots can perform very efficiently and economically the simple repetitive tasks that human workers hate to do. Other tasks where the use of robots is especially beneficial include those that are conducted in hazardous environments. These typically include painting, welding and dangerous material handling or removal. With the recent advance vision and force-torque sensor integration, manipulators have been able to accomplish relatively difficult assembly and inspection tasks. These improvements make robotic systems more flexible and enable them to carry out a wide variety of tasks. To take advantage of this versatility, the robotic controller must be able to operate with precision at high speeds while not being affected by changing loads and disturbance. It is very difficult to obtain all these qualities without introducing burdensome computations to the control algorithm. Classes of controllers that offer an excellent compromise use the theories of fuzzy logic and sliding mode. Controllers that use the theory of fuzzy logic and sliding mode are commonly called Sliding Mode Fuzzy Controllers (SMFC).

Fuzzy logic has been around since 1965 when L.A Zadeh[52] [1] laid the foundation of the so-called linguistic model. Fuzzy sets theory was proposed by L.A. Zadeh to provide a tool to help solve ill-defined problems. Fuzzy sets theory provides a systematic framework for dealing with different types of uncertainty within a single conceptual framework. In papers published in 1973 and 1974, L.A. Zadeh outlined the basic ideas underlying fuzzy control. Among those outlined are the concept of linguistic variables, fuzzy IF-THEN rules, fuzzy algorithms, the compositional rule of inference and the execution of fuzzy instructions. One of the most active areas of fuzzy sets theory is the field of the Fuzzy Logic Control (FLC). Fuzzy logic control has many advantages which makes it a very attractive area. First, it is suitable for both linear and nonlinear systems. Second, it allows for imprecise mathematical models and measuring sensors. Third, it is more robust than classical controllers. Fourth, it can combine both linguistic and crisp information in the same framework.

Sliding Mode Control (SMC) is a special type of control technique that is capable of making a control system very robust with respect to system parameter variations and external disturbances. In addition, the technique provides an easy way to design the control law for a plant, linear or nonlinear. It was pioneered in the Soviet Union in the early 1950's by S. V. Emelyanov and his cohorts[33]. The technique did not receive wide attention in the Western world until recently. In the 1970's, researchers discovered additional attractive properties of sliding mode control and have developed methods for control law design. The feasibility of the technique has not only been predicted by theory, but has also been demonstrated by numerous computer simulations and hardware

[1] Journal mode used for this dissertation is AIAA

experiments. Therefore, sliding mode control technique has become mature and ready to be applied. However, the major disadvantage of this technique is the chattering problem that may cause fatigue, mechanical failure and loss of energy.

In their previous works, a new method called sliding mode fuzzy control (SMFC) which combines SMC with FLC was introduced[103]. The goal is to reduce the chattering of SMC. In SMC the phase plane is divided by the switching manifold and the coordinate axes into mutually exclusive regions. These regions constitute a group of crisp sets for each of which a control law is defined. Chattering occurs when feedback gains are chosen improperly. On the other hand, SMFC treats these regions as fuzzy sets. The systems state may partially belong to one region or another so that several control laws may 'fire' at the same time. The final control signal will result from the compromise of these fired control laws. Using this strategy, the chattering of SMC can be reduced, while its robustness is retained.

1.1 Objective

Most of the fuzzy robot controllers with nonlinear multi-input multi-output (MIMO) systems are designed with a two-dimensional phase plane in mind. In this dissertation, the performance and the robustness of this kind of robot controller, which arises from their property of driving the system into the so-called sliding mode (SM), is shown. This method will make the controlled system is invariant to parameter fluctuations and disturbances. In addition, near the switching line, the continuous distribution of the control values in the phase plane causes a behavior similar to that of a sliding mode controller (SMC) with a boundary layer (BL). Even in the presence of high

model uncertainties, this gives assured tracking quality. Then, the boundary layer at the fuzzy controller is introduced to obtained further improvement. Furthermore, the stability of the closed-loop system can be obtained when using the principle of the SMC for the fuzzy controller.

In addition, a new adaptive motion control scheme for robust performance control of robot manipulators is presented. The proposed scheme is designed by SMFC based on extended Kalman filter. The new design adaptive SMFC method requires only position measurements. These measurements and the input torques are used in an extended Kalman filter (EKF) to estimate the inertial parameters of the full nonlinear robot model as well as the joint positions and velocities. These estimates are used by the SMFC to generate the input torques. The combination of the EKF and the SMFC is shown to result in a stable adaptive control scheme and is called trajectory-tracking adaptive robot with extended Kalman (TAREK) method. The theory behind TAREK method provides clear guidelines on the selection of the design parameters for the controller. The proposed controller is applied to a two-link robot manipulator. Computer simulations show the robust performance of both proposed schemes.

1.2 Dissertation Outline

Chapter II discusses the dynamics of robotic manipulators. The dynamic model of a two-link manipulator is formulated using rigid-body dynamic theory.

Chapter III briefly introduces the proportional Integral Derivative (PID) control. Then, the Computed Torque Control method is used to design the PID controller.

Chapter IV explains the details of sliding mode control. The theory of sliding mode control is first illustrated with a simple example. Then, it is expanded and used to design a controller. The steps in the design of the sliding mode controller are clearly explained along with the design assumptions. In addition, a summary of the published literature on previous research in the area of sliding mode control is introduced. Finally, we give a comprehensive derivation of an effective sliding mode control algorithm used to control a robotic manipulator.

In chapter V, a Fuzzy logic controller is considered in detail. Basic definitions and basic fuzzy mathematics required in implementing FLC are presented. The structure of FLC and its design parameters are also considered in this chapter.

In chapter VI, fuzzy robot controllers with nonlinear multi-input multi-output (MIMO) systems are designed with a two-dimensional phase plane in mind. In this chapter, the performance and the robustness of this kind of robot controller as it arises from its property of driving the system into the so-called sliding mode (SM) is shown. Therefore, this chapter clarifies the details of sliding mode fuzzy control (SMFC). In section 6.1, a background on sliding mode fuzzy control is discussed. In section 6.2, we give a short review of the ordinary SMC with BL. In section 6.3, we describe the diagonal form FLC and derive the similarities between the control law of a diagonal form SMFC and the control law of an SMC with BL, which will describe the design of the control law of an SMFC for the tracking control problem. In Section 6.4, we show that we can derive a higher order FSMC from a higher order SMC with BL. Finally, we apply the design method of SMFC to MIMO robotics system.

Chapter VII gives an overview of the simulation and discussion of results and shows the selection of the values for all the controllers parameters. It also shows the stability performance of the Sliding Mode Fuzzy Controller.

Chapter VIII introduces linear and nonlinear system identification. Then, a derivation of nonlinear type system identification called Extended Kalman Filter (EKF) is presented.

Chapter IX presents a new adaptive motion control scheme for robust performance control of robot manipulators. The proposed scheme is designed by SMFC based on extended Kalman filter. The new design adaptive SMFC method requires only position measurements. A combination of the EKF and the SMFC is shown to result in a stable adaptive control scheme called trajectory-tracking adaptive robot with extended Kalman (TAREK) method.

Chapter X discusses the conclusion that can be drawn from the result of all methods used in this dissertation along with recommendations for future work in the area of Sliding Mode Fuzzy Control.

CHAPTER II

Dynamics of a Two-Link Manipulator

The dynamics of the robot arm deals with the mathematical formulations of the robot arm equations of motion. The dynamic equations of motion of a manipulator are a set of mathematical equations describing the dynamic behavior of the manipulator. These kind of equations of motion are useful for computer simulation of the robot arm motion, the design of suitable control equations for a robot arm and the structure of a robot arm. This chapter describes the dynamics of manipulators for the control purpose. Maintaining the dynamic response of a computer-based manipulator is the purpose of manipulator control in accordance with some prespecified system performance and desired goals. In general, the dynamic performance of a manipulator directly depends on the efficiency of the control algorithms and the dynamic model of the manipulator. The control problem consists of obtaining dynamic models of the physical robot arm system and then specifying corresponding control laws or strategies to achieve the desired system response and performance.

By using known physical laws such as the laws of Newtonian mechanics and Lagrangian mechanics, the actual dynamic model of a robot arm can be obtained. This leads to the development of the dynamic equations of motion for the various articulated joints of the manipulator in terms of specified geometric and inertial parameters of the links. Conventional approaches like the Lagrange-Euler (L-E) and Newton-Euler (N-E) formulations could then be applied systematically to develop the actual robot arm motion equations.

Various forms of robot arm motion equations describing the rigid-body robot arm dynamics are obtained from these two formulations, such as Uicker's Lagrange-Euler equations[1-2], Hollerbach's Recursive-Lagrange (R-L) equations[3], Luh's Newton-Euler equations[4] and Lee's generalized d'Alembert (G-D) equations[5]. These motion equations are "equivalent" to each other in the sense that they describe the dynamic behavior of the same physical robot manipulator. However, the structure of these equations may differ as they are obtained for various reasons and purposes.

2.1 Lagrange Formulation

A manipulator's equations of motion are essentially a description of the relationship between the input joint torques and the motion of the arm linkage. This relationship is needed in a robotic simulation because the control system only specifies the torques that should be applied to the joints of the manipulator. The exact motion that results from the application of the torques needs to be found to insure an accurate simulation. Unfortunately, the precise model of a manipulator can never be found. However, as long as the major dynamic effects are included, a reasonable estimation is sufficient.

A common approach to finding the equations of motion of a dynamic system is the Lagrangian formulation[6-7]. The Lagrangian L is defined as the difference between the kinetic energy K and the potential energy P of the system

$$L = K - P \tag{2.1}$$

and the equations of motion of the system are given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right)-\frac{\partial L}{\partial q_i}=\tau_i \qquad i=1,2,\cdots,n \tag{2.2}$$

where

q_i = generalized coordinate o f the robot arm
$\dot{q}_i$ = first tme derivative o f the generalized coordinate, q_i
τ_i = generalized force (or torque) applied to the system at joint i to drive link i.

This method will be applied to the simple model of a two-link manipulator shown in Figure 2.1. The mass of each link, m_i, is lumped at the end of a massless rod of length l_i. The links are connected to each other and to the base via a revolute joint. The generalized coordinates, q_i, describe the position of each of the revolute joints, and the generalized torques, τ_i, are applied at the joints by servomotors which are not modeled.

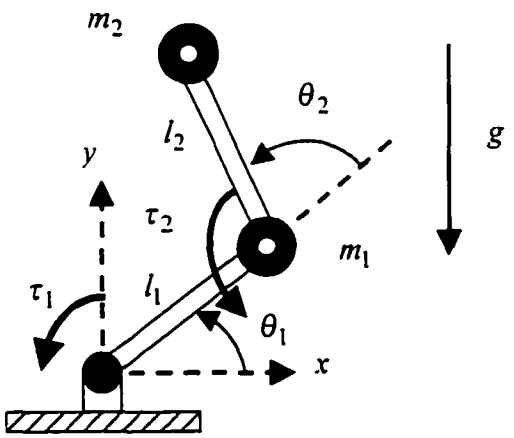


Figure 2.1 Schematic representation of a two-link manipulator

The joint variable for the two-link manipulator is

$$q = [\theta_1 \quad \theta_2] \tag{2.3}$$

and the generalized force vector is

$$\tau = [\tau_1 \quad \tau_2] \tag{2.4}$$

with τ_1 and τ_2 torques supplied by the actuators.

2.1.1 The Kinetic and Potential Energy

For link 1 the kinetic and potential energies are

$$K_1 = \tfrac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 \tag{2.5}$$

$$P_1 = m_1 g l_1 \sin(\theta_1). \tag{2.6}$$

For link 2,

$$x_2 = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{2.7}$$

$$y_2 = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{2.8}$$

$$\dot{x}_2 = -l_1 (\dot{\theta}_1) \sin(\theta_1) - l_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) \tag{2.9}$$

$$\dot{y}_2 = -l_1 (\dot{\theta}_1) \cos(\theta_1) - l_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2), \tag{2.10}$$

so that the velocity squared is

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos(\theta_2). \tag{2.11}$$

Therefore, the kinetic energy for link 2 is

$$K_2 = \tfrac{1}{2} m_2 v_2^2 = \tfrac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 + \tfrac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos(\theta_2). \tag{2.12}$$

The potential energy for link 2 is

$$P_2 = m_2 g y_2 = m_2 g [l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)]. \tag{2.13}$$

2.1.2 Lagrange's Equations

The Lagrangian for the entire arm is

$$
\begin{aligned}
L &= K - P = K_1 + K_2 - P_1 - P_2 \\
&= \tfrac{1}{2}(m_1 + m_2) l_1^2 \dot{\theta}_1^2 + \tfrac{1}{2} m_2 l_2^2 \left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 + m_2 l_1 l_2 \left(\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2\right) \cos(\theta_2) \\
&\quad -(m_1 + m_2) g l_1 \sin(\theta_1) - m_2 g l_2 \sin(\theta_1 + \theta_2).
\end{aligned} \tag{2.14}
$$

From equation (2.2), the torque at joint one is equal to

$$
\tau_1 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} \tag{2.15}
$$

where

$$
\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) l_1^2 \dot{\theta}_1 + m_2 l_2^2 \left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 + m_2 l_1 l_2 \left(2\dot{\theta}_1 + \dot{\theta}_2\right) \cos(\theta_2) \tag{2.16}
$$

$$
\begin{aligned}
\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} &= (m_1 + m_2) l_1^2 \ddot{\theta}_1 + m_2 l_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right)^2 + m_2 l_1 l_2 \left(2\ddot{\theta}_1 + \theta_2\right) \cos(\theta_2) \\
&\quad - m_2 l_1 l_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right) \sin(\theta_2)
\end{aligned} \tag{2.17}
$$

$$
\frac{\partial L}{\partial \theta_1} = -(m_1 + m_2) g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_1 + \theta_2). \tag{2.18}
$$

After substituting (2.17) and (2.18), equation (2.15) can be rewritten as

$$
\begin{aligned}
\tau_1 &= \left[(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(\theta_2)\right] \ddot{\theta}_1 \\
&\quad + \left[m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2)\right] \ddot{\theta}_2 - m_2 l_1 l_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right) \sin(\theta_2) \\
&\quad + (m_1 + m_2) g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_1 + \theta_2).
\end{aligned} \tag{2.19}
$$

The torque at joint two is equal to

$$
\tau_2 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} \tag{2.20}
$$

where

$$
\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 + m_2 l_1 l_2 \dot{\theta}_1 \cos(\theta_2) \tag{2.21}
$$

$$
\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2^2 \left(\ddot{\theta}_1 + \ddot{\theta}_2\right)^2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_2) - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_2) \tag{2.22}
$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l_2 \left(\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2\right) \sin(\theta_2) - m_2 g l_2 \cos(\theta_1 + \theta_2). \tag{2.23}$$

After substituting (2.22) and (2.23), equation (2.20) can be rewritten as

$$\begin{aligned} \tau_2 = &\left[m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2)\right]\ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \sin(\theta_2) \\ &+ m_2 g l_2 \cos(\theta_1 + \theta_2) \end{aligned} \tag{2.24}$$

Therefore, the arm dynamics are two coupled nonlinear differential equations.

2.1.3 Manipulator Dynamics

The equations of motion for the two-link manipulator are most conveniently written in matrix form. In this form, the dynamic effects are divided into inertial and interaction parts as shown in equation (2.25).

$$\begin{aligned} &\begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \\ &+ \begin{bmatrix} m_2 l_1 l_2 \left(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right) \sin(\theta_2) \\ m_2 l_1 l_2 \sin(\theta_2) \end{bmatrix} + \begin{bmatrix} (m_1 + m_2) g l_1 \cos(\theta_2) + m_2 g l_2 \cos(\theta_1 + \theta_2) \\ m_2 g l_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \\ &= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}. \end{aligned} \tag{2.25}$$

These Manipulator dynamics are in the standard form

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \tag{2.26}$$

with $M(q)$ the inertia matrix, $V(q, \dot{q})$ the Coriolis/centripetal vector and $G(q)$ the gravity vector. Note that M(q) is symmetric.

To obtain the general robot arm dynamical equation, we determine the arm kinetic and potential energies, then the Lagrangian, and then substitute into Lagrange's equation (2.4-24) to obtain the final result[9–11].

2.2 Complete Manipulator Model

The dynamic equations derived so far only include the torques that arise from rigid body mechanics. A class of torques that are not yet included is due to friction. In reality, a robot arm is always affected by friction and disturbances. Therefore, we shall generalize the arm model by writing the manipulator dynamics as

$$M(q)\ddot{q}+V(q,\dot{q})+F(\dot{q})+G(q)+\tau_d=\tau \tag{2.27}$$

with q the joint variable n-vector and τ the n-vector of generalized forces. *M(q)* is the inertia matrix, $V(q,\dot{q})$ the Coriolis/centripetal vector, and *G(q)* the gravity vector. We have added a *friction* term

$$F(\dot{q})=F_v\dot{q}+F_d \tag{2.28}$$

with F_v the coefficient matrix of *viscous friction* and F_d a *dynamic friction* term. Also added is a *disturbance* τ_d, which could represent, for instance, any inaccurately modeled dynamics.

Since friction is a local effect, we may assume that $F(\dot{q})$ is uncoupled among the joints, so that

$$F(\dot{q})=\text{vec}\{f_i(\dot{q}_i)\}\equiv\begin{bmatrix} f_1(\dot{q}_1) \\ \vdots \\ f_n(\dot{q}_n) \end{bmatrix} \tag{2.29}$$

with $f_i(\cdot)$ known scalar functions that may be determined for any given arm. We have defined the vec $\{\cdot\}$ function for future use.

The viscous friction may often be assumed to have the form

$$F_v\dot{q}=\text{vec}\{v_i\dot{q}_i\} \tag{2.30}$$

with v_i known constant coefficients. Then $F_v = \text{diag}\{v_i\}$, a diagonal matrix with entries v_i. The dynamic friction may often be assumed to have the form

$$F_d(\dot{q}) = \text{vec}\{k_i \ \text{sgn}(\dot{q}_i)\}, \tag{2.31}$$

with k_i known constant coefficients and the signum function defined for a scalar x by

$$\text{sgn}(x) = \begin{cases} +1, & x > 0 \\ \text{indeterminate} & x = 0 \\ -1 & x < 0. \end{cases} \tag{2.32}$$

Then,

$$\text{sgn}(x) = \text{vec}\{\text{sgn}(x_i)\}. \tag{2.33}$$

A bound on the friction terms may be assumed of the form

$$\|F_v\dot{q} + F_d(\dot{q})\| \le \text{v}\|\dot{q}\| + \text{k} \tag{2.34}$$

with **v** and **k** known for a specific arm and $\|\cdot\|$ a suitable norm.

Since the arm equation has a disturbance term τ_d, we shall assume that it is bounded so that

$$\|\tau_d\| \le \text{d}, \tag{2.35}$$

where d is a scalar constant that may be computed for a given arm and $\|\cdot\|$ is any suitable norm. Friction is not an easy term to model, and indeed it may be the most contrary term to describe in the manipulator dynamics model. Some more discussion on friction may be found in[12-13].

We shall sometimes write the arm dynamics as

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_d = \tau, \tag{2.36}$$

where

$$N(q)\dot{q} \equiv V(q,\dot{q}) + F(\dot{q}) + G(q) \tag{2.37}$$

represent nonlinear terms.

There are other dynamic effects which are not included in this model. Examples include factors such as the torque due to the flexibility of the links and disturbances resulting from backlash in gear trains. Effects like these are extremely difficult to model and in reality may not contribute as much to the dynamics as the effects that have been included.

CHAPTER III

PID Computed Torque-Control

3.1 PID Control

3.1.1 Introduction

When future historians write the history of Engineering in the twentieth century, they certainly will conclude that Proportional-Integral-Derivative (PID) controllers were the most popular controllers of the century. Many thousands of Instrument and Control Engineers worldwide use such controllers in their daily work. According to a survey held in 1977, 34 out of 37 listed industrial analogue controllers were of the PID type[14]. The same is true until today and well over ninety percent of existing control loops involve PID controllers[15-16]. These controllers will remain dominant in the next century because of their remarkable effectiveness, simplicity of implementation and broad applicability. Although these controllers became commercially available in the 1930s[17], interest in their design remains very high even today. Early PID controllers were pneumatic and gained widespread industrial acceptance during the 1940s. Their electronic counterparts entered the market in the 1950s. Over the past thirty years, an enormous amount of effort has been expended in designing these controllers. Hundreds of research papers, a number of M.S./Ph.D. thesis and books[18-19] have been written on this subject during the period.

Despite these advancements and the popularity of this approach, the design of PID controllers is still a challenge for engineers and researchers. Since the 1940s, many

methods for tuning single-loop and multi-loop/multivariable PID controllers have been proposed, but every method has some limitations.

3.1.2 Proportional -Integral-Derivative (PID) Structure

About 90 to 95% of all control problems are solved by the PID controller[20], which comes in many forms. It is packaged in standard boxes for process control and in simpler versions for temperature control. It is a key component of all distributed systems for process control. Specialized controllers for many different applications are also based on PID control. The PID controller has gone through many changes in technology. The term PID is widely used because there are commercially available modules that have knobs for the user to turn to set the values of each of the three control types[21].

In some industrial control applications, good results are achieved despite a poor knowledge of a process model. The widely used proportional-integral-derivative (PID) controller can be tuned to give good performance results based on knowledge of dominant system time constants. This fact does not affect the idea that good models are required; it simply reinforces the point that models should fit their intended purpose[22].

The development of the feedback control of industrial processes has become standard. This field of feedback control, characterized by processes that are not only highly complex but also nonlinear and subject to relatively long time delays between actuator and sensor, developed the proportional-integral-derivative (PID) control. The PID controller was first described by Callender et al. in 1936[22]. This technology was based on largely experimental work and simple linearized approximations to the system dynamics. It led to standard experiments suitable to applications in the field and

eventually to satisfactory "tuning" of the coefficients of the PID controller. The PID controller is the result of combining the PI and PD controller (as shown in Figure 3.1). The transfer function of this controller is:

$$G_c(s) = \frac{E_1(s)}{E(s)} = k_d s + k_p + \frac{k_1}{s} \quad . \tag{3.1}$$

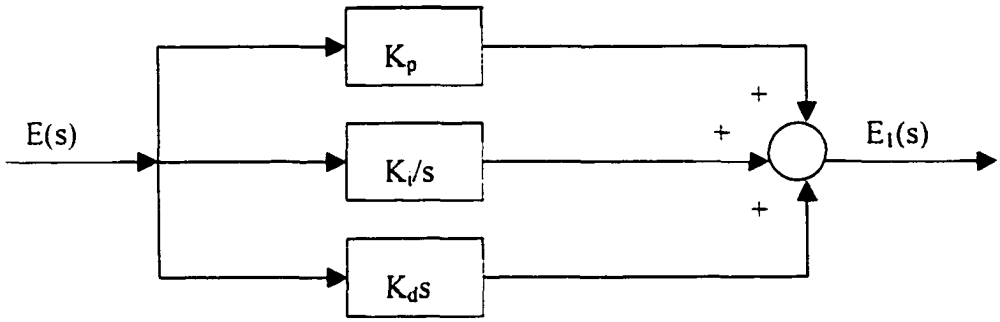


Figure 3.1 PID controller

In explaining the PID, it has been assumed that the plant to be controlled is completely known to us. In practice, this is not always the case. It may still be possible to obtain good performance of the closed-loop system by introducing a PID controller as shown in the block diagram of Figure 3.1. The arithmetic difference between a commanded input, the set-point, and the current output represents an error: how far the output must move to be at the commanded value. This error is called the proportional or P term. The time derivative of the error is the derivative or D term, and the integral of the error over time is the integral or I term. Each of these three terms is amplified by individual gains, the results are summed, and the sum is applied as the input to the controlled system.

The input to the plant consists of three components: (1) $k_p E$ which is proportional to the error; (2) $k_i \frac{E}{s}$ which is proportional to the integral of the error, and (3) $k_d sE$ which is proportional to the derivative of the error. The first component increases the loop gain of the system and thereby reduces its sensitivity to plant parameter variations. The second component increases the order of system and reduces the steady-state error. The last component tends to stabilize the system by introducing the derivative term. The values of the gain constants k_p, k_i and k_d can often be determined by trial and error if $G_c(s)$ is not known exactly. If the parameters of the plant are subject to large variations, the gain constants can be adjusted to improve the performance.

3.1.3 Example of PID Control

To illustrate the stability performance of the PID control we will use an example of DC motor[21]:

Lets consider a DC motor with the following gains: K= 5, Td= 0.0004, Ti= 0.01. A (the speed)= 10, and B (the load torque)= 50. τ_1 (the electrical time constant)= 1/60, and τ_2 (the mechanical time constant)= 1/600. We will discuss the effect of proportional, PI-, and PID control on the response of the system.

Figure 3.2 illustrate the effects of proportional, PI, and PID feedback on the step disturbance response of the system. Note that adding the integral term increases the oscillatory behavior but lowers the error and that adding the derivative term reduces the oscillation while maintaining a low error.

The response of the system to a reference step input is shown in Figure 3.3. It shows the presence of the steady-state offset for proportional control and no steady-state errors for PI or PID control. Note the reduction of the oscillatory behavior due to the addition of the derivative term.

These responses were computed using MATLAB. As an example, for the PI controller the transfer function from the disturbance input to the output is

$$\frac{Y(s)}{W(s)} = \frac{T_I B s}{T_I \tau_1 \tau_2 s^3 + T_I(\tau_1 + \tau_2)s^2 + T_I(1 + AK)s + AK}. \tag{3.2}$$

and from the reference to the output is

$$\frac{Y(s)}{R(s)} = \frac{AK(T_I s + 1)}{T_I \tau_1 \tau_2 s^3 + T_I(\tau_1 + \tau_2)s^2 + T_I(1 + AK)s + AK}. \tag{3.3}$$

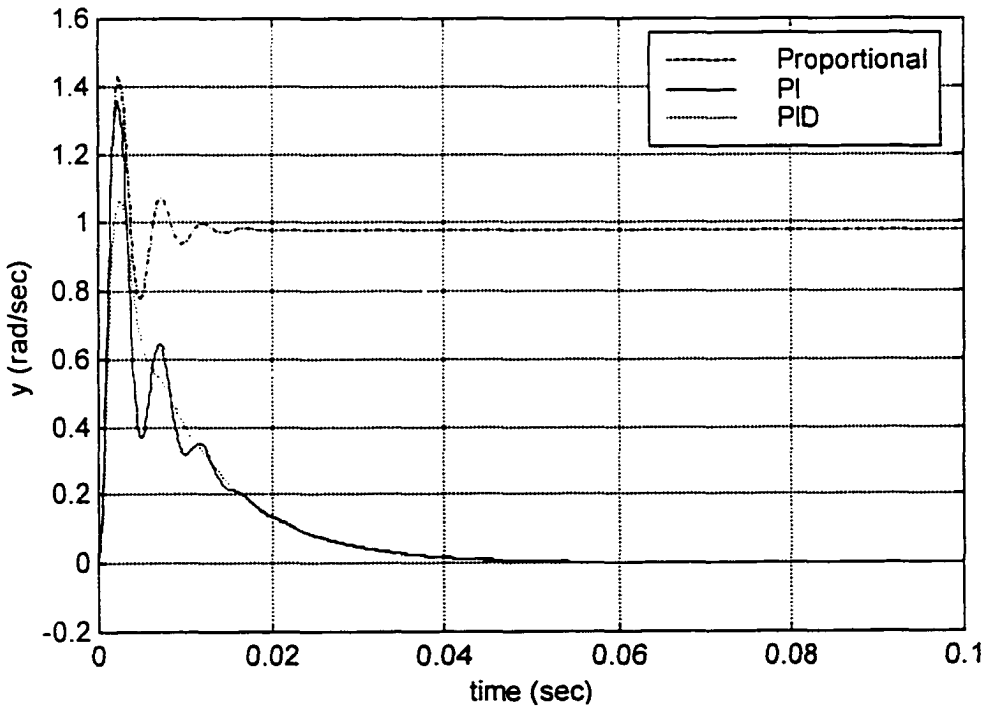


Figure 3.2 Transient response to step disturbance input

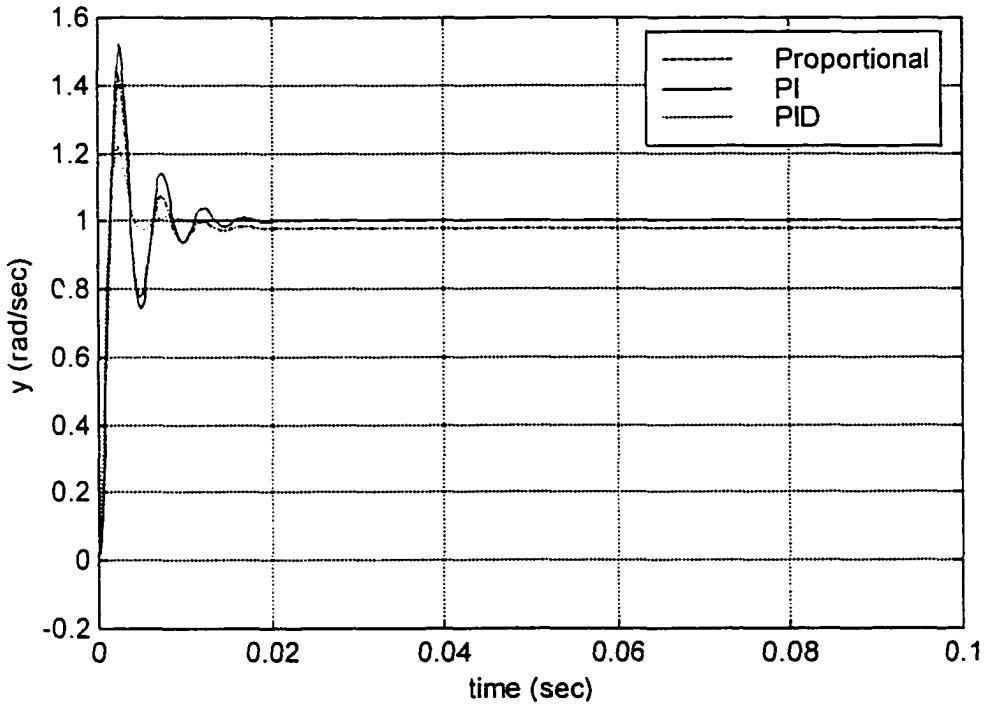


Figure 3.3 Transient response to step reference input

3.2 PID Computed-Torque Control

Through the years researchers have proposed many sorts of robot control schemes. A more sophisticated scheme in which the magnitude of nonlinear disturbing and loading torques is computed using the dynamic equations to compensate these disturbances by means of a feedforward scheme may by employed [23]. This method is the "computed torque controller" or "nonlinear control" method. Computed torque, at the same time, is a special application of *feedback linearization* of nonlinear systems, which has gained popularity in modern systems theory [24-25]. Computed-torque control allows us to conveniently derive very effective robot controllers while providing a framework to bring together classical independent joint control and some modern design techniques.

3.2.1 Derivation of Inner Feedforward Loop

The robot arm dynamics are

$$M(q)\ddot{q}+V(q,\dot{q})+F_v\dot{q}+F_d(\dot{q})+G(q)+\tau_d=\tau \tag{3.4}$$

or

$$M(q)\ddot{q}+N(q,\dot{q})+\tau_d=\tau \tag{3.5}$$

with the joint variable $q(t)\in\mathbb{R}^n$, $\tau(t)$ the control torque and $\tau_d(t)$ a disturbance.

Suppose that a desired trajectory $q_d(t)$ has been selected for the arm motion according to the discussion in[26]. To ensure trajectory tracking by the joint variable, define an output or ***tracking error*** as

$$e(t)=q_d(t)-q(t). \tag{3.6}$$

To demonstrate the influence of the input z(t) on the tracking error, differentiate twice to obtain

$$\dot{e}=\dot{q}_d-\dot{q} \tag{3.7}$$

$$\ddot{e}=\ddot{q}_d-\ddot{q}. \tag{3.8}$$

Solving now for $\ddot{q}$ in and substituting into the last equation yields

$$\ddot{e}=\ddot{q}_d+M^{-1}(N+\tau_d-\tau). \tag{3.9}$$

Defining the control input function

$$u=\ddot{q}_d+M^{-1}(N-\tau) \tag{3.10}$$

and the disturbance function

$$w=M^{-1}\tau_d, \tag{3.11}$$

we may define a state $x(t)\in\mathbb{R}^{2n}$ by

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{3.12}$$

and write the *tracking error dynamics* as

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} w. \tag{3.13}$$

This is a linear error system in Brunovsky canonical form consisting of n pairs of double integrators $1/s^2$, one per joint. It is driven by the control input $u(t)$ and the disturbance $w(t)$. Note that this derivation is a special case of the general feedback linearization [24,25,27].

The feedback linearizing transformation (3.10) may be inverted to yield

$$\tau = M(\ddot{q}_d - u) + N. \tag{3.14}$$

We call this the *computed-torque control law*. The importance of these manipulations is as follows. There has been no state-space transformation in going from (3.4) to (3.13). Therefore, if we select a control $u(t)$ that stabilizes (3.13) so that $e(t)$goes to zero, then the nonlinear control input $\tau(t)$ given by (3.14) will cause trajectory following in the robot arm (3.4). In fact, substituting (3.14) into (3.5) yields

$$M\ddot{q} + N + \tau_d = M(\ddot{q}_d - u) + N \tag{3.15}$$

or

$$\ddot{e} = u + M^{-1}\tau_d, \tag{3.16}$$

which is exactly (3.13).

The stabilization of (3.13) is not difficult. In fact, the nonlinear transformation (3.10) has converted a complicated nonlinear controls design problem into a simple

design problem for a linear system consisting of n decoupled subsystems, each obeying Newton's laws.

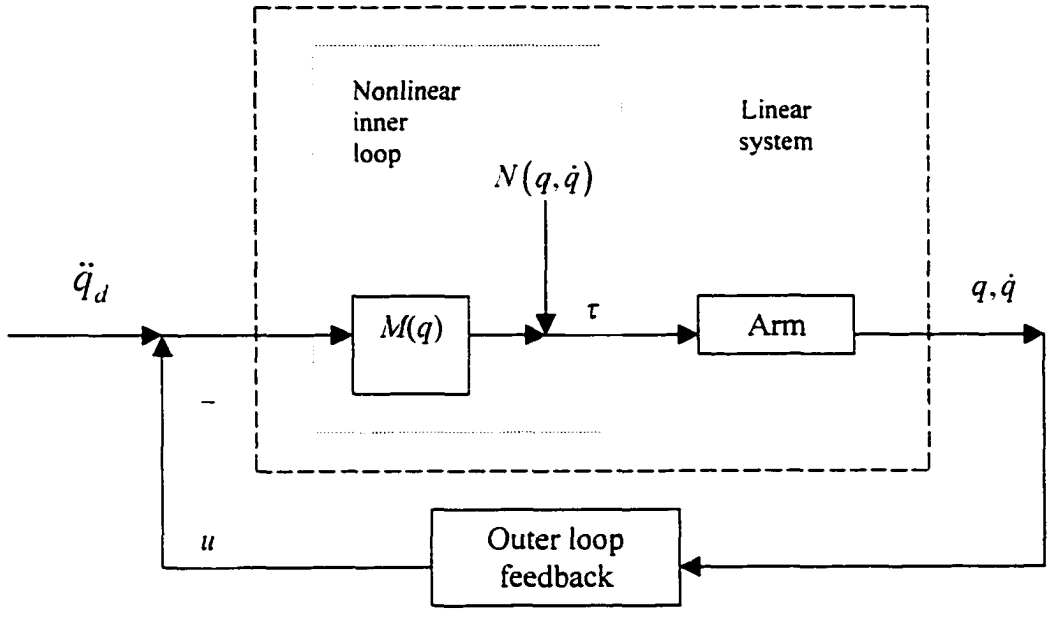


Figure 3.4 Computed-torque control scheme showing inner and outer loops

The resulting control scheme appears in Figure 3.4. It is important to note that it consists of an *inner nonlinear loop* plus an *outer control signal* $u(t)$. We shall see several ways for selecting $u(t)$. Since $u(t)$ will depend on $q(t)$ and $\dot{q}(t)$, the outer loop will be a feedback loop. In general, we may select a dynamic compensator $H(s)$ so that

$$U(s) = H(s)E(s), \tag{3.17}$$

$H(s)$ can be selected for good closed-loop behavior. According to (3.16), the closed-loop error system then has transfer function

$$T(s) = s^2 I - H(s). \tag{3.18}$$

It is important to realize that computed-torque depends on the inversion of the robot dynamics, and indeed is sometimes called *inverse dynamics control*. In fact, (3.14)

shows that $\tau(t)$ is computed by substituting $\ddot{q}_d - u$ for $\ddot{q}$ in (3.5); that is, by solving the robot *inverse dynamics problem.* The caveats associated with system inversion, including the problems resulting when the system has non-minimum-phase zeros, all apply here. (Note that in the linear case, the system zeros are the poles of the inverse. Such non-minimum-phase notions generalize to nonlinear systems.) Fortunately, the rigid arm dynamics are in minimum phase.

There are several ways to compute (3.14) for implementation purposes. Formal matrix multiplication at each sample time should be avoided. In some cases the expression may be worked out analytically. A good way to compute the torque $\tau(t)$ is to use the efficient Newton-Euler inverse dynamics formulation[28] with $\ddot{q}_d - u$ in place of $\ddot{q}(t)$.

3.2.2 PID Outer-Loop Design

Selecting proportional-plus-integral-plus derivative (PID) feedback for the auxiliary control signal $u(t)$ results in the PID computed-torque controller

$$\dot{\varepsilon} = e \tag{3.19}$$

$$u = -K_v\dot{e} - K_p e - K_i\varepsilon. \tag{3.20}$$

Then the overall robot arm control input becomes

$$\tau = M(q)\left(\ddot{q}_d + K_v\dot{e} + K_p e + K_i\varepsilon\right) + N(q,\dot{q}). \tag{3.21}$$

A block diagram of the PID computed-torque controller is shown in Figure 3.5.

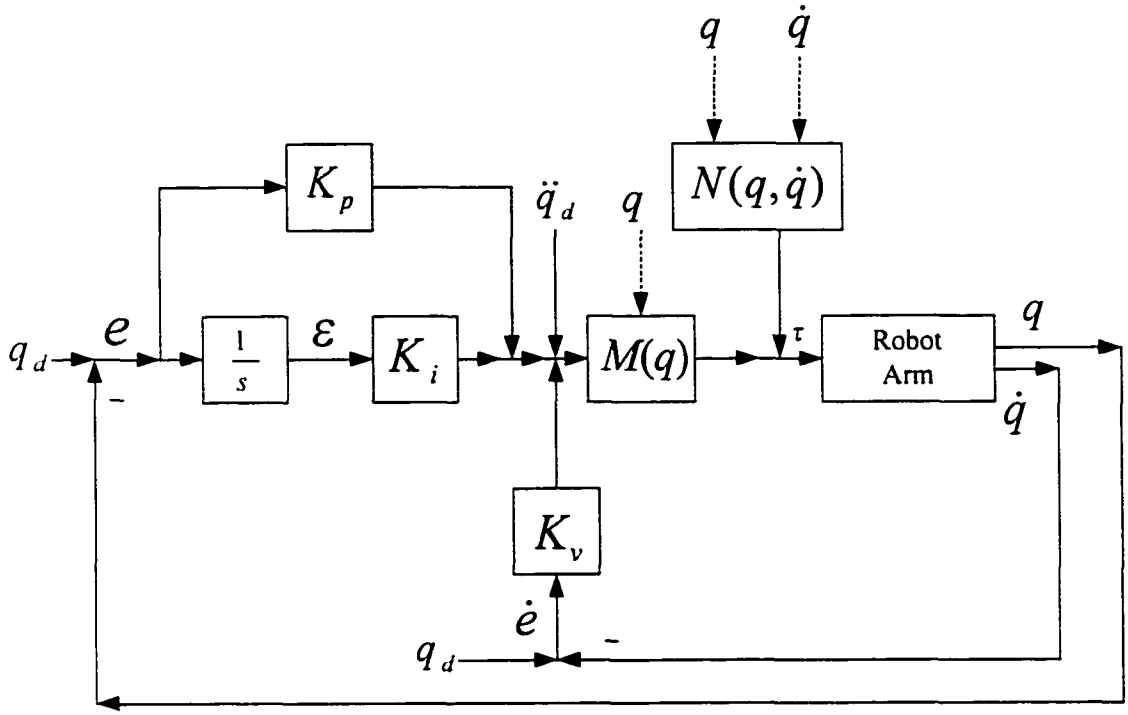


Figure 3.5 PID computed-torque controller

The closed-loop error dynamics are

$$\ddot{e} + K_v\dot{e} + K_p e + K_i\varepsilon = w \tag{3.22}$$

or in the state space form

$$\frac{d}{dt}\begin{bmatrix}\varepsilon\\ e\\ \dot{e}\end{bmatrix} = \begin{bmatrix}0 & I & 0\\ 0 & 0 & I\\ -K_i & -K_p & -K_v\end{bmatrix}\begin{bmatrix}\varepsilon\\ e\\ \dot{e}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ I\end{bmatrix} w \tag{3.23}$$

The closed-loop characteristics polynomial is

$$\Delta c(s) = \left| s^3 I + K_v s^2 + K_p s + K_i \right|. \tag{3.24}$$

3.2.2.1 Choice of PID Gains

Taking the $n \times n$ gain matrices diagonal is the usual choice for control gains, so that

$$K_v = diag\{K_{v_i}\}, \quad K_p = diag\{K_{p_i}\}, \quad \text{and} \quad K_i = diag\{K_{i_i}\}. \tag{3.25}$$

Then,

$$\Delta c(s) = \prod_{i=1}^{n}\left(s^3 I + K_{v_i} s^2 + K_{p_i} s + K_{i_i}\right), \tag{3.26}$$

and the error system is asymptotically stable as long as the K_{v_i}, K_{p_i} and K_{i_i} are all positive. Therefore, as long as the disturbance $w(t)$ is bounded, so is the error $e(t)$. Thus boundedness of $w(t)$ is equivalent to boundedness of $\tau_d(t)$.

It is important to note that although selecting the PID gain matrices diagonal results in decoupled control at the outer-loop level, it does not result in a decoupled joint-control strategy. This is because multiplication by $M(q)$ and addition of the nonlinear feedforward terms $N(q,\dot{q})$ in the inner loop scrambles the signal $u(t)$ among all the joints. Thus, information on all joint positions $q(t)$ and velocities $\dot{q}(t)$ is generally needed to compute the control $\tau(t)$ for any one given joint.

The standard form for the third-order characteristic polynomial is

$$p(s) = s^3 + (\alpha + 2\zeta\omega_n)s^2 + (2\alpha\zeta\omega_n + \omega_n^2)s + \alpha\omega_n^2 \tag{3.27}$$

with α the real root, ζ the damping ratio, and ω_n the natural frequency. Therefore, desired performance in each component of the error $e(t)$ may be achieved by selecting the PID gains as

$$k_{i_i} = \alpha\omega_n^2, \quad k_{p_i} = 2\alpha\zeta\omega_n + \omega_n^2, \quad k_{v_i} = \alpha + 2\zeta\omega_n \tag{3.28}$$

with ζ and ω_n the desired damping ratio and natural frequency for joint error i. It may be useful to select the desired responses at the end of the arm faster than near the base where the masses that must be moved are heavier.

It is undesirable for the robot to exhibit overshoot since this could cause impact if, for instance, a desired trajectory terminates at the surface of a workpiece. Therefore, the PID gains are usually selected for ***critical damping*** $\zeta = 1$. In this case

$$k_{v_i} = 2\sqrt{k_{p_i}}, \quad k_{p_i} = k_{v_i}^2/4. \tag{3.29}$$

By using the Routh test it can be found that for closed-loop stability we require that

$$k_{i_i} < k_{v_i} k_{p_i}. \tag{3.30}$$

that is, the integral gain should not be too large.

3.2.2.2 Selection of the Natural Frequency

The natural frequency ω_n governs the speed of response in each error component. It should be large for fast responses and is selected depending on the performance objectives. Thus the desired trajectories should be taken into account in selecting ω_n. We discuss now some additional factors in this choice.

There are some *upper limits* on the choice for ω_n [8]. Although the links of most industrial robots are massive, they may have some flexibility. Suppose that the frequency of the first flexible or resonant mode of link i is

$$\omega_r = \sqrt{k_r/J} \tag{3.31}$$

with J the link inertia and k_r the link stiffness. Then, to avoid exciting the resonant mode, we should select $\omega_n < \omega_r/2$. Of course, the link inertia J changes with the arm configuration, so that its maximum value might be used in computing ω_r. Another upper bound on ω_n is provided by considerations on actuator saturation. If the PID gains are too large, the torque $\tau(t)$ may reach its upper limits.

Also, the choice of the PID gains is provided from error-boundedness considerations as follows. The transfer function of the closed-loop error system in (3.22) is

$$e(s) = \left(s^3 I + K_v s^2 + K_p s + K_i\right)^{-1} w(s), \tag{3.32}$$

or if K_v, K_p and K_i

$$\varepsilon_i(s) = \frac{1}{s^3 + K_{v_i} s^2 + K_{p_i} s + K_{i_i}} w(s) \equiv H(s) w(s) \tag{3.33}$$

$$e_i(s) = \frac{s}{s^3 + K_{v_i} s^2 + K_{p_i} s + K_{i_i}} w(s) \equiv sH(s) w(s) \tag{3.34}$$

$$\dot{e}_i(s) = \frac{s^2}{s^3 + K_{v_i} s^2 + K_{p_i} s + K_{i_i}} w(s) \equiv s^2 H(s) w(s). \tag{3.35}$$

We assume that the disturbance and M^{-1} are bounded, so that

$$\|w\| \le \left\|M^{-1}\right\| \|\tau_d\| \le \overline{md} \tag{3.36}$$

with $\overline{m}$ and $\overline{d}$ known for a given robot arm. Therefore,

$$\|\varepsilon_i(t)\| \le \|H(s)\| \|w\| \le \|H(s)\| \overline{md} \tag{3.37}$$

$$\|e_i(t)\| \le \|sH(s)\| \|w\| \le \|sH(s)\| \overline{md} \tag{3.38}$$

$$\|\dot{e}_i(t)\| \leq \|s^2 H(s)\| \, \|w\| \leq \|s^2 H(s)\| \overline{md}. \tag{3.39}$$

Now selecting the L_2 -norm, the operator gain $\|H(s)\|_2$ is the maximum value of the Bode magnitude plot of $H(s)$. For a critically damped system

$$\sup_{\omega} \|H(j\omega)\|_2 = 1/k_{i_i}. \tag{3.40}$$

Therefore,

$$\|\varepsilon_i(t)\|_2 \leq \overline{md}/k_{i_i} \tag{3.41}$$

and

$$\sup_{\omega} \|j\omega H(j\omega)\|_2 = 1/k_{p_i}, \tag{3.42}$$

so that

$$\|e_i(t)\|_2 \leq \overline{md}/k_{p_i}. \tag{3.43}$$

Moreover,

$$\sup_{\omega} \|j\omega^2 H(j\omega)\|_2 = 1/k_{v_i}, \tag{3.44}$$

so that

$$\|\dot{e}_i(t)\|_2 \leq \overline{md}/k_{v_i}. \tag{3.45}$$

Thus, in the case of critical damping, the position error decreases with k_{p_i}, the velocity error decreases with k_{v_i}, and the steady-state error decreases with k_{i_i}.

The reader is referred to [29–31] for a complete discussion relative to the PID computed-torque control.

CHAPTER IV

Sliding Mode in Variable Structure Systems

This chapter is divided into two parts. The first part clarifies the details of Variable Structure System theory with sliding mode control and summarizes the previous research done on Variable Structure Control (VSC) with Sliding Mode Control (SMC). To clearly explain the theory of Variable Structure Systems, a detailed discussion is given in the first section of this part. The second section shows how the Variable Structure System theory can be used to design a Sliding Mode Controller. The design method is demonstrated through a simple example, which clearly defines the steps and the assumptions that are made. The last section of this part discusses the comprehensive research that has been published on Variable Structure Control in the last fifteen years. The purpose of the second part of this chapter is to clearly explain the derivation of an effective Sliding Mode Control algorithm and to show how this algorithm is applied to a two-link robotic manipulator.

4.1 Background of Variable Structure Systems

4.1.1 Variable Structure System (VSC) with Sliding Mode

Variable structure control (VSC) with sliding mode is a special type of control technique that is capable of making a control system very robust with respect to system parameter variations and external disturbances. In addition, the technique provides an easy way to design the control law for a plant, linear or nonlinear.

VSC was pioneered in the Soviet Union in the early 1950's by S. V. Emelyanov and his cohorts[33]. The technique did not receive wide attention in the Western world until recently. This was mainly due to a number of problems, including the lack of a practical design procedure, existence of chattering in the system, need of measuring all state variables and scarcity of literature in this area in English. Since the late 1970's, researchers have discovered additional attractive properties of VSC and have developed methods for control law design. The feasibility of the technique not only has been predicted by theory but also has been demonstrated by numerous computer simulations and hardware experiments. As a result, VSC technique has become mature and ready to be applied.

To explain the theory of Variable Structure Systems, a simple example will be discussed. Consider the second-order system,

$$\ddot{x} - \xi \dot{x} + \Psi x = 0, \quad \xi > 0 \tag{4.1}$$

where Ψ is changed discontinuously according to

$$\Psi = \begin{cases} \alpha & \text{if } xs > 0 \\ -\alpha & \text{if } xs < 0 \end{cases} \quad \alpha > 0 \tag{4.2}$$

and the switching line, s, is given by the following relationship[32]:

$$s = \dot{x} + cx \,. \tag{4.3}$$

If $\Psi = \alpha$, the state-space trajectory behaves as an unstable focus; if $\Psi = -\alpha$, then the state-space trajectory is hyperbolic. Both of these systems are unstable individually. However, if the state-space is divided up and portions of each trajectory are used, the combination can be stable. Figure 4.1 shows the state space trajectories of a stable

Variable Structure System where c, the slope of the switching line, is less than the slope of the stable asymptote of the hyperbola.

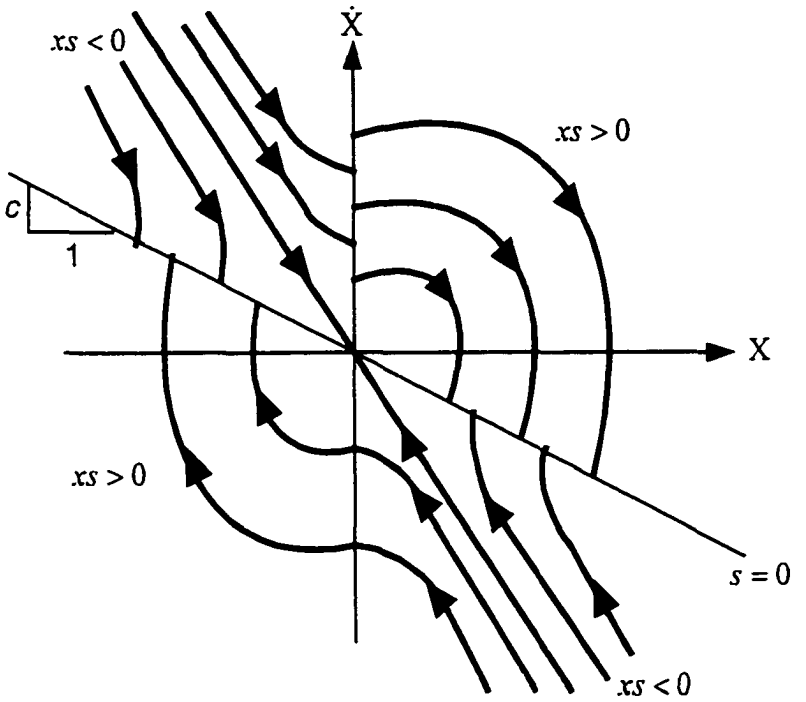


Figure 4.1 Variable structure state-space trajectories

To insure that the phase plane trajectories are directed towards the switching line, $s = 0$, the following inequalities must be satisfied:

$$\lim_{s\to-0} \dot{s} > 0 \quad \text{and} \quad \lim_{s\to+0} \dot{s} < 0. \tag{4.4}$$

These inequalities are based upon the following Lyapunov function, $V(s)$, which ensures that s will be equal to zero in a finite time.

$$V(s) = \tfrac{1}{2}s^2 \tag{4.5}$$

which leads to

$$\frac{dV}{dt} = s\dot{s} \leq -\eta|s|, \quad \eta > 0 \ . \tag{4.6}$$

This condition maintains the robustness of the system because, once the trajectory reaches the switching line, it is forced to remain on it and stay in sliding mode. After integration, equation (4.6) also guarantees that the time it takes to reach the switching line, t_r, is bounded according to

$$t_r \leq \frac{|s(t=0)|}{\eta} \ . \tag{4.7}$$

4.1.2 Sliding Mode Control

A Sliding Mode Controller can be designed using the principles shown above. Consider the following second-order system,

$$\ddot{x} = f + u \tag{4.8}$$

where u is the control input and x is the system variable[10]. The dynamics, f, are not exactly known and estimated by $\hat{f}$. The estimation error on f is assumed to be *bounded* by some known function $F = F(x, \dot{x})$ with the constraint

$$\left|\hat{f} - f\right| \leq F \ . \tag{4.9}$$

In order for the system to track a desired position and velocity, x_d and $\dot{x}_d$ respectively, a sliding surface is defined as

$$s = \dot{e} + \lambda e, \tag{4.10}$$

where

$$e = x - x_d \quad \text{and} \quad \dot{e} = \dot{x} - \dot{x}_d \ . \tag{4.11}$$

Taking the derivative of equation (4.10) with respect to time and making a substitution for $\ddot{e}$ yields

$$\dot{s} = \ddot{x} - \ddot{x}_d + \lambda\dot{e} = f + u - \ddot{x}_d + \lambda\dot{e} \,. \tag{4.12}$$

When in ideal sliding mode, both s and $\dot{s}$ are equal to zero. The best approximation of the control law that would achieve $\dot{s} = 0$ is obtained from equation (4.12) as

$$\hat{u} = -f + \ddot{x}_d - \lambda\dot{e} \,. \tag{4.13}$$

In order to satisfy the sliding condition given by equation (4.6) despite uncertainty in the dynamics input f, a discontinuous term is added to $\hat{u}$ to make up the total control input, u,

$$u = \hat{u} - b\ \mathrm{sgn}(s) \tag{4.14}$$

where $b = b(x, \dot{x})$ is a variable gain for the discontinuous part of the control input and sgn(s) is the signum function defined as

$$\mathrm{sgn}(s) = \frac{|s|}{s}, \quad s \neq 0. \tag{4.15}$$

Using equations (4.12) to (4.14) the discontinuous term gain, b, can be chosen such that the sliding condition is guaranteed,

$$s\dot{s} = s\left[f - \hat{f} - b\ \mathrm{sgn}(s)\right] = \left(f - \hat{f}\right)s - k|s| \le -\eta|s| \tag{4.16}$$

so that

$$b \ge F + \eta \,. \tag{4.17}$$

As F, which represents the disturbances and parameter variations in the model, increases, the discontinuity in the control input must be increased by increasing the gain, b. A Sliding Mode Controller designed in this fashion assures that the system will reach the sliding surface and remain in sliding mode. This, in turn, assures that the position and velocity errors approach zero asymptotically.

The major problem with the above analysis is that it assumes that are ideal sliding mode can be realized. In reality, no switching mechanism could change the state of the control input fast enough to keep the system trajectory exactly on the sliding surface and in sliding mode. There is a certain switching delay that allows the state trajectory to cross the sliding surface before the control input is changed to force it back on to the sliding surface. This causes the trajectory to chatter along the sliding surface, and as a result, the trajectory only stays within the ideal sliding mode. Figure 4.2 shows a typical phase plane trajectory for a Sliding Mode Controller. The first portion of the trajectory is the reaching phase which begins at the initial condition (*I.C.*) and ends when the sliding surface, $s = 0$, is reached. From this point on the system is in sliding mode and will continue towards the origin until the position and velocity errors, e and $\dot{e}$ respectively, become zero. Figure 4.2 also illustrates the chattering effect along the sliding surface. The system trajectory appears to bounce from one side of the sliding surface to the other until the origin is reached.

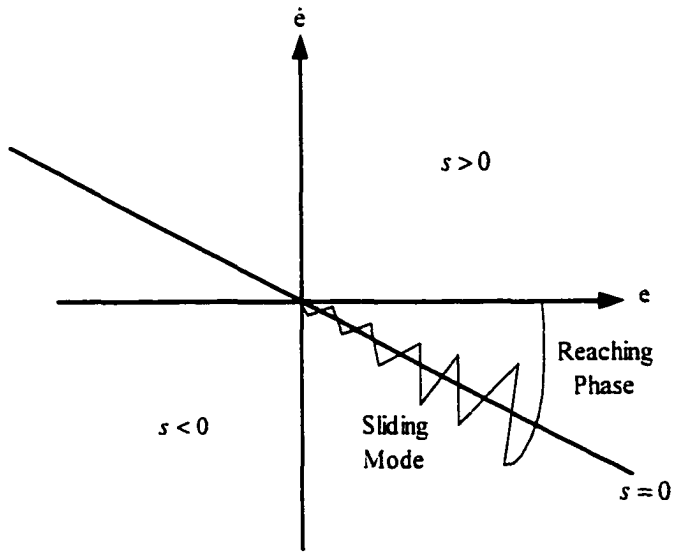


Figure 4.2 Typical SMC phase plane trajectory

In order to insure that the system remains stable, the sliding condition must be met in the neighborhood of the sliding surface that the trajectory chatters inside. This is the reason equation (4.6) is bounded by $-\eta|s|$ and not zero, which is customary for Lyapunov functions. By having the more stringent constraint, sliding mode with its inherent chattering remains stable.

Chattering can pose problems for Sliding Mode Controllers used on robotic manipulators. The reason for concern is that manipulators can have high natural frequencies which might be excited by a control signal which chatters when in sliding mode. If those natural frequencies are excited, the controller will make the system unstable instead of keeping it stable. Since chattering is so undesirable, a lot of effort has been made to find ways to eliminate it. The various methods that have been used will be a large part of what is explained in the following section.

4.1.3 Summary of Previous Research

The theory of Variable Structure Systems was first developed in the U.S.S.R. during the 1950's. Most of that work has not been translated and remains in the Russian language. Today we must rely on survey articles that assemble and report on what was learned during that time. V. I. Utkin has written three such papers. His first survey was published in 1977[32] and sums up most of what was known about Variable Structure Systems up to that point. He begins with the basic principles of VSS and applies them to time-invariant and time-varying plants as well as multi-input systems. Utkin proves the disturbance rejection properties of Variable Structure Systems and the existence of a sliding mode. In Utkin's second survey, published in 1983[33], more time is spent on the

mathematical aspects of the description of Variable Structure Systems with emphasis on recent research trends. Only a few applications are mentioned because he states that he wanted to stimulate new ways of approaching VSC and not give simply a list of results. A third article by Utkin, published in 1987[34], sketches the entire scope of scientific problems within the framework of sliding mode control theory. He reviews the most important research trends in the field of Variable Structure Control and shows its promise for a wide range of applications.

A survey published in 1988 by DeCarlo, *et al.*[35] is similar to Utkin's articles in the way disturbance rejection and sliding modes are proven and areas of needed research are pointed out. The difference, however, is in the way the authors lay out the steps involved in designing a Variable Structure Controller. In addition to proving that the favorable qualities of Variable Structure Control exist, ways to take advantage of these attributes are given. This ideal is taken a step further by Walcott and Zak[36] in an article that describes four experiments which demonstrate the fundamental principles of VSC and can be repeated in a senior/graduate level controls laboratory course.

The first time Variable Structure Systems theory was applied to the control of a robotic manipulator was in 1978 in an article by K. D. Young[37]. This was an important step because the complexity of manipulators makes them difficult to control. In the absence of friction and other disturbances, the dynamics of a n degree-of-freedom manipulator can be written as n coupled second-order nonlinear differential equations.

Variable Structure Control had already been used for a wide range of applications in the steel, power, chemical and aerospace industries and was valued for being a very robust strategy that does not require accurate modeling. For these reasons, it showed

great potential for use on robotic manipulators, so Young designed a Variable Structure Controller for a two-joint manipulator and simulated point-to-point indexing on a hybrid analog-digital computer. His simulation demonstrates the applicability of the Variable Structure Control approach to manipulator control design as well as its ability to eliminate nonlinear dynamic interactions of the manipulator joints by introducing sliding modes. Young concludes that VSC is much easier to implement than existing methods and requires less a *priori* knowledge of the manipulator dynamics. It is also mentioned that the Variable Structure Controller produces a rapidly changing discontinuous control signal that might adversely affect the physical hardware of the manipulator. Young, however, does not indicate how to remedy the high frequency control activity. He only states that the simulated manipulator must have filtered out a lot of the high-frequency behavior because the joint position trajectories were smooth.

The next paper to address the application of Variable Structure Control to robot manipulators was written in 1983 by Slotine and Sastry[38]. The authors begin by discussing discontinuous differential equations and proving sliding mode existence, giving reference to earlier work reported in the Soviet literature. The design with sliding mode control is illustrated for single and multiple input, linear and non-linear time--varying systems. The designs are proved to be robust assuming an ideal sliding mode. Due to switching delays, ideal sliding cannot be achieved in reality so the authors use a continuous control law to approximate ideal sliding more closely. Instead of using the discontinuous signum function they use an interpolation called the saturation function defined as,

$$\text{sat}(y)=\begin{cases} y & \text{if } |y|\le 1 \\ \text{sgn}(y) & \text{if } |y|>1 . \end{cases} \tag{4.19}$$

Equation (4.14) now appears as,

$$u=\hat{u}-k\ \text{sat}\left(\frac{s}{\Phi}\right) \tag{4.20}$$

where 2Φ is the boundary layer thickness of the control input. Graphically, the control input would appear as shown in Figure 2.3 with $u=\hat{u}$ when $s=0$.

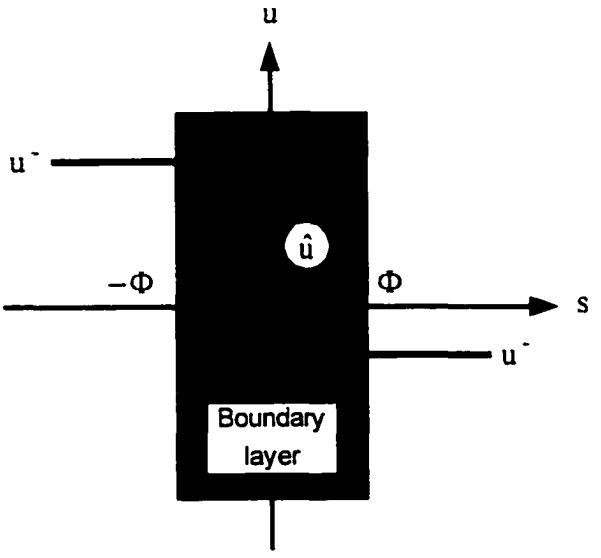


Figure 4.3 Control input interpolation

The saturation function results in a boundary layer surrounding the sliding line which is illustrated in Figure 4.4. One half of the boundary layer width, ε, represents the maximum error during approximate sliding mode, expressed as equation (4.21), where λ is the slope of the sliding line:

$$|e|\le\frac{\Phi}{\lambda^{n-1}}=\varepsilon. \tag{4.21}$$

Slotine and Sastry used their Continuous Sliding Mode Control theory to simulate a two-link manipulator during trajectory control. They found that the disturbance rejection properties of SMC were preserved while not generating the undesirable chattering phenomenon that was apparent when the control input was discontinuous. The sacrifice for interpolating the discontinuous control signal is that the tracking accuracy can only be guaranteed to lie within half of the boundary layer width, ε. The advantage is that the control torques are smooth and do not excite the unmodeled high-frequency dynamic modes of the manipulator.

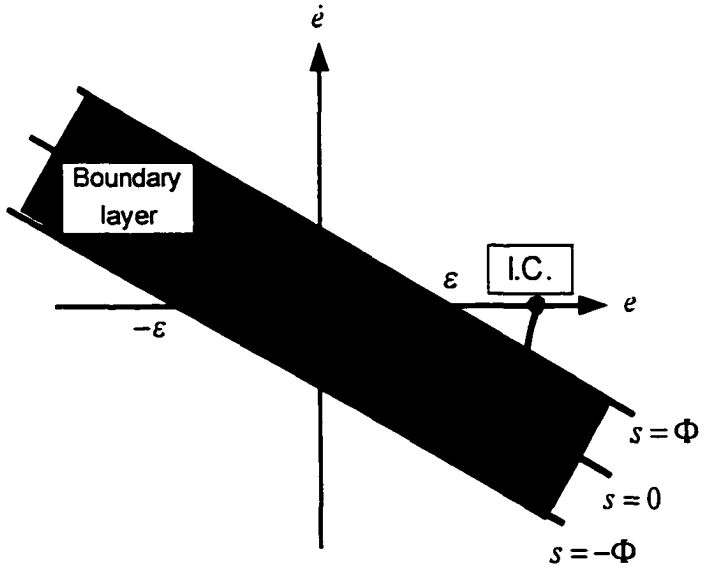


Figure 4.4 Sliding line boundary layer

An alternative to using a boundary layer to make the control input continuous, is the "conti" function which has been used to approximate the signum function[39–41]. A small positive constant, δ, is used which distorts the discontinuity according to

$$\text{conti}(s) = \frac{s}{|s| + \delta} \quad . \tag{4.22}$$

If $\delta = 0$ then conti(s) = sgn(s). A drawback to this method of smoothing is that there is no physical meaning for δ. It is simply a distortion parameter with no units.

Sliding Mode Control design is not limited to the method discussed in Section 4.1.2. That method is called the equivalent control method because the control law consists of two elements. One element is the (equivalent) non-linear compensation part which is of low frequency and would keep the trajectory on the sliding surface in the absence of parametric variations and disturbances. The second element, by being discontinuous, provides the high frequency part of the control input and overcomes any effects of estimated or incomplete dynamic modeling.

All Sliding Mode Controllers use the theory of sliding surfaces to insure that position and velocity errors approach zero. In order to make sure the attractiveness of the sliding surface, Lyapunov's method is most often used. However, it is not the only method that can be used. Morgan and Ozguner[42] used the regulated derivative control algorithm and showed it to be effective for the control of a robotic manipulator. The objective of this control algorithm is to regulate the derivative of the switching variable, $\dot{s}$, to a constant R. To accomplish this, the following law is implemented.

$$\dot{s} = -R\,\mathrm{sgn}(s), \quad R > 0 \tag{4.23}$$

It can be seen that $s\dot{s} < 0$ will always be satisfied, assuring the attractiveness of the sliding surface and existence of sliding mode. A boundary layer was not used by Morgan and Ozguner or by Choi and Jayasuriya[43], in a similar implementation, to ease the chattering action. Therefore, considerable chattering was evident and alternate methods to reduce it were not proposed.

A drawback common to all of the methods discussed so far has been the need to take the inverse of the manipulator's inertial matrix when computing the discontinuous gains to suppress disturbances. This is a problem because the terms of the inertial matrix are complicated functions of the manipulator's position and having to take the inverse of the matrix can raise the number of operations exponentially. Although this does not affect the computation done by the controller at each control interval, it makes the design of the controller much more difficult than is needed. By choosing a Lyapunov function equal to,

$$V(s) = \tfrac{1}{2} s^T M(\theta) s \tag{4.24}$$

where $M(\theta)$ is the inertial matrix of the manipulator and s^T is the transform of the vector of sliding surfaces, $s(t)$, defined as,

$$s(t) = C\, e(t) + \dot{e}(t) \tag{4.25}$$

where $C = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, and $\lambda_i > 0$ are the slopes of the sliding surfaces for each of the individual joints of the manipulator[52]. The derivative of the Lyapunov function is,

$$\frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s + s^T M(\theta) \dot{s} \tag{4.26}$$

and must satisfy the following inequality to insure the attractiveness of each of the sliding surfaces:

$$\frac{dV}{dt} \le -s^T K\, s \le 0, \tag{4.27}$$

where $K = \mathrm{diag}(k_1, k_2, \ldots, k_n)$, and $k_i > 0$ represent the gains that increase the speed of approach to the sliding surface in the reaching phase.

The simple insertion of the inertial matrix into the Lyapunov function makes the design of the control algorithm exceedingly simple compared to other approaches,

especially in the case of multi-link manipulators whose inertial matrix is extremely difficult to invert. There are two examples in literature of the implementation of this design technique that do not use a boundary layer around the sliding surfaces[44–45]. The impressive accuracy and robustness of the controller to varying loads is shown although chattering is apparent. When a boundary layer is used[46–47], the control input is completely smooth while maintaining excellent disturbance rejection characteristics. Similar to previous results, however, the presence of the boundary layer effects the accuracy because the trajectory is only guaranteed to stay within the boundary layer width.

A convenient way to make sure the accuracy provided when a boundary layer is not used, while suppressing most of the chattering in sliding mode, is to alter the boundary layer width. If the boundary layer width shrinks to zero as the origin is approached, then the best of both situations can be preserved[48–49]. An illustration of this altered boundary layer is shown in Figure 4.5.

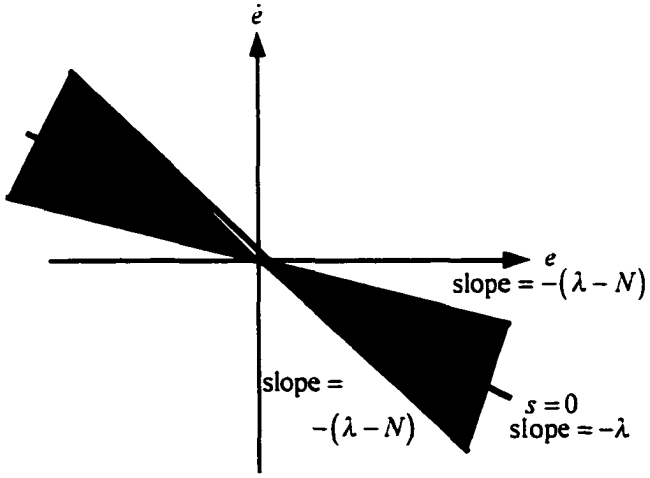


Figure 4.5 Altered sliding line boundary layer

To create a boundary layer of this type, the saturation function of equation (2.19) is used in the control law as,

$$u = \hat{u} - k \text{ sat}\left(\frac{s}{n}\right) \tag{4.28}$$

where the variable boundary layer half thickness, n, changes according to,

$$n = N|e|, \quad 0 < N < \lambda. \tag{4.29}$$

A boundary layer of this kind cannot suppress chattering at the state plane origin. However, if a small amount of chattering can be tolerated, the average steady-state error will be smaller than the steady-state error if the boundary layer width was constant. This method of boundary layer design was used with the enhanced Lyapunov function, Equation (4.24), by Yeung and Chen[50]. They showed the implementation of this design method for use in a set-point regulation problem for a two-link robotic manipulator. The controller proved to have all of the favorable qualities that Sliding Mode Controllers can possess without the harmful tendency to chatter while in the transient sliding mode. Chattering was unfortunately found in the steady-state about the set-point resulting from the absence of a boundary layer. The authors mention that the chattering could be eliminated by using Slotine's approach[51] discussed earlier. By suggesting the use of the "balance condition," Yeung and Chen effectively defeat the purpose of using the altered sliding line boundary layer which they recommend.

4.2 Sliding Mode Controller Design

In Part 4.1, an overview was given of the various types of Sliding Mode Controllers that have appeared in past literature. All of the methods were proven

successful applications of the theory of Variable Structure Systems with sliding mode. The strategies that possess the best design are the ones that do not require taking the inverse of the manipulator's inertial matrix. In Chapter Two, the dynamic model of a two-link manipulator was derived and shown to be a highly non-linear, coupled, second-order system. These types of complicated systems are the most challenging ones to control. The purpose of this part is to clearly explain the derivation of an effective Sliding Mode Control algorithm, first published in 1988 by Chen, *et al.* [46], and to show how this algorithm is applied to a two-link robotic manipulator.

4.2.1 Development of the Control Algorithm

The first step in the design of any controller is the formulation of the dynamic model of the system that is to be controlled. This task was completed in the previous part and the results will be carried over into this part. The purpose of designing a Sliding Mode Controller is to show that the theory of Variable Structure Systems with Sliding Mode can be used in controller design. The proof is in the successful simulation. To make the simulation as realistic as possible, two different manipulator models will be used. The most complete model will be used in the integration to find the movement of the arm due to the applied torques. The second, a rather approximate model, will be used in the design of the controller. This is done because the exact model of an actual manipulator can never be found and used in the design of a controller.

The model that will be used for the development of the control algorithm is given below as equation (4.30). It is shown in the typical form for manipulator models and has approximate values for the masses.

$$\tau = M(\theta)\ddot{\theta} + N(\theta,\dot{\theta}) \tag{4.30}$$

where

$$M(\theta) = \hat{M} + \Delta M$$
$$N(\theta,\dot{\theta}) = \hat{N} + \Delta N. \tag{4.31}$$

The cap, '^', denotes the assumed value and the delta, 'Δ', denotes the estimating error.

The sliding surface is defined as equation (4.30) and depends on the vector of position and velocity errors, e and $\dot{e}$, and the slope matrix, C.

$$s(t) = C\, e(t) + \dot{e}(t) \tag{4.32}$$

where

$$e(t) = \theta(t) - \theta_d(t)$$
$$\dot{e}(t) = \dot{\theta}(t) - \dot{\theta}_d(t) \tag{4.33}$$
$$C = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \text{ and } \lambda_i > 0.$$

If the manipulator inertial matrix, $M(\theta)$ is multiplied by the derivative of equation (4.32), and a substitution is made for $\ddot{e}$, the result is

$$M(\theta)\dot{s} = M(\theta)\ddot{\theta} - M(\theta)\ddot{\theta}_d + M(\theta)C\dot{e}(t). \tag{4.34}$$

Equation (4.30) can be used to substitute $\tau - N(\theta,\dot{\theta})$, for $M(\theta)\ddot{\theta}$ giving,

$$M(\theta)\dot{s} = \tau - N(\theta,\dot{\theta}) - M(\theta)\ddot{\theta}_d + M(\theta)C\dot{e}(t). \tag{4.35}$$

When in ideal sliding mode, both s and $\dot{s}$ are equal to zero. The best approximation of the control law that would achieve $\dot{s} = 0$ is obtained from equation (4.35) as,

$$\hat{\tau} = \hat{M}\ddot{\theta}_d - \hat{M}C\dot{e} + \hat{N}. \tag{4.36}$$

A discontinuous signum function and a proportional term are subtracted from the "equivalent control torque" of equation (4.36) to make up the complete control torque defined as equation (4.37).

$$\tau = \hat{\tau} - Q\,\mathrm{sgn}(s) - Ps \tag{4.37}$$

where

$$\mathrm{sgn}(s) = \frac{|s|}{s}, \quad s \neq 0$$
$$Q = \mathrm{diag}(q_1, q_2, \ldots, q_n)$$
$$P = \mathrm{diag}(p_1, p_2, \ldots, p_n).$$

The discontinuous and proportional terms are included in the control torque to insure that the manipulator's joint trajectories converge onto the sliding surface, despite parametric variations and disturbances. The control gain matrices, Q and P, are chosen using Lyapunov theory. A candidate Lyapunov function, $V(s)$, is shown below. $M(\theta)$ is the manipulator inertial matrix and s^T is the transpose of the sliding surface.

$$V(s) = \tfrac{1}{2} s^T M(\theta) s \tag{4.38}$$

Inertial matrices are by definition positive definite. With the inclusion of $M(\theta)$ and the s^T terms, equation (4.38) will always remain positive definite which is a requirement for a Lyapunov matrix function. A favorable consequence of including the inertial matrix in the candidate Lyapunov function and in the derivation of the "equivalent control torque", (4.36), is that the inverse of the inertial matrix does not need to be found. This is an important advantage because the inverse of the inertial matrix can be very difficult to compute.

It should be noted that the inverse of the inertial matrix is needed for the integration scheme discussed in chapter 2. Integration, however, is only necessary during a simulation and not required for the use of a Sliding Mode Controller on an actual manipulator.

According to Lyapunov's theorem, if the following condition is maintained, $s(t)$ will always approach zero, guaranteeing the existence of sliding mode and the stability of the controller.

$$\frac{dV}{dt} \leq -s^T K s \tag{4.39}$$

where

$$K = \text{diag}(k_1, k_2, \ldots, k_n), \text{ and } k_i > 0. \tag{4.40}$$

Taking the derivative of the Lyapunov function gives

$$\frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s + s^T M(\theta) \dot{s}. \tag{4.41}$$

Using equation (4.35) in a substitution for $M(\theta)\dot{s}$ yields

$$\frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s + s^T \left[\tau - N - M\ddot{\theta}_d + MC\dot{e} \right]. \tag{4.42}$$

Equations (4.36) and (4.37) are, then substituted into the previous result giving,

$$\begin{aligned} \frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s + s^T \Big[& \hat{M}\ddot{\theta}_d - \hat{M}C\dot{e} + \hat{N} - Q\ \text{sgn}(s) - Ps \\ & - N - M\ddot{\theta}_d + MC\dot{e} \Big] \end{aligned} \tag{4.43}$$

Grouping the exact and assumed terms allows a substitution with the estimating error terms of equation (4.31) resulting in

$$\begin{aligned} \frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s + s^T \Big[& \left(M - \hat{M}\right) C\dot{e} - \left(M - \hat{M}\right)\ddot{\theta}_d - \left(N - \hat{N}\right) \\ & - Q\ \text{sgn}(s) - Ps \Big]. \end{aligned} \tag{4.44}$$

Substituting equations (4.31) yields

$$\frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s + s^T \left[\Delta MC\dot{e} - \Delta\hat{M}\ddot{\theta}_d - \Delta N - Q\ \text{sgn}(s) - Ps \right]. \tag{4.45}$$

The final result will be divided into two parts and must collectively satisfy condition (4.39) as follows:

$$\frac{dV}{dt} = \tfrac{1}{2} s^T \dot{M}(\theta) s - s^T P s + s^T \left[\Delta M C \dot{e} - \Delta \hat{M} \ddot{\theta}_d - \Delta N \right] - s^T Q \operatorname{sgn}(s)$$
$$\frac{dV}{dt} = \dot{V}_1 + \dot{V}_2 \le -s^T K s \tag{4.46}$$

where

$$\dot{V}_1 = -s^T \left[P - \tfrac{1}{2} \dot{M}(\theta) \right] s \le s^T K s \tag{4.47}$$

$$\dot{V}_2 = s^T \left[\Delta M C \dot{e} - \Delta M \ddot{\theta}_d - \Delta N \right] - s^T Q \operatorname{sgn}(s) \le 0. \tag{4.48}$$

Using equation (4.37) and the Gerschgorin theorem[62], the elements of P should be chosen as

$$p_i = \sum_{j=1}^{n} \tfrac{1}{2} \dot{M}_{ij}^{\max} + k_i \quad i = 1, \ldots, n \tag{4.49}$$

where

$$\dot{M}_{ij}^{\max} \ge \left| \dot{M}_{ij} \right|. \tag{4.50}$$

This insures that the following matrix, which is equal to $\left[P - \dot{M}(\theta)/2 \right]$, is always positive definite and greater than or equal to K, satisfying condition (4.47).

$$\begin{bmatrix} \sum_{j=1}^{n} \frac{\dot{M}_{1j}^{\max}}{2} - \frac{\dot{M}_{11}}{2} + k_1 & -\frac{\dot{M}_{12}}{2} & \cdots & -\frac{\dot{M}_{1n}}{2} \\ -\frac{\dot{M}_{21}}{2} & \sum_{j=1}^{n} \frac{\dot{M}_{2j}^{\max}}{2} - \frac{\dot{M}_{22}}{2} + k_2 & \cdots & -\frac{\dot{M}_{2n}}{2} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\dot{M}_{n1}}{2} & -\frac{\dot{M}_{n2}}{2} & \cdots & \sum_{j=1}^{n} \frac{\dot{M}_{nj}^{\max}}{2} - \frac{\dot{M}_{nn}}{2} + k_n \end{bmatrix}$$

Using equation (4.48) the elements of Q should be chosen as

$$q_i = \sum_{j=1}^{n} \left\{ \Delta M^{\max} C \right\}_{ij} \left| \dot{e}_j \right| + W_i^{\max} + \Delta N_i^{\max} \tag{4.51}$$

where

$$\begin{aligned} \Delta M_{ij}^{\max} &\geq \left| \Delta M \right|_{ij} \\ W_i^{\max} &\geq \left| \Delta M^{\max} \ddot{\theta}_d^{\max} \right|_i \\ \Delta N_i^{\max} &\geq \left| \Delta N_i \right| \end{aligned} \tag{4.52}$$

Once these conditions for choosing the control gain matrices, Q and P, are met, conditions (4.47) and (4.48) will always be true, which in turn assures that the original condition (4.39) is satisfied. It can be seen from this development that the joint control torques of equation (4.37) will force the manipulator joint trajectories onto the sliding surface despite parametric variations within the given limits. This means that the joints of the manipulator will follow the desired joint paths given by vectors $\theta_d(t)$, $\dot{\theta}_d(t)$ and $\ddot{\theta}_d(t)$.

The few requirements that need to be fulfilled are the existence of an approximate model of the manipulator and an estimate of the amount of error in the approximation. In addition, bounds must be placed on the maximum velocity and acceleration for each joint of the manipulator. This need for *a priori* knowledge of the manipulator is minimal and characteristic of the information required by the majority of controllers used today.

4.2.2 Chattering Suppression

A conclusion that can be drawn about the chattering effect, discussed in Part One, is that chattering appears as a result of the discontinuous part of the control torque.

Therefore, every technique that has been used to eliminate chattering and does not involve the use of observers shares one characteristic in common, a boundary layer, surrounding the sliding surface, to smooth the discontinuity across it. A few implementations have used varying widths for the boundary layer while most have kept the controller simple and used a constant width. For the control algorithm developed in this chapter, a constant boundary layer width will be used. Therefore, instead of the discontinuous signum function used in the control torque equation (4.37), a smooth saturation function, defined as equation (4.53), will be used.

$$\text{sat}(s/n)=\begin{cases} s/n & \text{if } |s| \le n \\ \text{sgn}(s/n) & \text{if } |s| > n, \end{cases} \tag{4.53}$$

where the boundary layer half-width, n, is defined as

$$n=[n_1, n_2, \ldots, n_n]^T, \text{ and } n_i > 0. \tag{4.54}$$

The control torque equation now appears as

$$\tau = M\ddot{\theta}_d - MC\dot{e} + \hat{N} - Q\ \text{sat}(s/n) - Ps. \tag{4.55}$$

CHAPTER V

Fuzzy Logic Control

Fuzzy logic control (FLC), which is based on fuzzy logic theory, has become one of the most important fields in artificial intelligence. Since the foundation of fuzzy logic by Zadeh in 1965[52], many theoretical and experimental researches in the area of fuzzy control have been performed. Fuzzy logic, on which fuzzy control is based, is much closer to the human thinking and natural language than the traditional logical systems. Basically, it provides an effective means of capturing the approximate, inexact nature of the real world. The essential part of FLC is a set of linguistic rules related by the dual concepts of fuzzy implication and compositional rule of inference. Therefore, it provides an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy.

The methodology of the FLC appears very useful when the process under control is very complex for analysis using the conventional mathematical modeling. Many industrial systems are of this nature, however, they can be controlled by human operators without knowing their underlying dynamics. So, fuzzy logic control may be viewed as a step toward a rapport between conventional precise mathematical control and human-like decision making. Experience shows that the FLC are superior to conventional control methods for ill-defined systems or for systems where no mathematical model have been developed. The structure of the FLC which is nonlinear and its ability to deal with imprecise data makes it more robust than the linear controllers.

There are several advantages of using fuzzy control over classical control methods. As Lotfi Zadeh, who is considered the father of fuzzy logic, once remarked: "In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper." Therefore, this chapter clarifies the details of fuzzy logic control (FLC). In section 5.1, a background on fuzzy logic control is discussed with some application of its use. To clearly explain the theory of fuzzy logic control, a detailed discussion is given in section 5.2, which presents the basic definitions and the basic fuzzy mathematics required for implementing FLC. In section 5.3, the structure of FLC with the main parameters of the FLC is discussed in detail and the algorithm of FLC is given. Finally, in section 5.4 conclusions and comments are given on the FLC.

5.1 Background of Fuzzy Logic Control

Fuzzy logic was first proposed by Lotfi A. Zadeh of the University of California at Berkeley in a 1965 paper[52]. He elaborated on his ideas in a 1973 paper that introduced the concept of "linguistic variables," which in this chapter equates to a variable defined as a fuzzy set. Other research followed, with the first industrial application is a cement kiln built in Denmark in 1975. Fuzzy systems were largely ignored in the US because they were associated with artificial intelligence (AI).

The Japanese interest in fuzzy systems was sparked by Seiji Yasunobu and Soji Miyamoto of Hitachi. In 1985, they provided simulations that demonstrated the superiority of fuzzy control systems for the Sendai railway; their ideas were adopted and fuzzy systems were used to control accelerating, braking and stopping when the line opened in 1987. In addition, in 1987 the international meeting of fuzzy researchers in

Tokyo helped promote interest in fuzzy systems. In that year, Takeshi Yamakawa demonstrated the use of fuzzy control (through a set of simple dedicated fuzzy logic chips) in an "inverted pendulum" experiment. The experiment was a classic control problem in which a vehicle tries to keep a pole mounted on its top by a hinge upright by moving back and forth.

During the last decade, there has been a rapid growth in the number and variety of applications of fuzzy logic. Applications range from consumer products such as washing machines, cameras and microwave ovens to industrial process control, the subway system in the city of Sendai, Japan[53], automobile transmission control[54-55], air conditioner control[56], robot control (Hirota Laboratory at Hosei University)[57], speech recognition (Ricoh)[57], control of cement kiln processes[58], fuzzy control of a model car[59], elevator control[60], nuclear reactor control[61-62], fuzzy logic controlled hardware systems[63-64], fuzzy computers[65] and the Old Dominion University wireless mobile Lego robot (WLMR)[66].

5.2 Fuzzy Sets and Fuzzy Logic

5.2.1 Fuzzy Logic

Fuzzy logic is the logic on which fuzzy control is based; it is much closer to the human thinking and natural language than the traditional logical system[67]. It can be viewed as a multi-level logic in comparison with the two levels [0,1] logic, so fuzzy logic is the best way to represent imprecise ideas easily.

5.2.1.1 Fuzzy Sets Versus Crisp Sets

The central concept of fuzzy logic is the fuzzy sets which provide basis for systematic way for manipulation of vague and imprecise concepts[68]. In particular, we can employ fuzzy sets to represent linguistically defined data such as cold, hot, etc. temperatures. Fuzzy sets differ from the conventional crisp sets in the idea of degree of membership function μ_F. For example, the set of cold temperatures in Norfolk it defined between [10,30]°F.

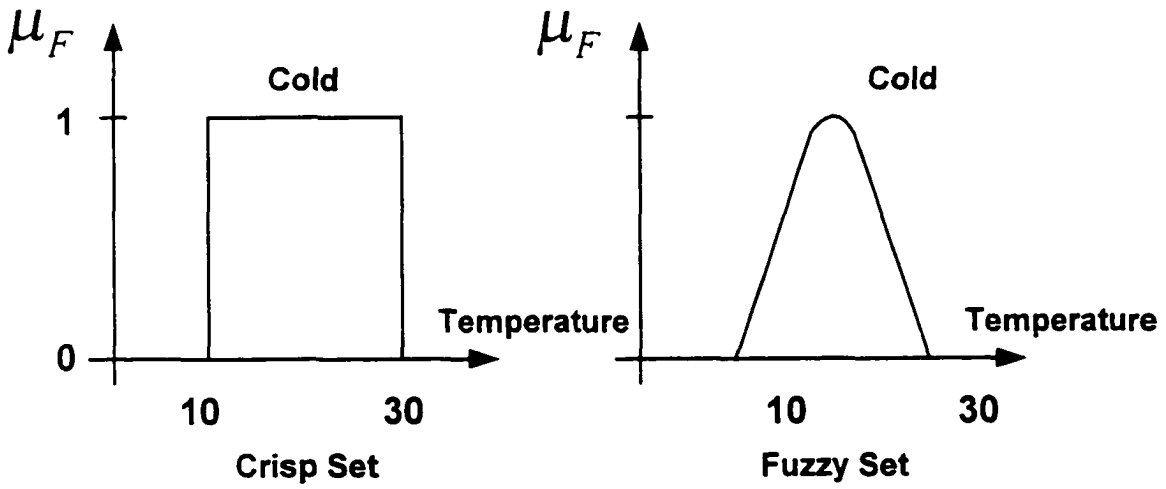


Figure 5.1 Crisp and fuzzy sets

The crisp set defining this linguistic variable {Cold Temperature} is defined as:

$$\mu_F = \begin{Bmatrix} 1, \text{ Temperature} \in [10,30] \\ 0, \text{ Temperature} \notin [10,30] \end{Bmatrix},$$

i.e.. in crisp sets the degree of membership function takes only 0 or 1. On the other hand, the fuzzy set defining this linguistic variable has a degree of membership function which takes values between [0,1] and has the certain distribution:

$$\mu_F = fn.(\text{Temperature}).$$

It is now apparent how fuzzy logic is identical to the human thinking in which there is no crisp definitions of variables rather than fuzzy.

5.2.1.2 Fuzzy Set and Terminology[67-68]

Definitions

1. Universal of discourse [U]: Is the collection of elements denoted generically b(u) and represents the domain of variables.

2. Fuzzy set [F]: A fuzzy set F on U is characterized by a membership function, μ_F, which takes values between [0,1]. It can be viewed as a generalization of the concept of an ordinary (crisp) set whose membership function takes only 0 or I. Thus a fuzzy set F in U is represented as:

$$F = \left\{u, \mu_F(u) \middle| u \in U\right\};$$

i.e., a set of ordered pairs of u and its degree of membership as either a discrete or continuous function.

3. Support, Crossover, Fuzzy Singleton: The support of a fuzzy set F is the crisp set of all points u in U such that: $\mu_F > 0$. The point at which $\mu_F(u) = 0.5$ is called the crossover point, and the fuzzy set whose support is a single point in U is referred to as fuzzy singleton.

4. Fuzzy Number: A fuzzy number $F \in U$ is a fuzzy set which is normal and convex; i.e.:

$$1- \max_{u \in U} \mu_F(u) = 1 \text{ ; Normal}$$

$$2- \mu_F(\lambda u_1 + (1-\lambda)u_2) \geq \min(\mu_F(u_1), \mu_F(u_1)); \text{ Convex}$$

where; $u_1, u_2 \in U; \lambda \in [0,1]$

5. Linguistic Variable: A linguistic variable is defined by the $\{u, T(u), U, R, S\}$ in which U is the variable name; $T(u)$ is the term set of u or the set of names of the linguistic values with each value being a fuzzy number defined on U; R is a syntactic rule for generating the names of u and S is the semantic rule for associating with each value its meaning.

For example, if temperature is interpreted as a linguistic variable, then its term set T(temp.) could be {Cold; Comfortable; Hot}, where each term in T(temp.) is characterized by a fuzzy number in a universe of discourse U=[0,50] ; these terms are given by the shown triangular fuzzy sets.

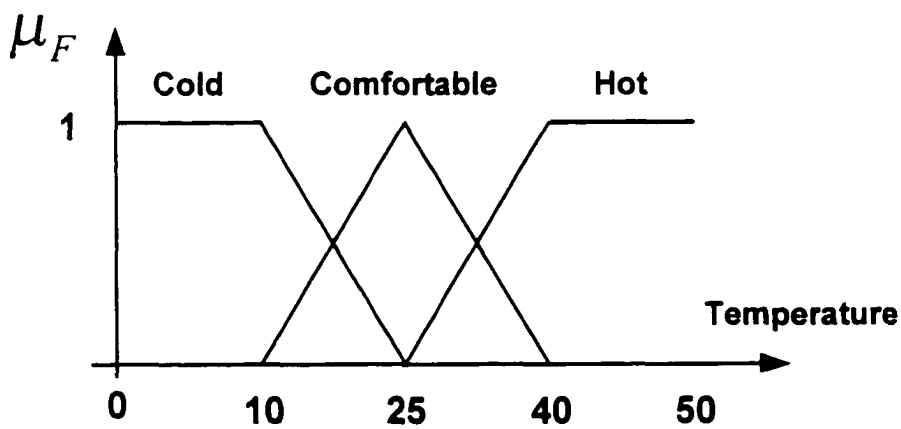


Figure 5.2 Linguistic variables

5.2.2 Fuzzy Mathematics

Let A, B be 2 fuzzy sets in U with membership function μ_A , μ_B respectively, the set operations of union, intersection and complement of fuzzy sets are defined via their membership functions as:

1. Union:

$$\mu(A \cup B) = \max\{\mu_A(u), \mu_B(u)\};$$

2. Intersection:

$$\mu(A \cap B) = \min\{\mu_A(u), \mu_B(u)\};$$

3. Complement:

$$\mu(A') = \{1 - \mu_A(u)\};$$

4. Cartesian Product:

If $A_1, \ldots, A_n$ are fuzzy sets in $U_1, U_2, \ldots, U_n$ respectively, then the Cartesian product of $A_1, \ldots, A_n$ is a fuzzy set in the product space $U_1 \times \ldots \times U_n$ with the membership function:

$$\mu_{A1 \times A2 \ldots \times An}(u_1, u_2, \ldots u_n) = \min\{\mu_{A1}(u_1), \ldots, \mu_{An}(u_n)\}; \quad \text{Minimum}$$

$$\mu_{A1 \times A2 \ldots \times An}(u_1, u_2, \ldots u_n) = \{\mu_{A1}(u_1) * \mu_{A2}(u_2) \ldots * \mu_{An}(u_n)\}; \quad \text{Product}$$

5. Fuzzy Relation:

An n-array fuzzy relation is a fuzzy set in $U_1 \times U_2 \ldots \times U_n$ and is expressed as:

$$R_{U_1 \times U_2 \ldots \times U_n} = \{((u_1, u_2, \ldots, u_n), \mu_R(u_1, u_2, \ldots, u_n)) | (u_1, \ldots, u_n) \in U_1, \ldots, U_n\}$$

6. Fuzzy Algorithm:

A group of rules collected together with the **OR** operator to form an algorithm; the relational matrix for the complete algorithm is found using the union operator of the whole m rules:

$$R = \cup_{i=1,2,\ldots,m}\{R_1, R_2, \ldots\ldots, R_m\};$$

7. Compositional Rule of Inference:

If R is a fuzzy relation $U \times V$ and A is a fuzzy set in U, then the fuzzy set B in V include by A is given by:

$$B = A * R; \quad \text{i.e. } \{A \text{ composition } R\}.$$

A well known special case is Zadeh's compositional rule of inference which uses the max-min operator with the membership function:

$$\mu_B(V) = \max_{A \in U} \{\min[\mu_A(u), \mu_B(v)]\}$$

and the max-product operator:

$$\mu_B(V) = \max_{A \in U} \{[\mu_A(u).\mu_B(v)]\}$$

5.3 Fuzzy Logic Control (FLC)

5.3.1 FLC Structural Issues

The basic FLC configuration shown in figure (5.3) comprises four main principal components; *Fuzzification, Knowledge Base, Decision Making Logic, and Defuzzification.*

1. The fuzzification involves two main functions, scaling the crisp controller inputs and converting them into the corresponding linguistic values.
2. The Knowledge base (KB) contains the necessary fuzzy sets definitions and the rule base which contains the rules characterizing the control policy.
3. The decision making logic (fuzzy algorithm) is responsible of inferring fuzzy control action based on the input fuzzy sets. It has the capability of simulating human decision making.
4. The defuzzification yields a non fuzzy (crisp) output based on the output fuzzy sets and scale it to the corresponding universe of discourse.

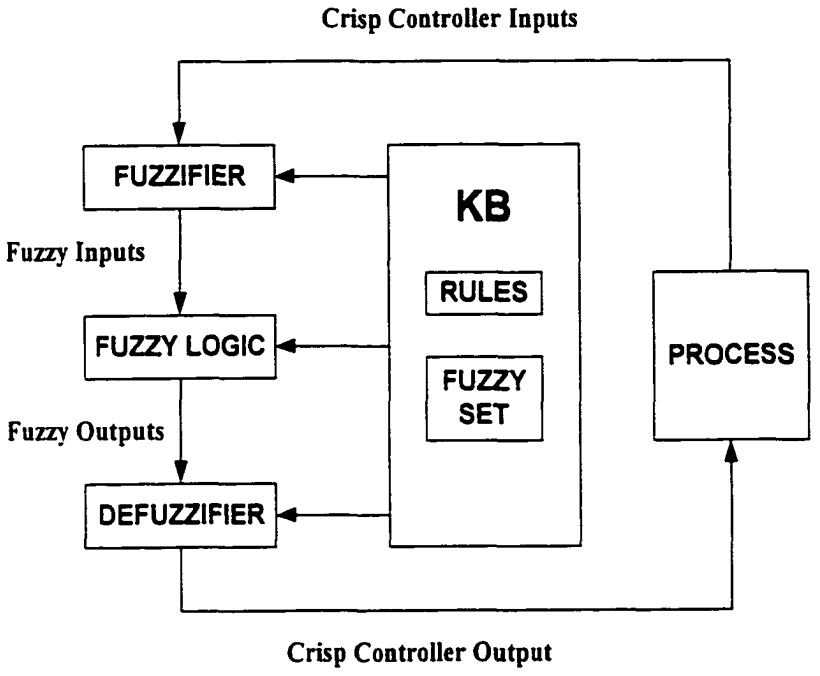


Figure 5.3 Basic fuzzy logic structures

5.3.2 Design Parameters of FLC[69]

The main design parameters of any FLC are the following:

1. Fuzzification operator.

2. Database:

   a) Discretization levels and normalization.

   b) Definition of fuzzy sets.

3. Rulebase:

   a) Choice of input variables and output variables of fuzzy control rules.

   b) Derivation of control rules (source).

4. Decision making logic:

a) Fuzzy implication operator (Mamadani or Larsen).

b) Inference operator.

5. Deftizzification strategies.

Now, we discuss each of the design parameters:

1. Fuzzifiaction:

In this step, a mapping of the observed crisp inputs into fuzzy sets is performed since all the operations of the FLC are based on fuzzy sets. In control application, the fuzzification operator is usually a fuzzy singleton; hence, no fuzziness is introduced in this case. This strategy has been widely used due to its simplicity; it interprets a crisp input e_o as a fuzzy set E with the membership function $\mu_E(e)$ equals to zero except at the point e_o at which it equals 1.

Other fuzzification operators have been used to put into consideration the randomness in the input crisp variables if they are noisy. An isosceles triangle was chosen to be a fuzzification operator for such cases with the vertex of this triangle at the mean value, while the base is twice the data standard deviation.

2. Data Base:

The Knowledge base (**KB**) of a FLC consists of two components namely a data base and rule base. The data base contains the necessary data for data manipulation in the FLC such as scaling factors, mapping functions, discretization levels and fuzzy sets definition.

a) Discretization/Normalization of universe of discourse:

Usually, the measured input data is transformed into a normalized universe of discourse (usually [-6,6]) using the mapping function:

$$x_i^n = F(G_{ui} * x_i),$$

where G_{ui} is the i-th input scaling factor. The mapping function F may be linear or nonlinear.

Sometimes the universe of discourse is discretized (quantized) into certain number of levels, for which the fuzzy sets are defined discretely. For this case, a look up table based on the discrete universes, which defines the output of the controller for all possible combinations of the input signals, can be implemented by off-line processing of the fuzzy algorithm in order to shorten the running time of the controller.

The choice of the number of quantization levels has an essential influence on how fine a control can be obtained.

b) Fuzzy Sets

The selection of the number of linguistic terms which describe the input and output variables and the shape of the membership function is a heuristic cut and trial operation in FLC cannot be found optionally with a definite method. However, these parameters are very flexible in their selection and can be related to the physics of the system.

The membership function can be defined discretely for discretized universes or functionally for continuous universes. Many shapes for the fuzzy sets have been used in control applications such as:

- Triangular

- Trapezoidal

- Gaussian (exponential)

- Sinusoidal.

For simplicity, triangular fuzzy sets with seven linguistic terms (NB, NM, NS, ZO, PS, PM, PB) are used in this dissertation for the input and output variables of FLC

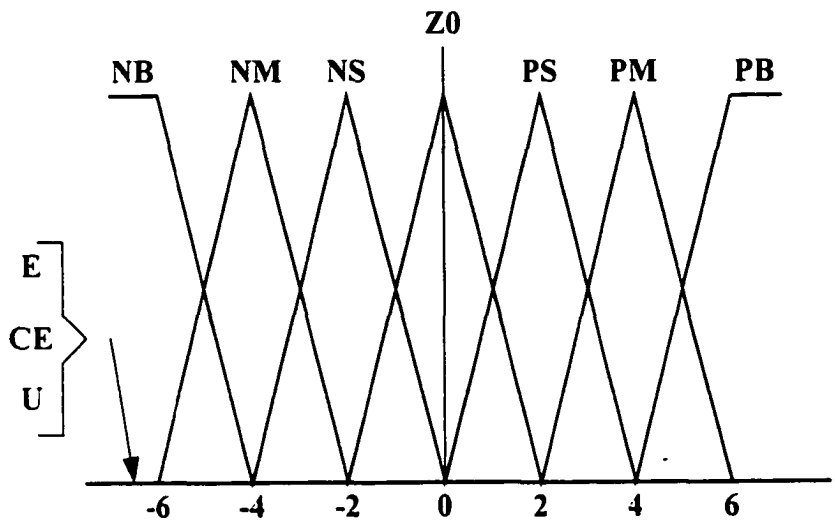


Figure 5.4 Fuzzy sets

where these terms have the following meaning:

NB negative big

NM negative medium

NS negative small

Z0 zero

PS positive small

PM positive medium

PB positive big

3. Rule Base:

The important part of any FLC is that it linguistically defines the control policy of the controller. Fuzzy control rules are usually in the form of conditional statements:

If <(antecedent)> Then <(consequent)>

which are easy to implement by fuzzy conditional statements in fuzzy logic.

a) Choice of Control Variables of Fuzzy Control Rules:

Fuzzy control rules are more conveniently formulated in linguistic rather than numerical terms. The proper selection of process state variables of the antecedent and consequent of fuzzy rules is essential to the characterization to the operation of the fuzzy controller. Experience with the controlled process plays an important role in the selection of the input and output variables. The controller input variables are usually chosen as the state, state error, state error derivative, state error integral.....etc.

For example, the fuzzy PD controller has control rules in the form:

If error is E_i and change in error is CE_i then control input is U_i

where E_i, CE_i and U_i are linguistic variables for the system error, change in error and control input respectively.

b) Source of Control Rules:

The derivation of the fuzzy rules is the most important part in FLC design and implementation. There are four main techniques for the derivation of the control rules,

and the best method depends on the system under consideration. They also can be used in combination.

1. Technique Based on Control Engineering Knowledge:

Fuzzy control rules can be derived heuristically based on some knowledge of the controlled process or by analyzing the behavior of the controlled process in the time domain, such that the deviation from the desired trajectory is minimized at each point. For example, if we consider the second order process time response shown in figure (5.5), a control rule can be generated at each point to follow the set point as follows:

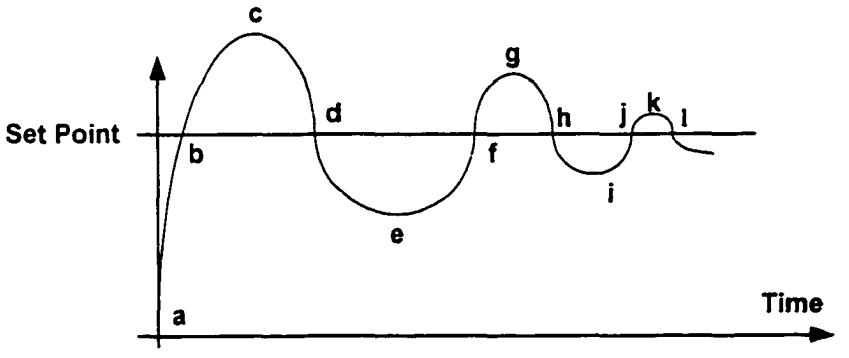


Figure 5.5 2nd order process time response

point (a): E is NB & CE is Z0 then U is PB

point (b): E is Z0 & CE is PB then U is NB

point (c): E is PB & CE is Z0 then U is NB

.

.

point (l): E is Z0 & CE is NS Then U is PS

A generalized PI Control rules for minimizing error and error integral was derived by MacVicar-Whelan. Using this method, we can find the effect of individual rules on the system performance measures such as overshoot, rise time...etc. Of course this method will not be effective for every system and there must be another systematic methods for rules generation.

u	Error Integral						
Error	NB	NM	NS	Z0	PS	PM	PB
NB	PB	PB	PB	PB	PM	PS	Z0
NM	PB	PB	PM	PM	PS	Z0	NS
NS	PB	PM	PS	PS	Z0	NS	NM
Z0	PM	PM	PS	Z0	NS	NM	NM
PS	PM	PS	Z0	NS	NS	NM	NB
PM	PS	Z0	NS	NM	NM	NB	NB
PB	Z0	NS	NM	NB	NB	NB	NB

Table 5.1 Generalized PI control rules

2. Technique Based on Operators Control Action:

In many industrial man-machine control systems the input-output relations are not known with sufficient precision to make it possible to employ classical control theory for modeling and simulation. Yet, skilled human operators can control such systems quite successfully without having any quantitative models in mind. In effect, a human operator employs a set of control rules. This is the main advantage of FLC that it can incorporate in the same structure both linguistic data from skilled human operator and also crisp data from the observed process input-output data.

This also gives the FLC its popularity in industrial control of very complex systems for which no mathematical models exist but which can be controlled by an expert human.

3. Technique Based on Learning Algorithms:

For many systems the two techniques mentioned earlier fail to give the desired performance, especially for multi-input multi-output (MIMO) systems and also for higher order systems. This leads to the adding of human learning capabilities to FLC by adding another set of meta rules which exhibit human-like learning ability to create and modify the general rule base based on the desired overall performance of the system.

4. Technique Based on Fuzzy Models of the Process:

For any process a fuzzy model can be generated from the input output observed data and some knowledge about the system. Based on these fuzzy models we can generate a set of fuzzy control rules for attaining optimal performance of a dynamic system. The set of fuzzy control rules forms the rule base of a FLC. Although this approach is somewhat more complicated, it yields better performance, especially for low order systems, and provides a more systematic method for generating the optimal rules and the theoretical analysis for FLC. However, this approach will be inapplicable if the system order is higher than second order.

4. Decision Making Logic:

a) Fuzzy Implication (composition) Operator:

By fuzzy implication function we mean the function that constraints the relations between the antecedents and consequents of a conditional fuzzy statement. The membership function value for each rule for a given controller input is calculated by

fuzzy implication (composition). Let us take the following example where the rule R_i is expressed as:

$$R_i: \quad \text{If } e \text{ is } E_i \text{ then } u \text{ is } U_i$$

Then the fuzzy implication is expressed as the Cartesian product of the antecedent (s) and consequent (s): $R_i = E_i \times U_i$

There are several definitions for implication, the most widely used in control applications are ***Minimum*** (Mamadani) and the ***Product*** (Larsen):

$$\mu_{Ri}(e,u) = \min\left\{\mu_{Ei}(e), \mu_{Ui}(u)\right\}; \qquad \text{Minimum}$$
$$\mu_{Ri}(e,u) = \left\{\mu_{Ei}(e).\mu_{Ui}(u)\right\}; \qquad \text{Product}$$

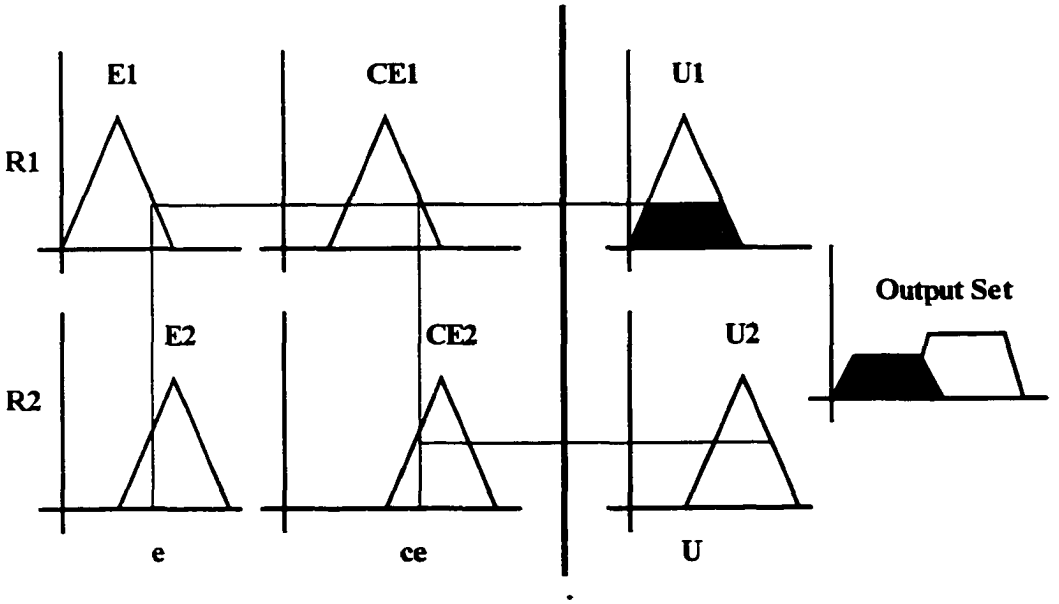
b. Fuzzy Inference

By inference we mean obtaining the controller output fuzzy set from the controller input and the control rules by compositional rule of inference.

The two main operators, defined before, are the max-min (Zadeh) and the max-product.

The fuzzy reasoning algorithm (implication and inference) for a set of control rules can be performed in two ways, either by rule matrices or rule by rule.

In the first method, which is suitable for discrete universes, the rule matrix for each rule



R_i is obtained throughout fuzzy implication. Then, the final control rule for the whole algorithm is obtained using the Union of all the matrices R_i. Finally, the output fuzzy set is obtained by the compositional rule of inference. In comparison, in the second, rule by rule method, figures (5.6) and (5.7), the membership function for each rule and the corresponding output fuzzy set is first obtained using either min. or product implication operators. Then, the final output fuzzy set is obtained using the union (max.). This method is more general since it is suitable for both discrete and continuous universe of discourse.

Figure 5.6 Graphical interpretation for rule by rule fuzzy decision making using Min. fuzzy implication

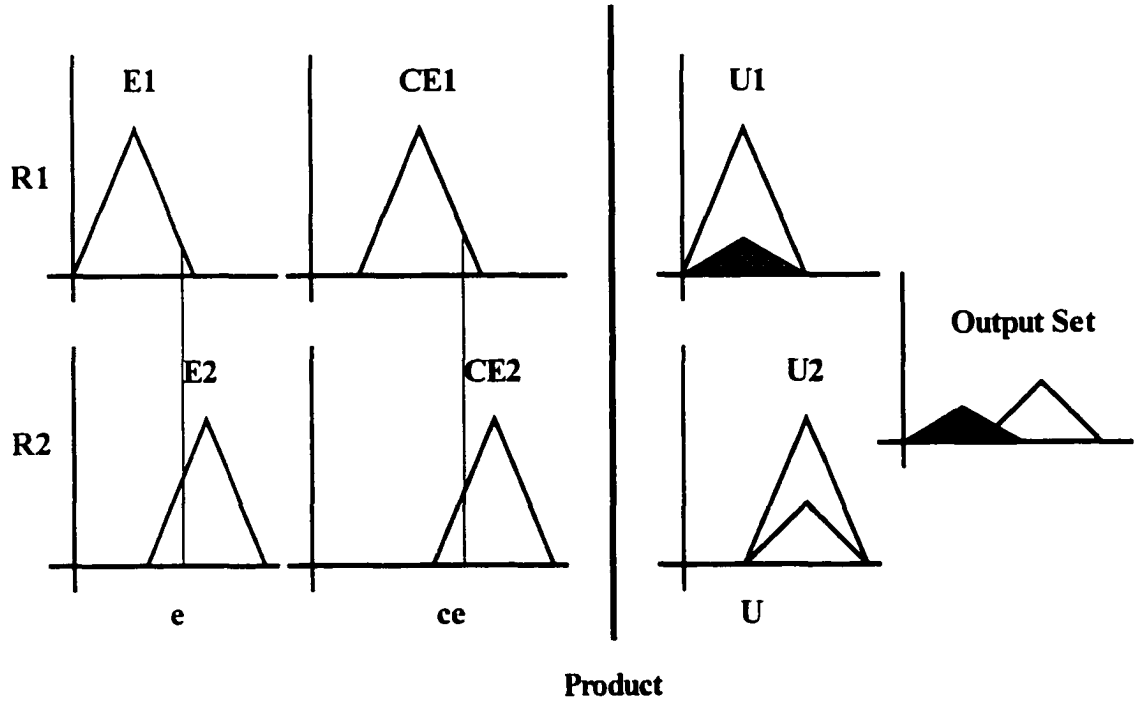


Figure 5.7 Graphical interpretation for rule by rule fuzzy decision making using Product fuzzy implication

5. Defuzzification Method:

Basically, defuzzification is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a non fuzzy (crisp) control actions as required by the process. A defuzzification strategy is aimed at producing a non fuzzy control action that best represents the output fuzzy set resulting from fuzzy implication and composition. The most commonly used methods are the max. criterion, the mean of maximum and center of area.

a) The max. method:

The max. method produces the point at which the output fuzzy set reaches the maximum value. If more than one value exists, it takes the min. control action point on the universe of discourse.

b. The mean of maximums method (MOM):

The MOM method generates the control action which represents the mean value of all local control actions whose membership function reaches the maximum.

c) The center of area method (COA):

This method produces the control action which represents the center of the output fuzzy set. It is noted that when the MOM is used the FLC is similar to the multi level relay system, while the COA yields smooth results like the conventional PID controllers.

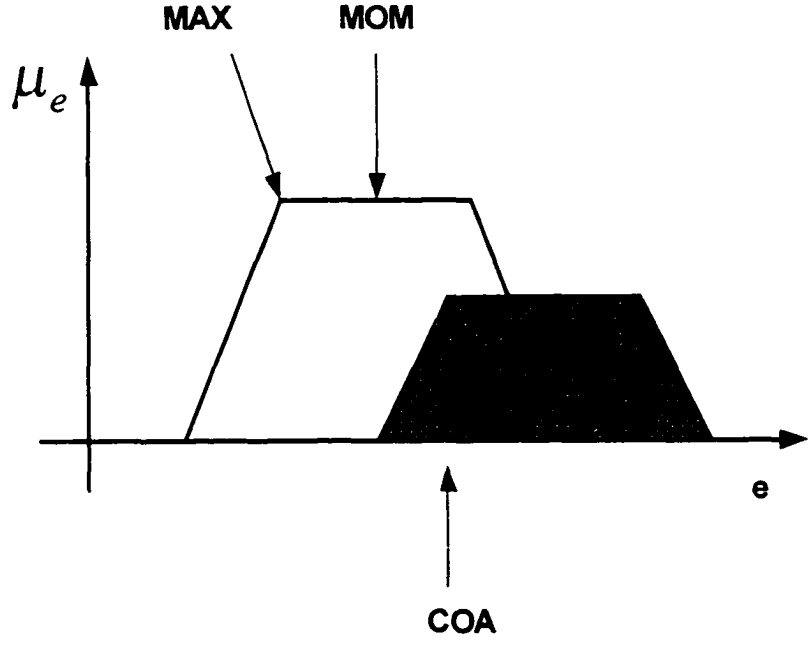


Figure 5.8 Defuzzification strategies

5.3.3 FLC Algorithm:

In the previous section, we considered the parameters of any FLC in detail. In this section, we give the algorithm used in implementing FLC with some comments on the scaling and mapping functions.

Without loss of generality we consider the PD FLC in which the controller inputs are the error (e) and error derivative (ce) [ce = e(k) - e(k-l)], and the controller output is the process input (u).

The control rules are in the form:

$$\text{If } e \text{ is } E_i \text{ and } ce \text{ is } CE_i \text{ then } u \text{ is } U_i$$

where E_i, CE_i, U_i are linguistic terms of error, change in error and process input respectively. Then, the scaling factors which scale the real universe of discourse into the normalized universe are Ge, Gce, Gu with:

$$e_n = F_m(e * Ge),$$
$$ce_n = F_m(ce * Gce); \qquad \text{input scaling}$$

where F_m is the mapping function which is usually linear or nonlinear logarithmic, to improve the control quality around the set point.

$$u = u_n * G_u; \qquad \text{Output scaling}$$

The selection of the scaling factors plays a very important role in the performance of the FLC; however, there is no systematic method to find the optimal scaling factors.

There are some guidelines which help the in the selection of the scaling factors such as the maximum error, maximum change in error and the maximum (available) control action and their maximum normalized values. In a final form, the FLC algorithm can be given as shown in Figure 5.9.

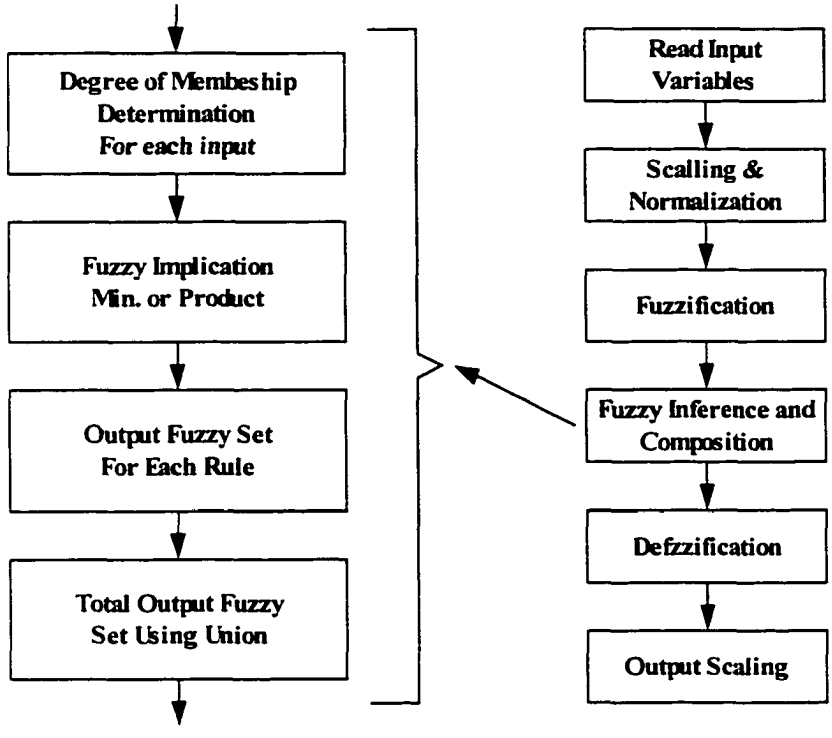


Figure 5.9 FLC Functional block diagram

5.4 Conclusion:

Having performed the previous studies in this chapter, the following main conclusions about fuzzy control are drawn:

1. Fuzzy control is capable of dealing with Systems without requiring detailed dynamic model by employing an approach which is close to the human decision making algorithm.
2. Fuzzy logic controllers are nonlinear controllers which involve some heuristic in their design, many of the design parameters depend on the characteristics of the process under control. However, there is no systematic method to find the optimal parameters rather than using simulation results.

3. Fuzzy logic control is not a replacement of classical control techniques; it requires the same kind of feedback loops required in classical control theory. However, fuzzy controllers are more robust than linear controllers and are capable of combining both crisp and linguistic data in the same frame work to find a suitable control strategy.

CHAPTER VI

Sliding Mode Fuzzy Control

Most of the fuzzy robot controllers with nonlinear multi-input multi-output (MIMO) systems are designed with a two-dimensional phase plane in mind. In this chapter, the performance and the robustness of this kind of robot controller, which arises from their property of driving the system into the so-called sliding mode (SM), is shown. This method will make the controlled system invariant to parameter fluctuations and disturbances. In addition, near the switching line the continuous distribution of the control values in the phase plane causes a behavior similar to that of a sliding mode controller (SMC) with a boundary layer (BL). Even in the presence of high model uncertainties, this gives assured tracking quality. Then, we introduce the boundary layer at the fuzzy controller to obtained further improvement. Furthermore, the stability of the closed-loop system can be obtain when using the principle of the SMC for the fuzzy controller. The choice of the scaling factors for the crisp inputs and outputs can be guided by the comparison of the fuzzy controller with the sliding mode controller and with the modified sliding mode controller, respectively.

Therefore, this chapter clarifies the details of sliding mode fuzzy control (SMFC). In section 6.1, a background on sliding mode fuzzy control is discussed. In section 6.2, we give a short review of the ordinary SMC with BL. In section 6.3, we describe the diagonal form SMFC and derive the similarities between the control law of a diagonal form SMFC and the control law of an SMC with BL which will describe the design of the

control law of an SMFC for the tracking control problem. Finally, we apply the design method of SMFC to MIMO robot manipulator.

6.1 Background of Sliding Mode Fuzzy Control

In the classical control theory, controllers are designed based on mathematical descriptions and system models. However, in the real world, as systems become more complex, it is more difficult to describe them mathematically and design model-based controllers to control these ill-defined systems. As an alternative to these model-based control schemes, fuzzy control (FC) research was initiated by Mamdani [70] based on the fuzzy set theory that Zadeh [52] proposed to enable people to formulate the qualitative linguistic characters apparent in our daily life. A comprehensive review of the classical design and implementation of the fuzzy logic controller can be found in the previous chapter. In this chapter, fuzzy control is combined sliding mode control. Several papers have been proposed on the relationship and combination of FC and SMC. Kim and Lee [71] proposed to design a fuzzy controller with the fuzzy sliding surface. Wu and Liu [72] formulated FC to become a class of SMC and developed a method to determine best values for parameters in FC rules by using sliding modes. Lin et al. [73] proposed a fuzzy sliding mode control scheme that improved SMC with the aid of FC.

6.2 Sliding Mode Control with Boundary Layer [74]

The remarkable property of the SMC is that the sliding mode occurs on the switching surface, and while in this mode, the system remains insensitive to parameter

uncertainties and external disturbances. In this section, we shall assume a dynamical system described by the nonlinear differential equation:

$$x^{(n)}(t) = f\left(x, \dot{x}, \ldots, x^{(n-1)}; t\right) + u(t) + d(t) \tag{6.1}$$

that can be expressed in a vector form as

$$x^{(n)}(t) = f(\underline{x}, t) + u(t) + d(t), \tag{6.2}$$

where $\underline{x}(t) = \left(x, \dot{x}, \ldots, x^{(n-1)}\right)^T$ is the state vector, $u(t)$ is a control variable and $d(t)$ is the disturbance. We shall use the continuous mode because the theory of VSS was developed predominantly for such continuous systems. The function $f(\underline{x}, t)$ is generally a nonlinear function of the state vector and time. It is assumed that the model $\hat{f}(\underline{x}, t)$ of this function (plant) is known with some uncertainty $\Delta f(\underline{x}, t)$:

$$f(\underline{x}, t) = \hat{f}(\underline{x}, t) + \Delta f(\underline{x}, t) \tag{6.3}$$

Let furthermore, Δf, d and $x_d^{(n)}(t)$ have the upper bounds with known values F, D and υ:

$$|\Delta f| \le F(\underline{x}, t); \quad |d| \le D(\underline{x}, t); \quad \left|x_d^{(n)}(t)\right| \le \upsilon(t). \tag{6.4}$$

In a typical control problem the aim of the control system is to track (especially to converge asymptotically to) a given desired state vector $\underline{x}_d^{(n)}(t) = \left(x_d, \dot{x}_d, \ldots, x_d^{(n-1)}\right)^T$. We shall denote the tracking error between the current and desired state vector by vector $\underline{e} = \left(e, \dot{e}, \ldots, e^{(n-1)}\right)^T$, where

$$\underline{e} = \underline{x}(t) - \underline{x}_d(t). \tag{6.5}$$

We shall also define the so-called *generalized error s* as a linear function of the coordinates of the error vector $\underline{e}$:

$$s(\underline{x},t)=\left(\frac{d}{dt}+\lambda\right)^{(n-1)} e; \quad \text{with } \lambda \geq 0. \tag{6.6}$$

The surface

$$s(\underline{x},t)=0, \tag{6.7}$$

will be called *the sliding surface* in the state space.

Starting from the initial conditions

$$\underline{e}(0)=0 \tag{6.8}$$

the tracking problem $\underline{x}=\underline{x}_d$ can be considered as the state vector $\underline{e}$ remaining on the sliding surface $s(\underline{x},t)=0$ for all $t \geq 0$. A sufficient condition for this behavior is to choose the control value so that

$$\frac{1}{2}\cdot\frac{d}{dt}\left(s^2(\underline{x},t)\right) \leq -\eta\cdot|s|; \quad \text{with } \eta \geq 0. \tag{6.9}$$

Considering $s^2(\underline{x},t)$ a Lyapunov function, it follows from equation (6.9) that the system controlled is stable. Looking at the phase plane we obtain: *The system is controlled in such a way that the state always moves towards the sliding surface.* The sign of the control value must change at the intersection of state trajectory $\underline{e}(t)$ and sliding surface. In this way, the trajectory is forced to move always towards the sliding surface as in Figure (6.1). A sliding mode along the sliding surface is thus obtained. By remaining in the sliding mode of equation (6.9), the system is invariant despite model uncertainties, parameter fluctuations and disturbances. However, sliding mode causes

high control activities which is an evident drawback for technical systems. Returning to equation (6.9), one obtains the conventional notation for sliding mode

$$s \cdot \dot{s} \leq -\eta \cdot |s| \quad \text{or alternatively} \quad \dot{s} \cdot \operatorname{sgn}(s) \leq -\eta. \tag{6.10}$$

In the following, without loss of universality, we focus on 2nd order systems. Hence, from equation (6.7) follows

$$s = \lambda e + \dot{e} \quad \text{and} \quad \dot{s} = \lambda \dot{e} + \ddot{e} = \lambda \dot{e} + \ddot{x} - \ddot{x}_d.$$

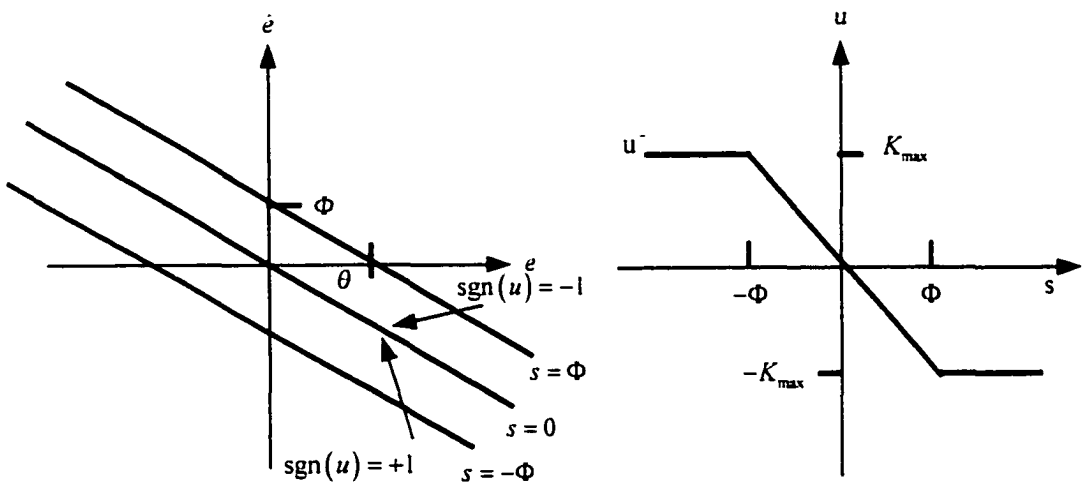


Figure 6.1 Sliding mode principle with boundary layer

From this and equation (6.2) follows

$$s \cdot \dot{s} = s \cdot \left(\lambda \dot{e} + f(\underline{x},t) + u + d - \ddot{x}_d\right) \leq -\eta \cdot |s|. \tag{6.11}$$

Rewriting this equation leads to

$$\left[f(\underline{x},t) + d + \lambda \dot{e} - \ddot{x}_d\right] \cdot \operatorname{sgn}(s) + u \cdot \operatorname{sgn}(s) \leq -\eta. \tag{6.12}$$

To achieve the sliding mode of equation (6.10) we choose u so that

$$u = \left(-\hat{f} - \lambda \dot{e}\right) - K(\underline{x},t) \cdot \operatorname{sgn}(s) \text{ with } K(\underline{x},t) > 0, \tag{6.13}$$

where $(-\dot{f}-\lambda\dot{e})$ is a compensation term and the 2nd term is the controller. With this, equation (6.12) can be written as

$$\left(f(\underline{x},t)-\hat{f}(\underline{x},t)d(t)-\ddot{x}_d\right)\cdot\mathrm{sgn}(s)-K(\underline{x},t)\le-\eta. \tag{6.14}$$

Installing the upper bounds of equation (6.4) in equation (6.14) one obtains finally

$$K(\underline{x},t)_{\max}\ge F+D+\upsilon+\eta. \tag{6.15}$$

To avoid drastic changes of the manipulated variable mentioned above, we substitute function $\mathrm{sgn}(s)$ by $sat\left(\frac{s}{\Phi}\right)$ in equation (6.13), where

$$sat(x)=\begin{cases} x & \text{if } |x|<1 \\ \mathrm{sgn}(x) & \text{if } |x|\ge 1 \end{cases}.$$

This substitution corresponds to the introduction of a boundary layer (BL) $|s|\le\Phi$ as in Figure (6.1). Thus, we have

$$u=-\hat{f}-\lambda\dot{e}-K(\underline{x},t)_{\max}\cdot sat\left(\frac{s}{\Phi}\right) \quad \text{where} \quad \Phi>0;\quad K(\underline{x},t)_{\max}>0. \tag{6.16}$$

From this follows with equation (6.11) and equation (6.14) the filter function

$$\dot{s}+K(\underline{x},t)\cdot\frac{s}{\Phi}=\Delta f+d-\ddot{x}_d \tag{6.17}$$

for unmodelled disturbances, model fluctuations and the acceleration of the desired values x_d for the input. The output s of the filter is the distance to the switching line. With the slope λ of the switching line $s = 0$ one obtains the guaranteed tracking precision

$$\theta=\frac{\Phi}{\lambda}. \tag{6.18}$$

The break frequency of filter (6.17) yields

$$\nu\Phi = \frac{K(\underline{x},t)_{\max}}{\Phi}. \tag{6.19}$$

On the other hand, even equation (6.7) is a filter with s as input and e as output. Hence, the break frequency λ of this filter should be, like ν_Φ, small compared to unmodelled frequencies ν_{su} :

$$\lambda \ll \nu_{su}. \tag{6.20}$$

Furthermore, ν_Φ should be less than or equal to the largest exceptable λ. From equation (6.19) we obtain the balance condition

$$\nu\Phi = \frac{K(\underline{x},t)_{\max}}{\Phi} = \lambda, \tag{6.21}$$

i.e., critical damping. Finally, the design rule for λ with regard to sample rate ν_{sample} and time constant t_{plant} of the plant [Slotine 85] can be denoted as

$$\lambda < \frac{\nu_{sample}}{2\cdot\left(1+t_{plant}\cdot\nu_{sample}\right)}. \tag{6.22}$$

6.3 Sliding Mode Fuzzy Control

Using the theory of sliding mode control (SMC) with boundary layer (BL) and comparing it with a fuzzy control whose rules have been derived from the phase plane as it appears in [75–86], leads to a new method called sliding mode fuzzy control (SMFC), which is an extension of sliding mode with boundary layer. Rewriting equation (6.13) to be used in this new method, results in the following equation

$$u = -F_{FC}(e,\dot{e},\lambda)\cdot\mathrm{sgn}(s), \tag{6.23}$$

where u is the control output and F_{Fz} is a non-linear non-continuous and positive function of $e, \dot{e}$ and λ, the error, change in error and the frequency, respectively. The rules are, in general, so conditioned that above the switching line a negative control output is generated and a positive one below it, similar to the SMC. Figure (6.2) shows an example[92] where close to the sliding surface (switching line) control outputs are smaller than at a larger distance. So, five steps should be followed to construct the SMFC before designing its rule modification:

1. Obtain the universe of discourse by normalizing the error vector e into e_N
2. Predefine membership functions of the components of e_N through fuzzification of e_N to μ_{e_N}
3. Predefine membership functions of the normalized control output u_N and fuzzy rules through calculation of the fuzzy output μ_u
4. Defuzzification of μ_u onto a normalized u_N
5. Denormalizing of u_N onto a physical control output u.

Normalization as mentioned in[68], is a state transformation. The actual control processing of fuzzy rules takes place within the normalized phase plane. The switching line $s = 0$ has to be transformed as follows:

Within the non-normalized phase plane we have $\lambda \cdot e + \dot{e} = 0$. In the normalized plane, we obtain $\lambda_N \cdot e_N + \dot{e}_N = 0$. If we summarize the relationship of the parameters e_N and $\dot{e}_N$ as

$$e_N = e \cdot N_e; \quad \dot{e}_N = \dot{e} \cdot N_{\dot{e}}; \quad N_e, N_{\dot{e}} - \text{normalization factors}$$

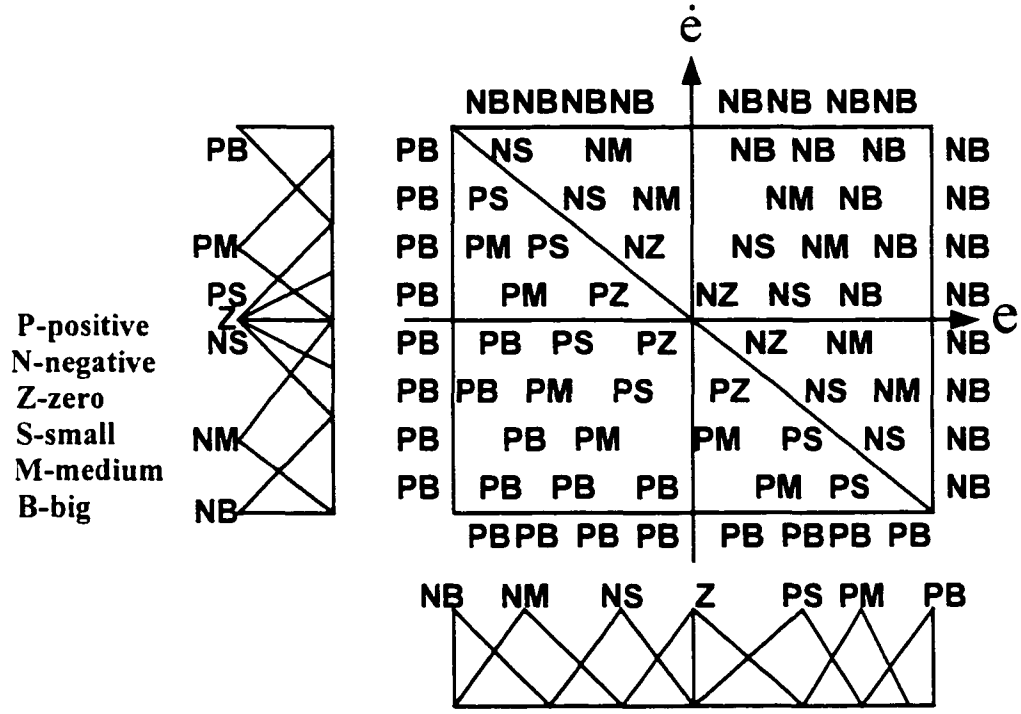


Figure 6.2 Rules in the normalized phase plane

λ will be formulated as

$$\lambda = \lambda_N \cdot \frac{N_e}{N_{\dot{e}}}. \tag{6.24}$$

From design rule (6.20) for λ one obtains the following rule

$$\lambda_N \cdot \frac{N_e}{N_{\dot{e}}} << v_{nu}. \tag{6.25}$$

Hence, from equation (6.25) we have obtained the break frequency above which all frequencies of the unmodelled dynamics and disturbances are located. After choosing the upper bound of $\lambda_N \cdot N_e / N_{\dot{e}}$, we design the rules with respect to the normalized phase plane as follows:

Rule 1: u_N should be negative above the switching line and positive below it

Rule 2: as the distance grows between the actual state and the switching line, $|u_N|$ should increase

Rule 3: as the distance grows between the actual state and the line perpendicular to the switching line, $|u_N|$ should increase, for the following reasons:

- to avoid the discontinuities at the boundaries of the phase plane
- speed up the arrival of the central domain of the phase plane

Rule 4: the maximum values $|u_N|_{max}$ should be used to cover the normalized states $e_N, \dot{e}_N$ that fall out of the phase plane with the respective sign of u_N.

The difference between the SMFC of equation (6.23) and the SMC of equation (6.13) is the compensation part. If there is no sufficient model of the nonlinear part $f(\underline{x},t)$ the upper bound of $f(\underline{x},t)$ has to be modified:

$$F = \left| f(\underline{x},t)_{max} \right|.$$

Thus, we obtain

$$F_{Fz}|_{max} \geq F + D + \upsilon + \eta \tag{6.26}$$

for the maximum of F_{Fz}. From this, the denormalization factor N_u can be calculated from the maximum F_{Fz_R} of in the normalized phase plane. $F_{Fz_R}|_{max}$, is obtained by *defuzzification* $N_{Fz_N}|_{max} = \left| Defuzz\left(\mu_{u=u_{max}}\right) \right|$. From this and $F_{Fz}|_{max} = R_u \cdot F_{Fz_N}|_{max}$ the denormalization factor N_u follows directly:

$$N_u = \frac{F_{Fz}|_{max}}{F_{Fz_N}|_{max}}.$$

This will lead to a partial compensation for the SMFC in the form

$$u = -\lambda \dot{e} - F_{Fz}(e, \dot{e}, \lambda) \cdot \mathrm{sgn}(s) \tag{6.27}$$

which is a conventional compensation strategy and fuzzy control in a hybrid version. Also, from equation (6.27), one automatically obtains a type of boundary layer (BL) due to the interpolation property of the fuzzy algorithm. However, this BL depends strongly on the number and size of the membership functions used in the fuzzy rules. Then, the SMFC generates a piecewise linear function $u = f(s)$, as in Figure (6.3). Yet, with an increasing number of membership functions $u = f(s)$ becomes more and more linear[94]. Hence, similar to (6.17) we obtain the following filter function for the *i*-th segment

$$\dot{s} + \frac{k_i}{\phi_i} \cdot s = u_i \cdot \mathrm{sgn}(s) + f(\underline{x}, t) + d - \ddot{x}_d \tag{6.28}$$

with

$$u_i = -\sum_{v=1}^{i-1} k_v + \frac{k_i \cdot \sum_{v=1}^{i-1} \phi_v}{\phi_v}$$

$$k_v, \phi_v > 0; \quad i = 1, 2, \ldots, n; \quad \sum_{v=1}^{0} [\ldots] = 0; \quad n - \text{number of segments.}$$

Equation (6.28) is a state dependent filter with different break frequencies[85] k_i/ϕ_i. With the current state at a larger distance from the switching line, the approach velocity to the line can be larger. Since at a large distance any unmodelled frequencies are not able to cause a change of sign of the manipulated variable, condition[87]

$$\frac{k_i}{\phi_i} \le \lambda \tag{6.29}$$

has to be fulfilled only for *i*=1. The tracking quality is guaranteed by the maximum values $F_{Fz}|_{max}$ and Φ_{max} as long as

$$\frac{F_{Fz}|_{\max}}{\Phi_{\max}} \le \lambda. \tag{6.30}$$

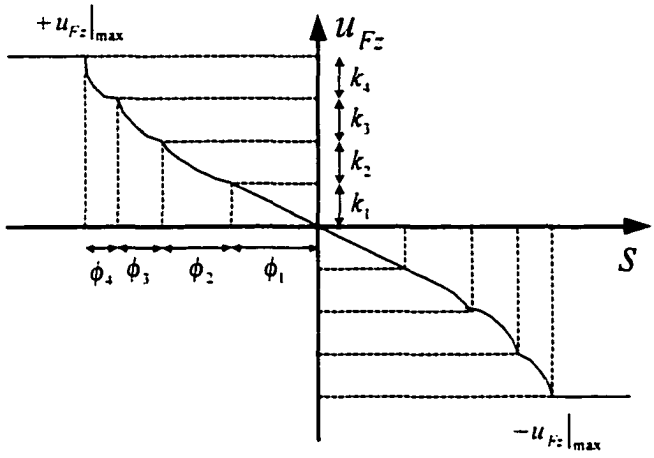


Figure 6.3 Nonlinear operating line

However, if we have a special plant with upper bounds

$$F_k + D_k + \upsilon_k \le F_{\max} + D_{\max} + \upsilon_{\max},$$

we are in the position to define a $F_{Fzk} \le F_{Fz}|_{\max}$, so that

$$F_{Fzk} \ge F_k + D_k + \upsilon_k + \eta.$$

From this we obtain the corresponding Φ_k for plant P_k. Furthermore, using the SMFC for different systems P_l with

$$F_{Fz}|_{\max} \ge F_{Fzl} \ge F_l + D_l + \upsilon_l + \eta,$$

one obtains for each system P_l a special domain Φ_l.

A sliding mode fuzzy control is faster and more robust with respect to changes of system parameters than a sliding mode control with boundary layer for the following reasons:

The frequency λ is obligatory for all systems P_i considered. From equation (6.29) one obtains a guaranteed Φ_{max} for all systems P_i. In addition to this, for each system P_i a $\Phi_i \leq \Phi_{max}$ can be ensured if the upper bounds F_i, D_i and υ_i are predefined. Furthermore, the manipulated variable τ varies with the distance of the state vector from the line perpendicular to the switching line. This achieves a smooth transition behavior of the control output at the boundaries of the phase plane. Finally, the variation of the slope of the operating line $u = f(s)$ permits a fast approach of the state vector to the switching line combined with smooth behavior close to the line.

On the other hand, a sliding mode control with boundary layer is designed for only one system and its upper bounds, and within the BL the approach velocity of the state vector to the switching line is constant. Therefore, disturbances in the control loop and fast changes of the desired values require a longer settling time than under sliding mode fuzzy control.

Based on equation (6.30), domains $|s| \leq \Phi_i$ for systems P_i have been formulated. If equation (6.29) is not satisfied, the system becomes more sensitive to unmodelled frequencies. At sufficient distance Φ from the switching line, this fact does not affect the system's behavior that much since no sign change of the manipulated variable u can be caused. However, in the neighborhood of the switching line undesired fast sign changes of u can occur especially close to the boundary of the normalized phase plane. It is therefore useful to build a boundary layer (BL) also for the sliding mode fuzzy control (SMFC). According to equation (6.16) and (6.27), the modified SMFC with BL has the property:

$$u = -\lambda \cdot \dot{e} - F_{FZ}(e, \dot{e}, \lambda) \cdot sat(s/\Phi). \tag{6.31}$$

This gives the advantages:

1. Crisp changes of u can be avoided at the boundary of the $(e_n, \dot{e}_n)$-plane.
2. Condition (6.29) is fulfilled if Φ is chosen as follows: Let $|u_n|_{max}$ be the maximum value of the crisp (defuzzified) value of u_n which will occur inside $(e_n, \dot{e}_n)$-plane and close to the switching line. Then Φ has to be $\Phi = |u_n|_{max}/\lambda$.
3. With this design one is able to reduce the number of rules, i.e. one can choose a minimum set of rules generating; however, a relatively rough gradation of u_n. This can be remedied by a BL, serving as interpolation agent.

The SMFC with BL provides adaptive tracking quality even under changing process parameters. It achieves this both inside and outside the layer together with a filtering of unmodelled frequencies. A well-designed SMFC with BL gives a smoother control than the SMC with BL. Obviously, once this framework has been determined, the number and shape of the membership functions of the controller must be optimized.

6.4 Design of SMFC for Robot Manipulator

Let the basic equation describing the motion of a robot arm be

$$\tau = M(q) \cdot \ddot{q} + N(q, \dot{q}), \tag{6.38}$$

where

q is a $(k \times 1)$ position vector,
$\dot{q}$ is a $(k \times 1)$ velocity vector,
$\ddot{q}$ is a $(k \times 1)$ acceleration vector,
$M(q)$ is a $(k \times k)$ matrix of inertia (invertible),
$N(q,\dot{q})$ is a $(k \times 1)$ vector of damping, centrifugal, coriolis, gravitational forces,
τ is a $(k \times 1)$ vector of torques.

Furthermore, let the system (6.38) have k degrees of freedom (d.o.f.). Normally, $k = 6$, so that the number of d.o.f. in the Cartesian space R_x^k is equal to the number of d.o.f. in the joint space R_q^k. The control problem is to follow a given trajectory $\theta_d(t)$ and to produce a torque vector τ such that the tracking error approaches 0 as $t \to \infty$. The control design steps are as follows:

1. Introduction of the overall control law,
2. Choice of the fuzzy values for the normalized controller inputs s_{iN} and the controller outputs τ_{FziN},
3. Design of the fuzzy rules for each link,
4. Choice of the slopes λ_i of the sliding lines $s_i = 0$,
5. Choice of the normalization factors N_{e_i} and $N_{\dot{e}_i}$ for e_{q_i} and $\dot{e}_{q_i}$,
6. Choice of the upper bounds from Section 6.4,
7. Design $|F_{iFz}|_{\max}$.

For a two-link robot manipulator, the values $F_{1Fz}|_{\max}$ and $F_{2Fz}|_{\max}$ could be calculated when we apply the equation of motion as follow:

$$M(q)\cdot\ddot{q}=-N(q,\dot{q})+\tau. \tag{6.39}$$

In this equation

$$M(q)=\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}=$$

$$\begin{bmatrix} (m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos(\theta_2) & m_2l_2^2+m_2l_1l_2\cos(\theta_2) \\ m_2l_2^2+m_2l_1l_2\cos(\theta_2) & m_2l_2^2 \end{bmatrix}, \tag{6.40}$$

and

$$N(q,\dot{q})=\begin{pmatrix} n_1 \\ n_2 \end{pmatrix}=$$

$$\begin{pmatrix} m_2l_1l_2\left(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2\right)\sin(\theta_2)+(m_1+m_2)gl_1\cos(\theta_2)+m_2gl_2\cos(\theta_1+\theta_2)+K\theta_1\left(\dot{\theta}_1\right) \\ m_2l_1l_2\sin(\theta_2)+m_2gl_2\cos(\theta_1+\theta_2)+K\theta_2\left(\dot{\theta}_2\right) \end{pmatrix}, \tag{6-41}$$

where $K\theta_1$ and $K\theta_2$ are the damping coefficients for the coordinates q_1 and q_2, respectively.

Calculation of the maximum values $F_{1Fz}|_{\max}$ and $F_{2Fz}|_{\max}$

The inverse of $M(q)$ in the control law (6.38) is given as

$$M^{-1}(q)=\begin{pmatrix} m_{11}^{-1} & m_{12}^{-1} \\ m_{21}^{-1} & m_{22}^{-1} \end{pmatrix}=$$

$$\begin{pmatrix} m_2l_2^2/D & -m_2l_2^2+m_2l_1l_2\cos(\theta_2)/D \\ -m_2l_2^2+m_2l_1l_2\cos(\theta_2)/D & (m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos(\theta_2)/D \end{pmatrix}. \tag{6.42}$$

Then we have

$$M^{-1}(q)\Delta N=\begin{pmatrix} \Delta f_1 \\ \Delta f_2 \end{pmatrix}=\begin{pmatrix} m_{11}^{-1}\cdot\Delta n_1 & m_{12}^{-1}\Delta\cdot n_2 \\ m_{21}^{-1}\cdot\Delta n_1 & m_{22}^{-1}\cdot\Delta n_2 \end{pmatrix}, \tag{6.43}$$

where $\Delta n_1=n_{d1}-n_1$ and $\Delta n_2=n_{d2}-n_2$. We obtain the $\left(m_j^{-1}\right)^T\cdot(m_i)$ for $i\neq j$ as follows:

$$m_{jl} = \left(m_{11}^{-1} \cdot m_{12} + m_{21}^{-1} \cdot m_{22}\right),$$
$$m_{lj} = \left(m_{12}^{-1} \cdot m_{11} + m_{22}^{-1} \cdot m_{21}\right). \tag{6.44}$$

For the $\left(m_i^{-1}\right)^T \cdot \left(m_1\right)$ we obtain

$$m_{il1} = \left(m_{11}^{-1} \cdot m_{11} + m_{21}^{-1} \cdot m_{21}\right),$$
$$m_{il2} = \left(m_{12}^{-1} \cdot m_{12} + m_{22}^{-1} \cdot m_{22}\right). \tag{6.45}$$

From (6.45) we establish $\beta_i^{\min}$ and $\beta_i^{\max}$:

$$\beta_1^{\min} \le \left(m_{11}^{-1} \cdot m_{11} + m_{21}^{-1} \cdot m_{21}\right) \le \beta_1^{\max},$$
$$\beta_2^{\min} \le \left(m_{12}^{-1} \cdot m_{12} + m_{22}^{-1} \cdot m_{22}\right) \le \beta_2^{\max}. \tag{6.46}$$

Then we fix G_i as

$$G_1 = \frac{1}{\sqrt{\beta_1^{\min} \cdot \beta_1^{\max}}}; \quad G_2 = \frac{1}{\sqrt{\beta_2^{\min} \cdot \beta_2^{\max}}}, \tag{6.47}$$

and β_i as

$$\beta_1 = \sqrt{\frac{\beta_1^{\max}}{\beta_1^{\min}}}; \quad \beta_2 = \sqrt{\frac{\beta_2^{\max}}{\beta_2^{\min}}}. \tag{6.48}$$

Then, the control torque τ can easily be obtained as

$$\tau = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \cdot \begin{pmatrix} \tilde{\tau}_1 \\ \tilde{\tau}_2 \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}, \tag{6.49}$$

where

$$\tilde{\tau}_1 = G_1 \cdot \left(\tau_1 - F_{1Fz}\big|_{\max} \cdot \operatorname{sgn}\left(s_1\right)\right)$$
$$\tilde{\tau}_2 = G_2 \cdot \left(\tau_2 - F_{2Fz}\big|_{\max} \cdot \operatorname{sgn}\left(s_2\right)\right)$$

and where

$$\tau_1 = \ddot{\theta}_{1d} - \lambda \cdot \dot{e}_{\theta 1},$$
$$\tau_2 = \ddot{\theta}_{2d} - \lambda \cdot \dot{e}_{\theta 2},$$
$$s_1 = \lambda \cdot e_{\theta 1} - \dot{e}_{\theta 1},$$
$$s_2 = \lambda \cdot e_{\theta 2} - \dot{e}_{\theta 2}.$$

In addition we have to determine the upper bounds F_i, U_i, M_{ij}, and $Q_i\,(i, j = 1,2)$:

$$|\Delta f_1| \le F_1; \quad |\Delta f_2| \le F_2;$$
$$|\tau_1| \le U_1; \quad |\tau_2| \le U_2.$$

From (6.45) we obtain

$$\left|m_{11}^{-1} \cdot m_{11} + m_{21}^{-1} \cdot m_{21}\right| < M_{12},$$
$$\left|m_{12}^{-1} \cdot m_{12} + m_{22}^{-1} \cdot m_{22}\right| < M_{21},$$

and finally

$$\ddot{\theta}_1^d < Q_1,$$
$$\ddot{\theta}_2^d < Q_2.$$

Then, we obtain the maximum values $F_{1Fz}|_{\max}$ and $F_{2Fz}|_{\max}$ as

$$F_{1Fz}|_{\max} > \beta_1 \cdot \left(F_1 + \left(1 - \beta_1^{-1}\right) \cdot T_1 + \sum_{j \ne i} G_j \cdot B_{ij} \cdot Q_j + \eta_1 \right),$$
$$F_{2Fz}|_{\max} > \beta_2 \cdot \left(F_2 + \left(1 - \beta_2^{-1}\right) \cdot T_2 + \sum_{j \ne i} G_j \cdot B_{ij} \cdot Q_j + \eta_2 \right).$$

CHAPTER VII

Simulation and Discussion of Results

In Chapter Two, the dynamic model of a two-link manipulator was discussed. Chapter Three, Four, and Six presented the derivation and implementation of PID Controller, Sliding Mode Controller, and Sliding Mode Fuzzy Controller for the following two-link manipulator. This chapter gives an overview of the simulation and discussion of results and shows the selection of the values for the controllers' parameters.

7.1 Robot Simulation and Selection of Controllers' parameters

The purpose of a robotic simulation is to mimic an actual manipulator and to show the effectiveness of each controller. In order for the simulation to yield results as close to the real situation as possible, two measures must be taken. The first one relates to the accuracy of modeling. Since the actual model of a manipulator can not generally be obtained for use in the design of the controller, that luxury will not be utilized in the simulation. Therefore, a more precise model will be used in the integration scheme than the one used in the controller, as was discussed in Chapter Two. The second measure comes from the fact that when the control torques are applied to the joints of a real manipulator, the resulting movement is continuous even though the control torques are calculated at discrete intervals.

In the following subsections we will show the selection of the robot manipulator parameters. In addition, we will work with each controller and show the selection of its parameters.

7.1.1 Selection of the Robot Manipulator Parameters

In order to use the control torque equation in a simulation of the two-link manipulator examined in Chapter Two, parameter values for the estimated and exact manipulator models must be assumed. The limits placed on the maximum velocity and acceleration of both joints are 1 radian/second ≈ 57.3 degrees/second and 1 radian/second^2 ≈ 57.3 degrees/second^2. The parameter values chosen are shown in Table (7.1), where m_i are the masses, l_i are the link lengths parameters for links one and two.

Parameter	Exact	Estimated	Maximum error
$m_1(kg)$	3	2.5	0.88
$m_2(kg)$	2	2.5	0.76
$l_1(m)$	0.375	0.375	0
$l_2(m)$	0.25	0.25	0

Table 7.1 Manipulator Parameters

The exact manipulator model will be used in the robot arm dynamics while the estimated model will be used in the controller.

7.1.2 PID Computed Torque Controller

To have a robust and more stable controller, we first work in the simulation for the PID Computed Torque Controller which was the first controller that we discussed in Chapter Three. The selected parameters for this controller are shown in Table (7.2). Those parameters were chosen after several runs of the simulation program. In sense, we

try reducing the position and velocity error as close to zero as possible in order to have a more precise and robust controller.

PID Gains	Gain Values During Simulation Tests				
	Test 1	Test 2	Test 3	Test 4	Test 5
k_p	100	150	200	250	300
k_d	20	30	40	50	60
k_i	500	750	1000	1250	1500

Table 7.2 PID Gains Values

7.1.2 Sliding Mode Controller

The only parameters that are unknown for the sliding mode controller at this point are λ_1, λ_2, n_1, n_2, k_1 and k_2. These parameters represent the slope of the sliding surfaces, the boundary layer half-widths, and the attractiveness of the sliding surfaces. The values for these unknowns must be found by trial and error when the controller is being used. The constants that have the greatest effect on the performance of the controller are the slopes and boundary layer widths. For this reason, the decisions to assign these values are more important than the others. To aid in the decision making process, a study of the performance of the controller, while varying the values of the parameters, is conducted.

Testing of the performance index can be used to choose the optimal value for the sliding surface slope and boundary layer width of each joint. Before this can be done, a control bandwidth must be chosen. The factors to consider in this decision are accuracy and implementability. The higher the control bandwidth, the more accurate the controller is. After testing, the highest control bandwidth that can reasonably be used is 400 Hz for

joints one and two. This will therefore be used to aid in the decision of selecting the parameter values. For joint one, the sliding surface slope is 250 and a boundary layer width is 0.025. For joint two, the sliding surface slope is 250 and a boundary layer width is 0.030. These are the parameter values that will be used in all subsequent simulations including the selection of remaining unknown parameters.

The last parameter values to be chosen are k_1 and k_2, the attractiveness parameters of the sliding surfaces. To aid in this decision a test was made of the performance index versus the attractiveness parameter for joints one and two. There exists a peak in the performance index for joint one at an attractiveness value of about five. Once past that peak, however, the performance index appears to drop off exponentially. Due to this decay, there is not an appreciable decrease in the performance index past an attractiveness value equal to 120. Chattering in joint one was not seen at all during the analysis. Therefore, the chattering boundary is assumed to lie somewhere above 200. Using these observations, the attractiveness parameter for joint one is chosen as 120. The test of the performance index for joint two is very different from joint one. There is a minimum found in the performance index at an attractiveness value of 70 while the chattering boundary was located at an attractiveness value about equal to 100. Consequently, the attractiveness parameter is chosen as 70 for joint two. These parameter values, along with the values found earlier, are shown in the parameter summary table below and are used in the simulation program to yield the results discussed in the following sections.

Parameter	Joint One	Joint Two
Sliding Surface Slope, λ	250.0	250.0
Boundary Layer Width, n	0.025	0.030
Attractivness, k	120.0	70.0

Table 7.3 Sliding Mode Controller Parameters

7.1.3 Sliding Mode Fuzzy Controller

In order to find the parameters for the sliding mode fuzzy controller of a two-link robot, one must follow the steps described in chapter six. Therefore, the following results were obtained.

Choice of upper bounds and design $|K_{iFz}|_{\max}$: Now we determine the bounds of the right hand side of (6.53). From that the maximum values of K_{1Fz} and K_{2Fz} that suffice for a stable motion of the robot within the fuzzy regions are determined by the restrictions of position and velocity profile. Table (7.4) shows the calculated values $m_{u1}, m_{u2}, m_{jl}, m_{ij}, \Delta f_1, \Delta f_2$ of (6.47-6.50) for the lower and upper bounds of the parameter estimates and the lower and upper restrictions of q_i and $\dot{q}_i$ $(i = 1,2)$.

From these values

$$|m_{u1}|_{\max} = 0.9012, |m_{u1}|_{\min} = 0.456, |m_{u2}|_{\max} = 0.9012, |m_{u2}|_{\min} = 0.456,$$
$$|m_{jl}|_{\max} = 2, |m_{ij}|_{\max} = 5, |\Delta f_1|_{\max} = 64, |\Delta f_2|_{\max} = 64,$$

so that

$$\beta_1^{\min} = |m_{u1}|_{\min} = 0.456, \beta_1^{\max} = |m_{u1}|_{\max} = 0.9012,$$
$$\beta_1^{\min} = |m_{u1}|_{\min} = 0.456, \beta_1^{\max} = |m_{u1}|_{\max} = 0.9012.$$

$\dot{q}_1$	-17.5	-17.5	-10	0	0	0	+10	+17.5	+17.5
$\dot{q}_2$	-17.5	-17.5	-10	0	0	0	+10	+17.5	+17.5
$m_{i1\max}$	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012
$m_{i1\min}$	0.456	0.456	0.456	0.456	0.456	0.456	0.456	0.456	0.456
$m_{i2\max}$	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012	0.9012
$m_{i2\min}$	0.456	0.456	0.456	0.456	0.456	0.456	0.456	0.456	0.456
$m_{j\max}$	2	2	0	2	2	2	0	2	2
$m_{j\min}$	-2	-2	-2	0	0	-2	-2	0	-2
$m_{ij\max}$	5	5	5	5	5	5	5	5	5
$m_{ij\min}$	-5	-5	-5	-5	-5	-5	-5	-5	-5
$\Delta f_{1\max}$	64	32	58	32	5	29	5	-3	13
$\Delta f_{1\min}$	-3	3	-5	-32	-5	-29	-52	-29	-52
$\Delta f_{2\max}$	64	29	-5	43	48	-5	64	29	-5
$\Delta f_{2\min}$	5	-29	-58	1	-48	-55	5	-29	-58

Table 7.3 Values of $m_{i1}, m_{i2}, m_j, m_{ij}, \Delta f_1, \Delta f_2$

From β_1 and β_2 we obtain

$$G_1 = G_2 = 1.56,$$
$$\beta_1 = \beta_2 = 1.406.$$

Further, from last part of section 6.5

$$F_1 = F_2 = 64,$$
$$M_{12} = 2, M_{21} = 5.$$

Then, choose furthermore

$$\ddot{q}_1^d = \ddot{q}_1^d = 0,$$
$$\lambda = 250,$$
$$\eta_1 = \eta_2 = 1,$$
$$\left|\dot{e}_{q1}\right|_{\max} = \left|\dot{e}_{q2}\right|_{\max} = 0.9012.$$

Then we have

$$\hat{U}_1 = \hat{U}_2 = 10.$$

Finally, we choose

$$Q_1 = Q_2 = 1.$$

With that we obtain from

$$K_{1Fz}|_{\max} > 1.406\left(64 + (1 - 0.711)\cdot 10 + 1.56\cdot 2\cdot 1 + 1\right) = 150.4,$$
$$K_{2Fz}|_{\max} > 1.406\left(64 + (1 - 0.711)\cdot 10 + 1.56\cdot 5\cdot 1 + 1\right) = 157.$$

Again, it has to be emphasized here that the transfer characteristics of the pure SMFCs for links 1 and 2 are chosen to be identical.

7.2 Simulation and Discussion of Results

7.2.1 A Simple Trajectory

The trajectory that was used during the performance study in the previous work for reference[28] was a "simple trajectory" that required both joints to move smoothly from 15 to 75 degrees in three seconds. The desired velocity and acceleration were obtained by taking the first and second derivatives of the desired position with respect to time. Because of its simplicity, we ignore this testing in order to challenge all three types of controllers with a difficult trajectory.

7.2.2 A Difficult Trajectory

This section shows that the manipulator joints, when using the control parameters for each of the controllers, can follow a more complicated trajectory. The "difficult trajectory" that is used has five changes in direction that require the manipulator joints to

stop and rotate the opposite way. The desired joint positions and velocities are shown as Figures (7.1), (7.5), and the actual position and velocity are shown as Figures (7.2), (7.3), (7.4), (7.6), (7.7) and (7.8) for PIDC, SMC, and SMFC respectively. By requiring both joints to follow the same path; the interactions between them are magnified. This can be understood by imagining that if joint one changes direction, the tendency is for joint two to whip back and rotate in the reverse direction.

The equations for the desired position, velocity and acceleration are shown below as equations (7.1) to (7.3). The desired velocity and acceleration are obtained by taking the first and second derivatives of the desired position with respect to time.

$$\theta_d(t) = 15 + 5\left(\cos(t) + \cos(3t)\right) \tag{7.1}$$

$$\dot{\theta}_d(t) = -5\left(\sin(t) + 3\sin(3t)\right) \tag{7.2}$$

$$\ddot{\theta}_d(t) = -5\left(\cos(t) + 9\cos(3t)\right) \tag{7.3}$$

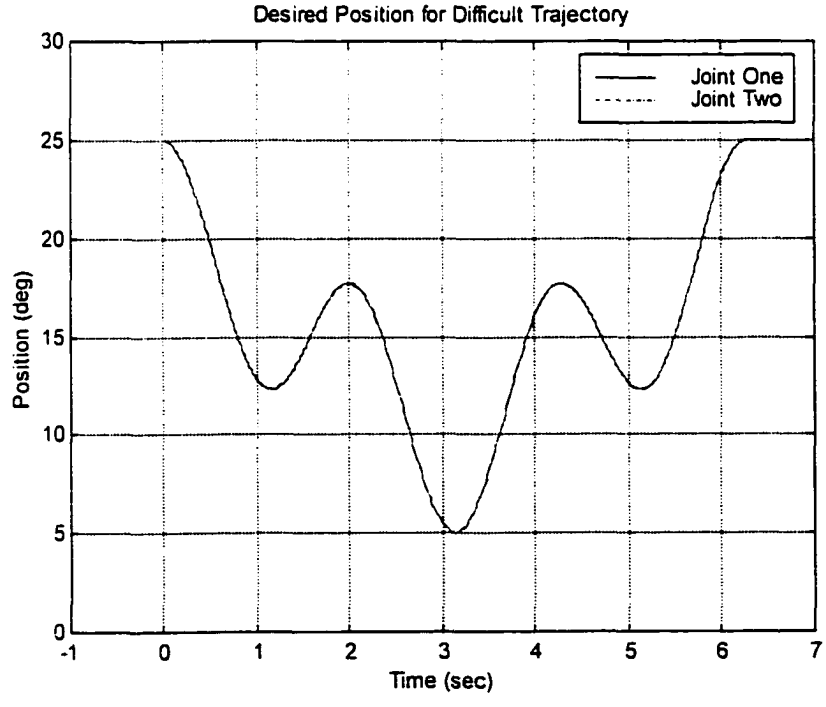


Figure 7.1 Desired Position for Difficult Trajectory

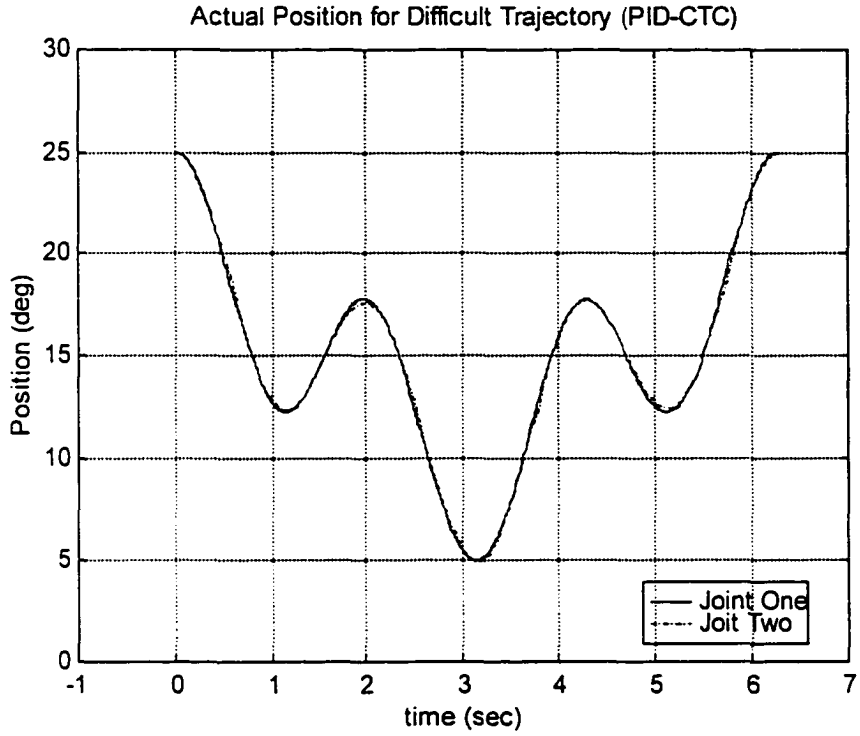


Figure 7.2 Actual Position for PID

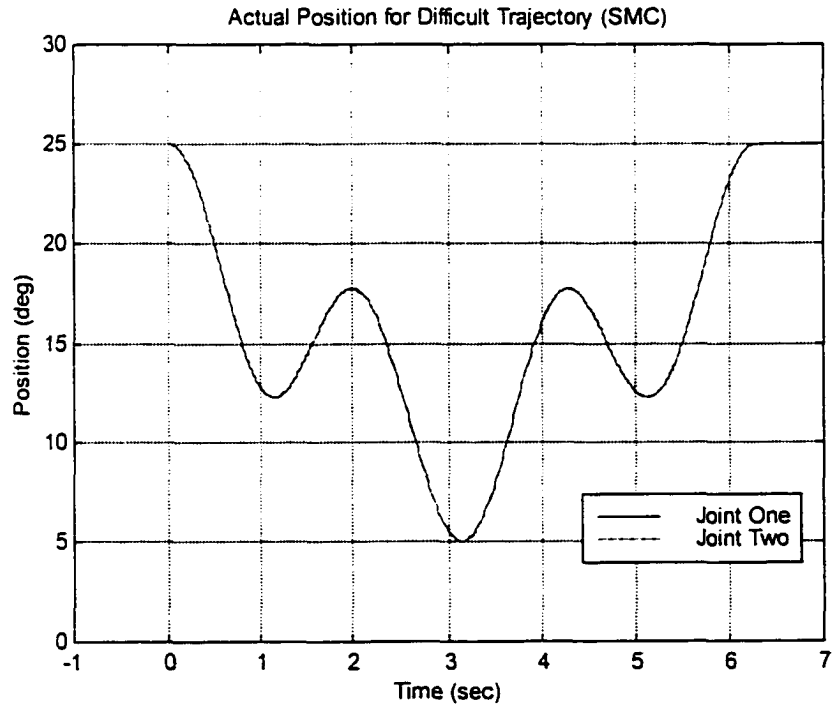


Figure 7.3 Actual Position for SMC

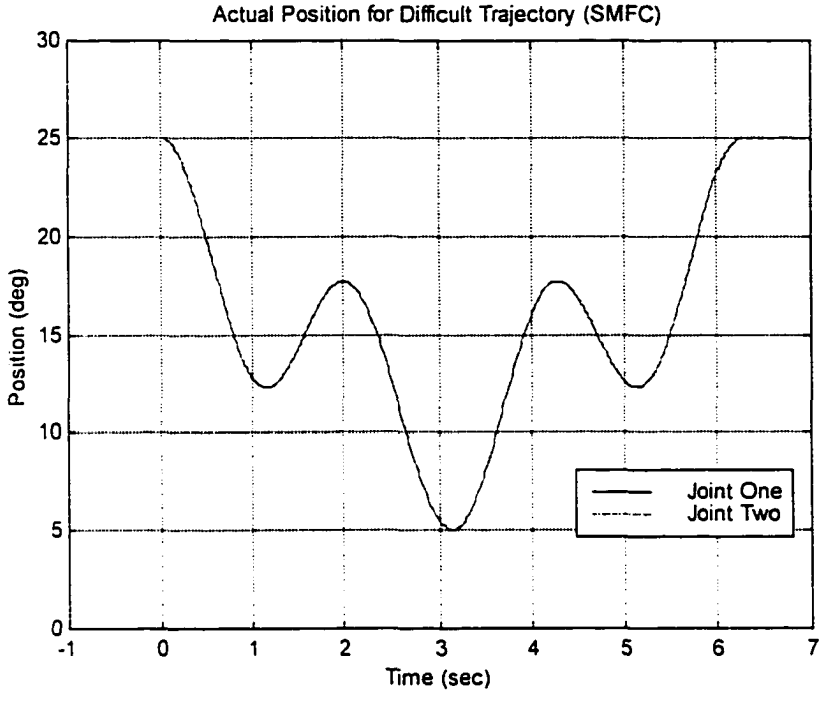


Figure 7.4 Actual Position for SMFC

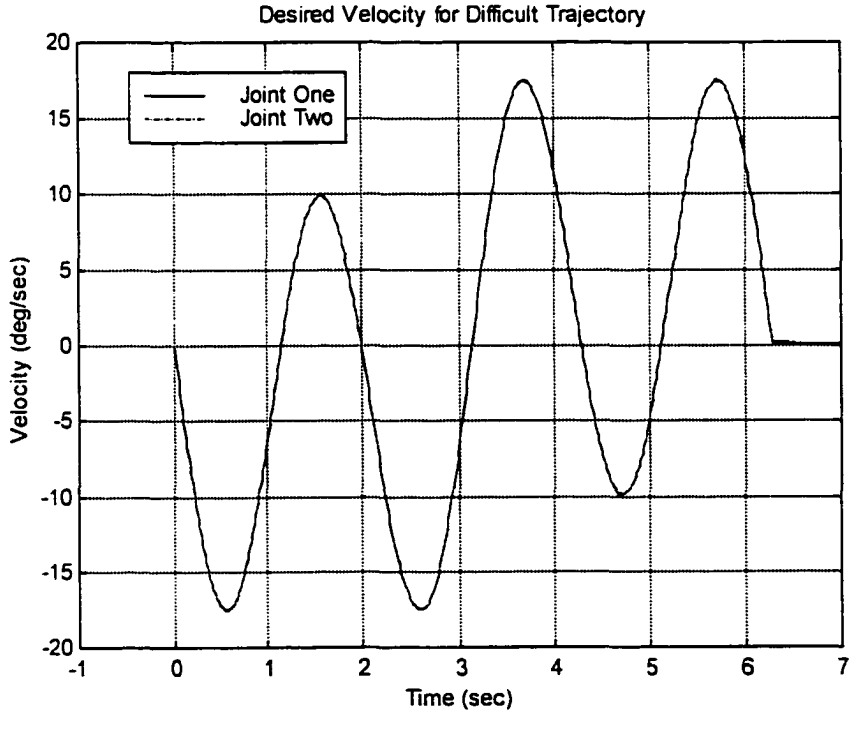


Figure 7.5 Desired Velocity for Difficult Trajectory

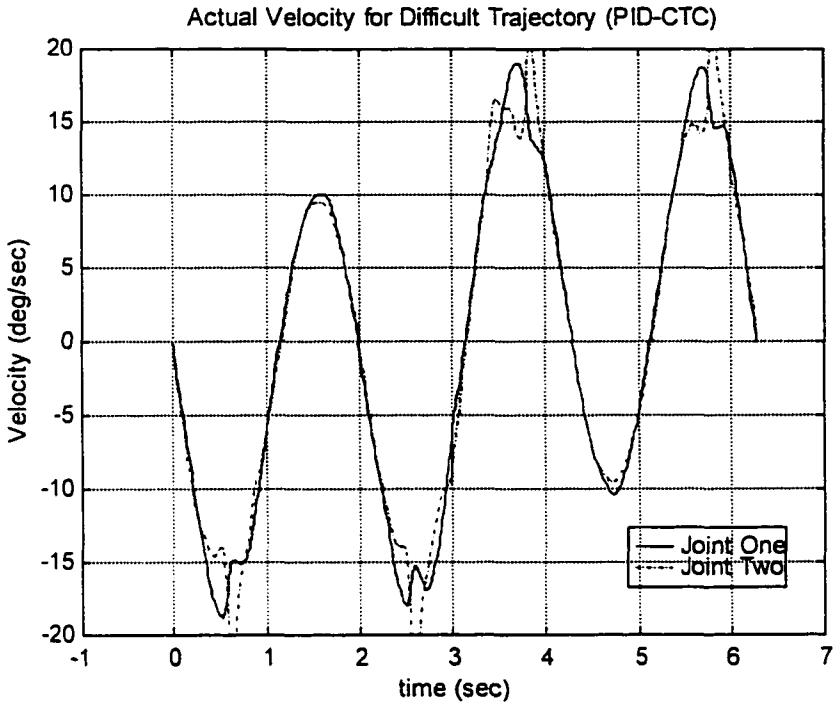


Figure 7.6 Actual Velocity for PIDC

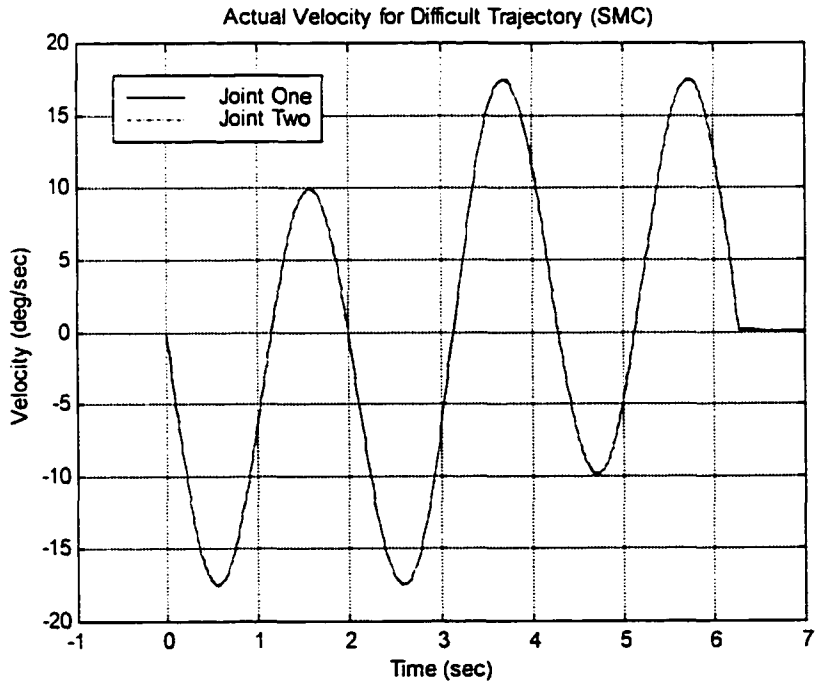


Figure 7.7 Actual Velocity for SMC

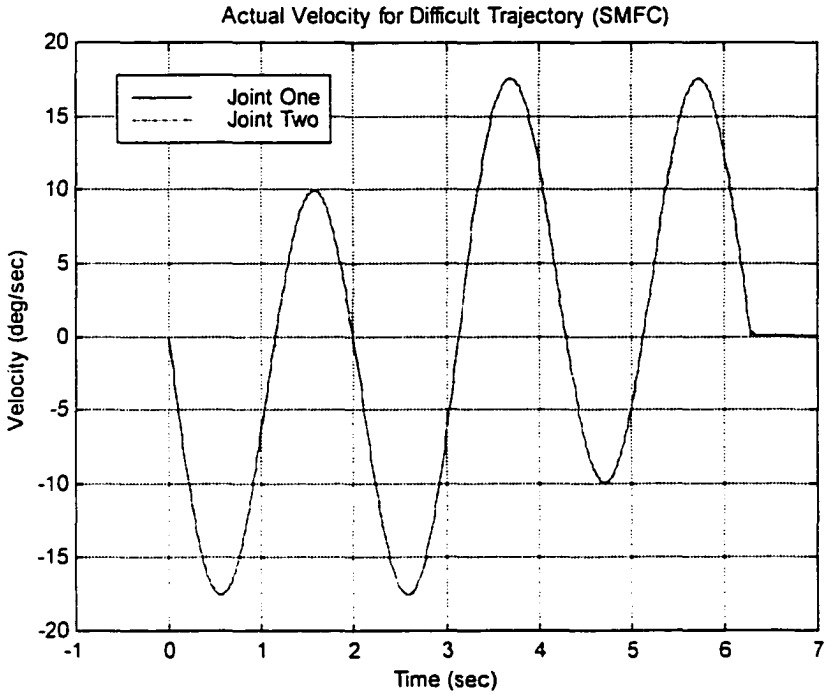


Figure 7.6 Actual Velocity for SMFC

The Sliding Mode Controller (SMC) and the Sliding Mode Fuzzy Controller (SMFC) are successful at keeping the manipulator's joint trajectories close to the desired trajectories. If the actual joint positions and velocities, calculated by the simulation program, were shown along with the desired positions and velocities, plotted on the previous pages, the lines would overlap and they could not be distinguished from each other. However, the PID Controller shown an overshoot in its velocity plot. A better way to show the accuracy of the PIDC, SMC and SMFC are the plots of position and velocity errors versus time for joints one and two, shown as Figures (7.9), (7,10), (7.11), (7.12), (7.13) and (7.14). The maximum position error in PIDC is -1.9×10^{-2} degrees for joint one and -6.1×10^{-2} degrees for joint two, while the maximum velocity errors is found to

be −0.179 degrees per second for joint one and 0.511 degrees per second for joint two. For The SMC, the maximum position error in is -2.41×10^{-2} degrees for joint one and -1.58×10^{-2} degrees for joint two, while the maximum velocity errors is found to be 0.095 degrees per second for joint one and 0.215 degrees per second for joint two. On the other hand. the maximum position error in SMFC is -2.41×10^{-2} degrees for joint one and -1.58×10^{-2} degrees for joint two, while the maximum velocity errors is found to be 0.095 degrees per second for joint one and 0.215 degrees per second for joint two.

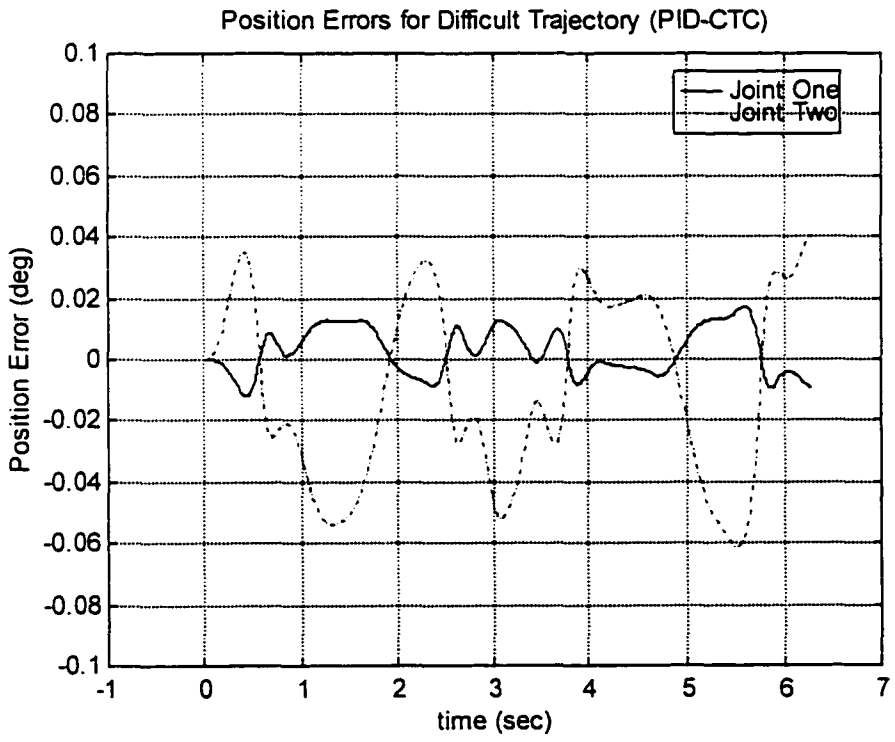


Figure 7.9 Position Errors for PID

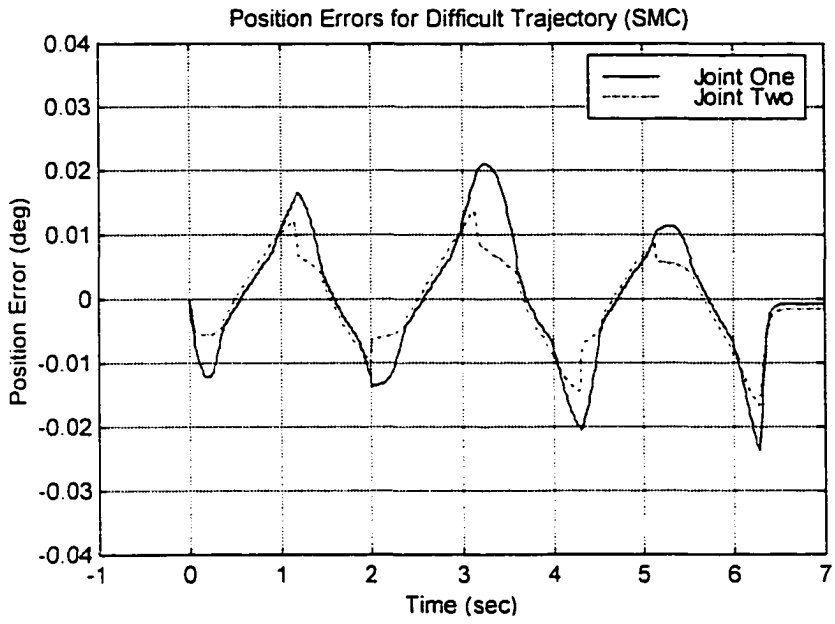


Figure 7.10 Position Errors for SMC

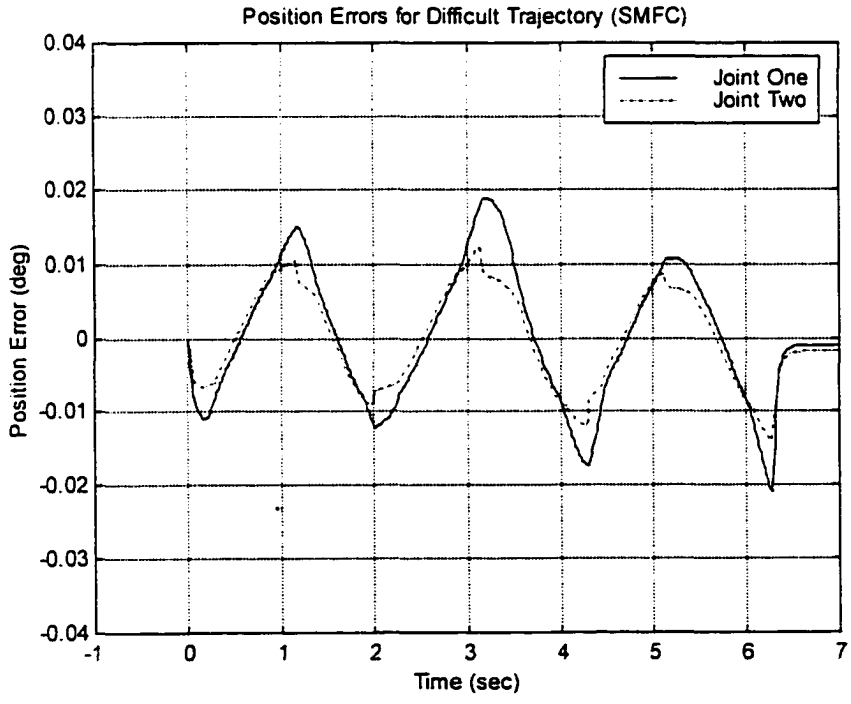


Figure 7.11 Position Errors for SMFC

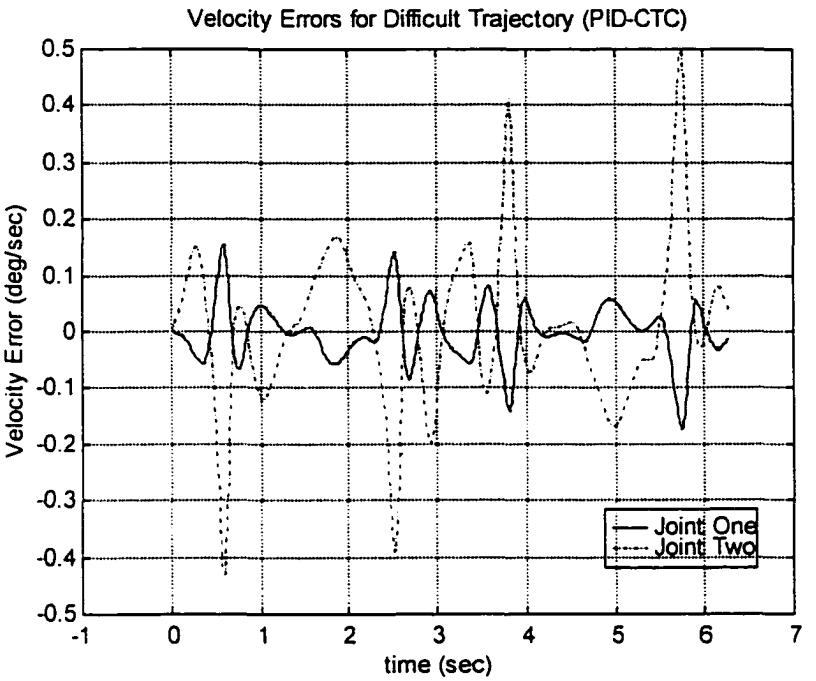


Figure 7.12 Velocity Errors for PIDC

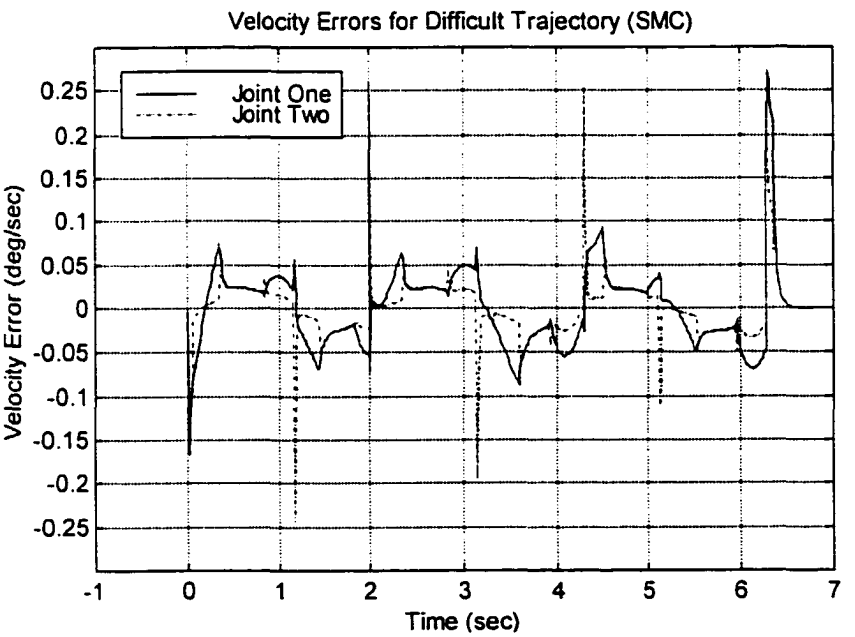


Figure 7.13 Velocity Errors for SMC

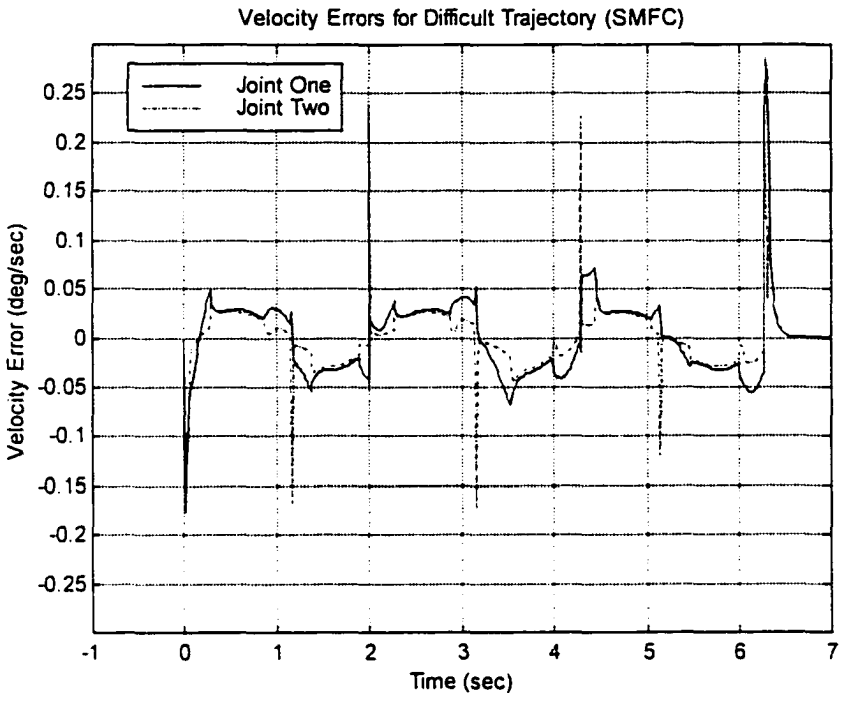


Figure 7.14 Velocity Errors for SMFC

It is also important to closely examine the values that are calculated by the control torque equation and applied to the manipulator joints for SMC and SMFC. The torque curve should ideally be smooth, although in reality, spikes cannot be completely avoided. This is because it takes less energy to keep an object in motion than it does to start or stop that motion. If Figure (7.13) and (7.14), the control torques versus time is examined, spikes can be seen exactly when the motion begins and ends. It is interesting to note that the torques do not go back to zero when the trajectory ends. This is because small errors are present which the SMC and SMFC controllers try to reduce. However, the torque for the PIDC as in Figure (7.15) started increasing with each movement of joints one and two. This is means that the robot must have larger motor than the other controllers in

order to follow the desired trajectory. So, this might consider one of the disadvantages for using PID controller.

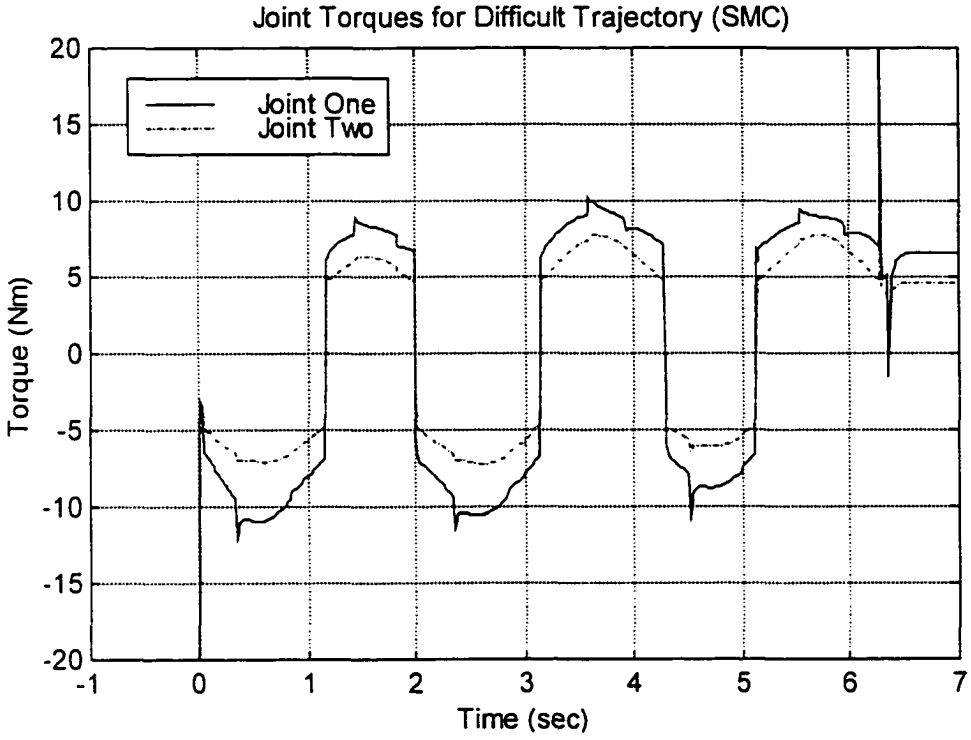


Figure 7.15 Joint Torques for SMC

Except for the PID controller, the results from the difficult trajectory show that the errors are the greatest when the desired motion starts or stops and when the desired velocity changes sign. These excitations cause the control torque to jump and occur at 0.00, 1.15, 2.00, 3.14, 4.29, 5.14 and 6.28 seconds.

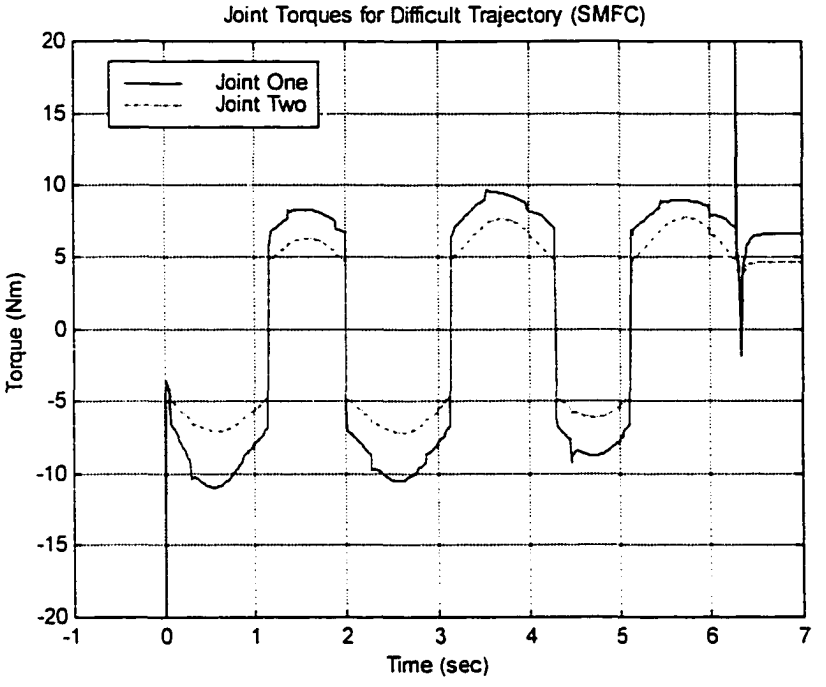


Figure 7.16 Joint Torques for SMFC

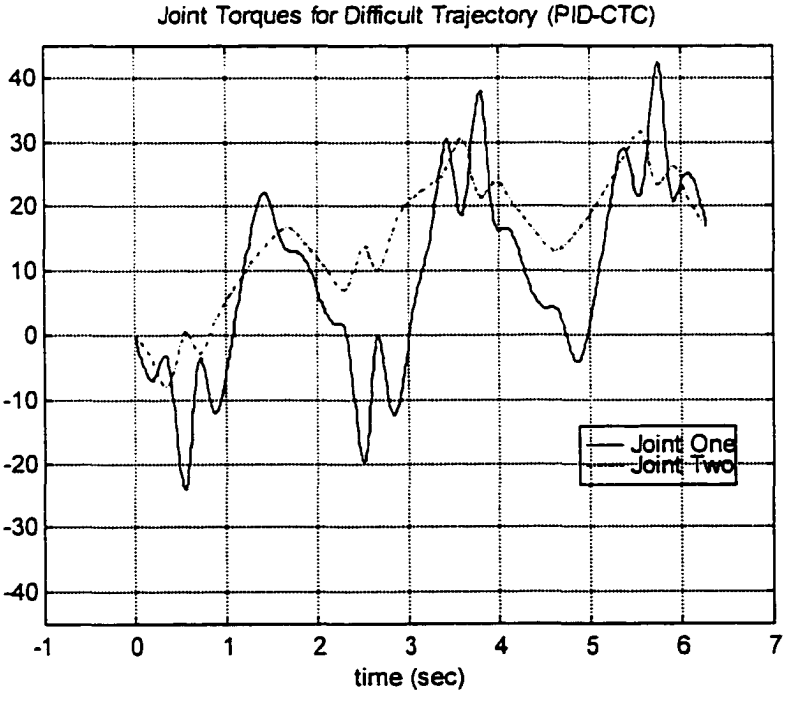


Figure 7.17 Joint Torques for PIDC

The smaller jumps in the velocity error and control torque that appear between the large ones described above are caused by sign changes in the desired acceleration. These sign changes in the acceleration arise when a maxima exists in the desired velocity. All of these excitations occur despite the fact that smooth trajectories are specified. If the desired trajectories were not smooth, these difficulties would only be compounded.

If we compare the performance of all three controllers, we will conclude the following. Even though the PID controller is simple to design, it has larger errors than the SMC and SMFC. So, we might say robustness and stability of the PID controller is not guaranteed. If we try to increase the gains value this might has its effect in the performance of the PID controller, but this will not be fare comparison with the other controllers. Because of this, we did not include the PID controller in the second test.

Both SMC and SMFC preformed very well in comparison to the PID controller regarding position and velocity errors. In addition, we limited their torques because limitation of drive torques offers some advantages, such as a reduction in energy consumption or avoidance of overheating of D.C. motors. Then we might say stability and robustness in both of these controllers are guaranteed. So, we will check for their performance again in the second test.

7.2.3 Test of Payloads

The simulation results presented up to this point have all assumed that the joint sensors had no error. One aspect of real manipulators that is not included in the simulation at this time is the fact that the robot will carry some sort of payload. This will be simulated in the program by increasing the mass of link two in the integrator model by

2.3 kg at 2.0 seconds, making the mass of link two equal to 4.3 kg. This will simulate the robot picking up an object that is three times the weight of the uncertainty designed into the controller. To simulate the object being placed down, the mass of link two will go back to the original mass of 2 kg when time is equal to 5.0 seconds. The resulting plots of the position and velocity errors for both SMC and SMFC are shown on the next page as Figures (7.18), (7.19), (7.20) and (7.21), while the control-torques are shown as Figures (7.22) and (7.23) respectively.

No deterioration can be seen in the positional accuracy of the manipulator joints. In fact, the load added to the link seem to have disturbed both controllers and given them the nudge it needed to overcome gravity and Coulomb friction and eliminate the steady-state errors. Unfortunately, the same disturbances that help eliminate the steady-state errors introduce a small amount of chattering. This can be seen by the extra jitters found in the velocity errors of Figures (7.20) and (7.21) when compared to the original velocity errors of Figures (7.13) and (7.14).

Another indication of the payload's presence is the slightly larger magnitude of chattering found in the control torque. This can be seen in Figures (7.22) and (7.23) by comparing the torques during the period when the payload is present, from two to five seconds, to the torques during the period when the payload is not present.

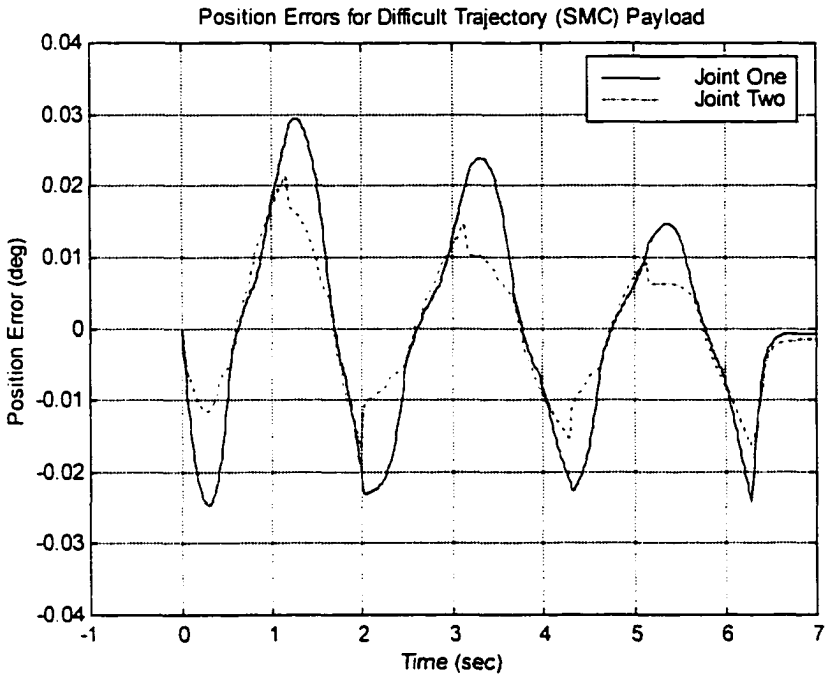


Figure 7.18 Position Errors for SMC (Payload)

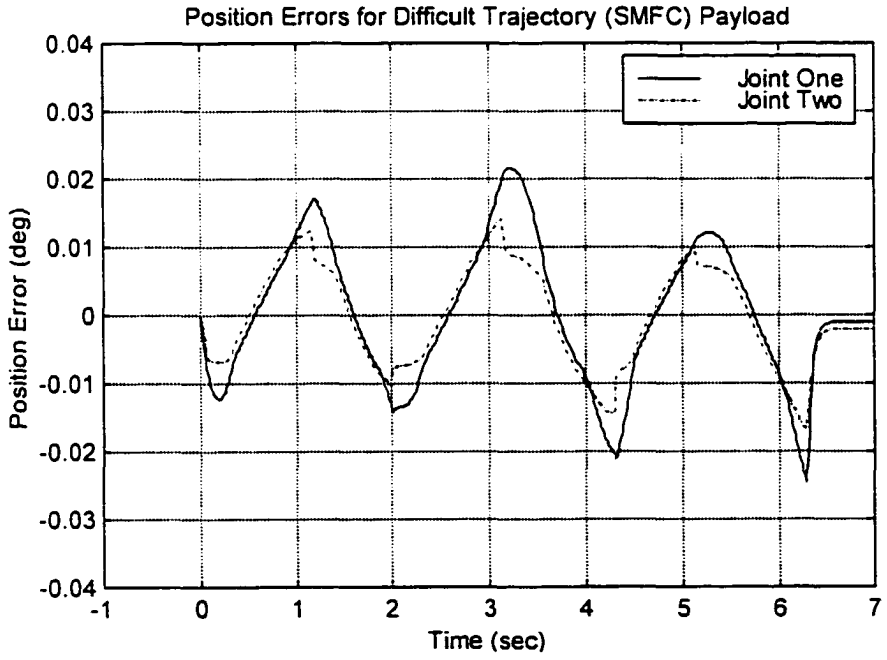


Figure 7.19 Position Errors for SMFC (Payload)

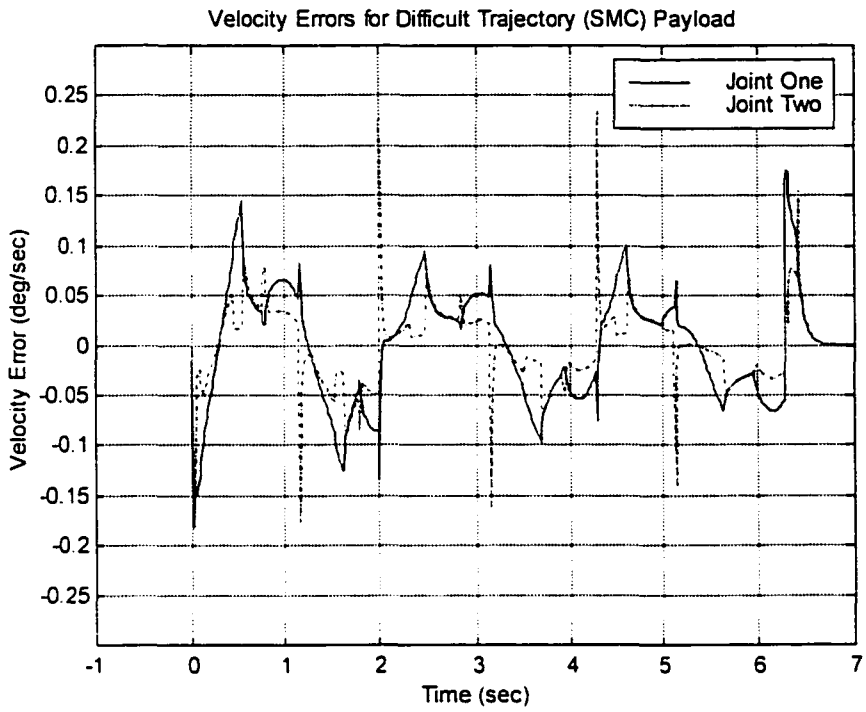


Figure 7.20 Velocity Errors for SMC (Payload)

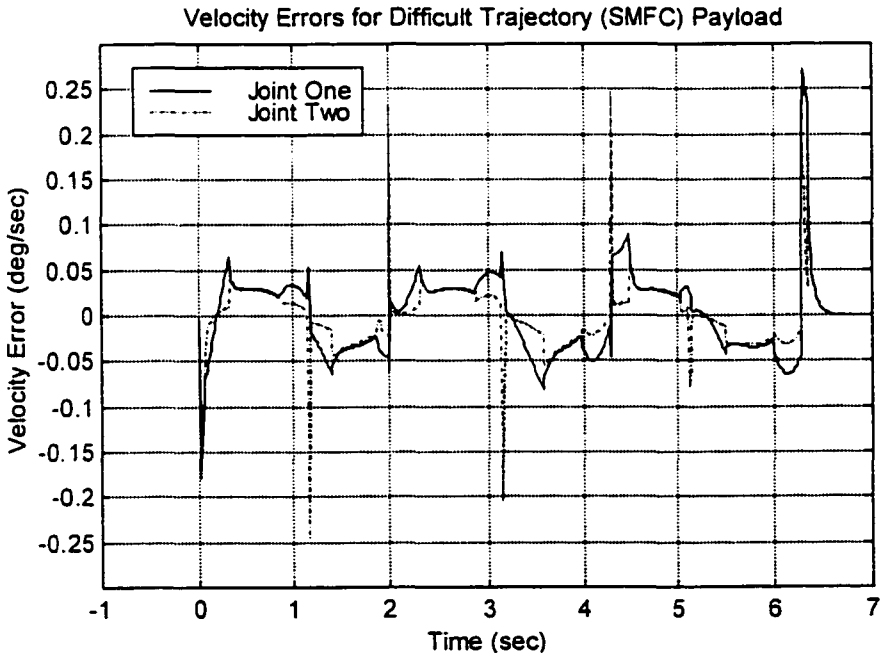


Figure 7.21 Velocity Errors for SMFC (Payload)

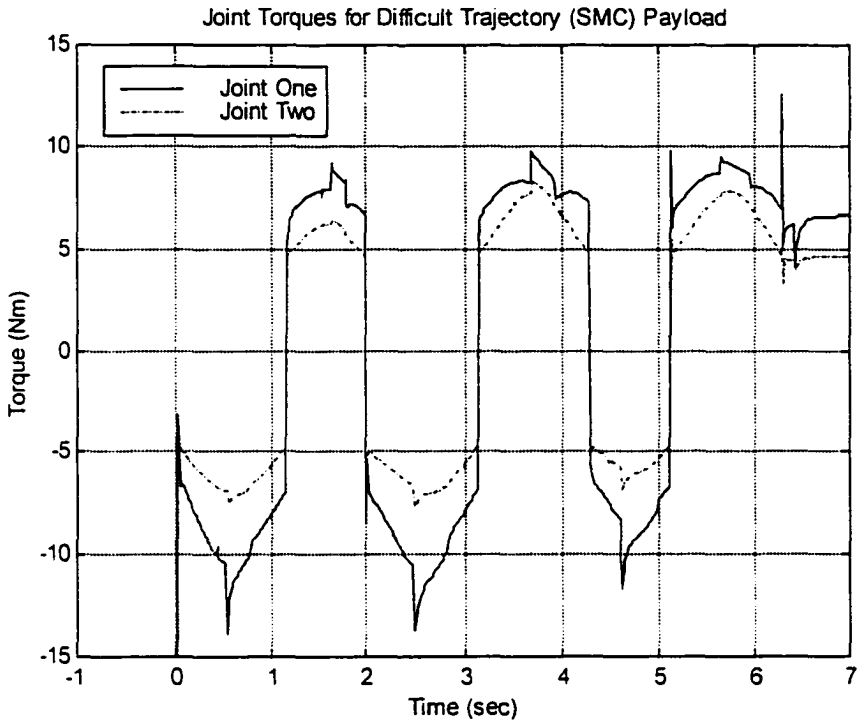


Figure 7.22 Joint Torques for SMC (Payload)

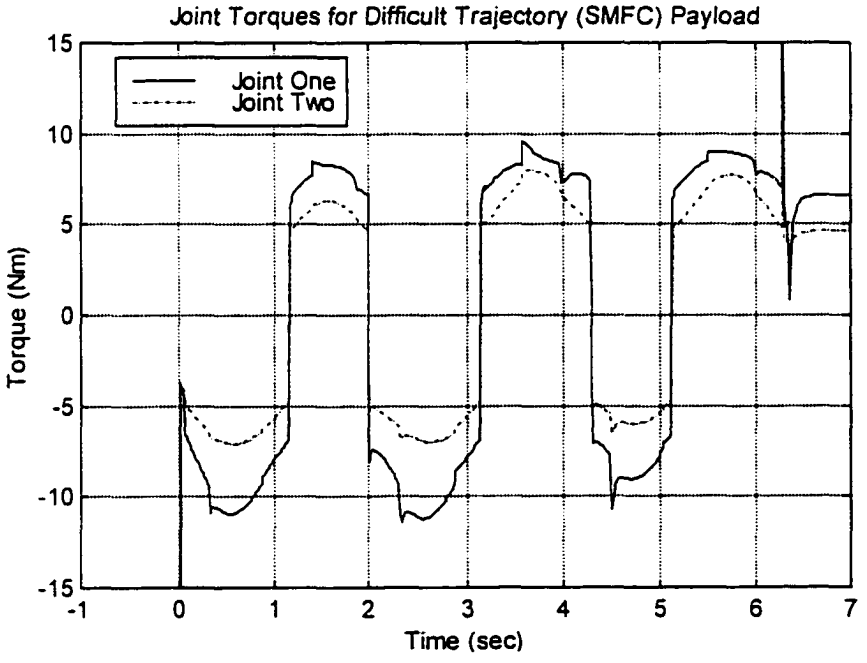


Figure 7.23 Joint Torques for SMFC (Payload)

If we try to compare the performance of SMC and SMFC controllers after the sensor noise and payloads test, we will conclude the following. Even though the SMC controller is simple to design, it has larger errors than the SMFC controller. So, we might say robustness and stability of the SMC controller is not guaranteed during this test. If we try to increase the boundary layer value, this might has its effect in the performance of the PID controller, but this will not be fare comparison with the SMFC controller. In contrast, SMFC controller preformed very well in comparison to the SMC controller regarding position and velocity errors. In addition, it uses less torque to follow the desired trajectory. Then we might say stability and robustness in the Sliding Mode Fuzzy Controller is guaranteed. This will make us recommended that SMFC could be used as robot controller but still not adaptive controller.

CHAPTER VIII

Extended Kalman Filter

8.1 System Modeling and Identification

Mathematical modeling is one of the most fundamental areas in science and engineering. A system model is a very useful and compact way to describe the knowledge about the system. Models can be used in many applications such as prediction, control, state estimation, simulation and analysis. Models for many systems can be derived based on the physical laws which require knowledge and insight about the process being modeled. From a mathematical point of view, the obtained model equations are generally in the form of linear or nonlinear algebraic, ordinary or partial differential equations. The advantage of this approach is that the variables and parameters obtained have a physical interpretation. However, if the system is too complex to allow physical modeling or the models obtained are not simple enough to be used, system identification is the solution.

System Identification, of which Professor Huang at Old Dominion University is one of the unique scholars in the area, is an extremely important and diverse field that spans many disciplines, e.g., signal processing, communications, control systems, statistics and many others. It deals with the problem of building mathematical models of dynamical systems based on observed input-output data. This approach does not depend on establishing rigorous mathematical models based on first principles; rather, it tries to establish a relationship between the input and output variables which does not depend on any physical or chemical laws.

Generally, the process of system identification can be divided into four steps: collecting input-output data, selecting model structure, estimating the parameters of the selected model structure and model validation to determine how "good" the developed model is. Thus, system identification can be thought of as determining the "best" model which describes the given data set among a set of candidate models. If the identified model is not acceptable according to model validity tests, a new model structure has to be tried. This shows that system identification is an iterative process and many trials might be needed before arriving at an acceptable model.

Since a mathematical description of a process is often a prerequisite to analysis and controller design, the study of system identification techniques has become an established branch of control theory. When choosing a nonlinear process model structure for control implementation, the following points have to be taken into account

- The parameter estimation methods should be applicable to the nonlinear process model.
- The nonlinear process model should be linear in the unknown parameters so that these parameters can be estimated.
- The process model should be sufficiently comprehensive to include all information about the process necessary to describe the input-output behavior for control purposes.
- The nonlinear process model should be suitable for derivation of nonlinear control algorithm.

While the theory of identification of linear dynamic systems has been already well established, e.g.,[87-90], the theory of identification of nonlinear systems is not yet

satisfactory. The need for nonlinear system models comes from the fact that most of the physical systems encountered in practice are nonlinear to some extent. Also, the linear models can only be used under limited conditions and special assumptions and are often found to be inadequate and might give rise to misleading results. Therefore, introducing nonlinear models is a necessity, especially for systems were accurate modeling is critical. The problem of proper descriptions of nonlinear systems is still under discussion. This can be attributed to the complexity and diversity of nonlinear systems. Thus, in the field of nonlinear system identification there is an obvious need for new ideas, methods and algorithms, and further research is required to develop improved identification techniques. Moreover, if these methods are to be used in control systems applications, they should be simple enough to allow the use of standard controller design methods.

8.2 Choice of Unique Method

The problem of identification of nonlinear systems can be divided into two groups: identification of deterministic systems (noise free situation) and stochastic systems (existence of plant and observation noise). The latter can be solved by means of the Extended Kalman Filter method in which the system is linearized along the reference trajectory, then transformed into a discrete-time equivalent form.

A combined parameter and state estimation procedure is then used. The Extended Kalman Filter method assumes validity of linearization, which makes it similar to Bellman's quasilinearization method.

8.3 Kalman Filters

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. The filter is very powerful in several aspects: it supports estimations of past, present and even future states, and it can do so even when the precise nature of the modeled system is unknown.

In nature, however, most physical problems or processes are nonlinear. Consequently, the nonlinear systems must be linearized (that is, approximated) before the linear filter theory can be applied. Specifically, the problem of combined state and parameter estimation was originally posed as a nonlinear state estimation problem using the Extended Kalman Filter (EKF). Since this requires a linear approximation of a nonlinear system about the current estimate, divergence may result if the initial estimate is poor. Moreover, not much is known about the convergence properties of the EKF, and the conditions for acceptability of the solution are vague. In the following subsections, we briefly state well-known formulations and results for Kalman filters and extended Kalman filters.

8.3.1 Continuous-Time Kalman Filter

The following results are taken from reference[91]. Let a system state $x(t)$ be generated by the following model:

$$\dot{x}(t) = F(t)x(t) + G(t)w(t). \tag{8.1}$$

For which we can observe

$$z(t) = H(t)x(t) + \upsilon(t), \tag{8.2}$$

where $w(t)$ and $\upsilon(t)$ are independent random processes with the following statistics:

$$\begin{gathered} E\{w(t)\}E\{\upsilon(t)\} = 0 \\ \operatorname{cov}\{w(t), w(\tau)\} = Q(t)\delta_D(t-\tau) \\ \operatorname{cov}\{\upsilon(t), \upsilon(\tau)\} = R(t)\delta_D(t-\tau) \\ \operatorname{cov}\{w(t), \upsilon(\tau)\} = \operatorname{cov}\{w(t), w(t_0)\} = \operatorname{cov}\{\upsilon(t), x(\tau)\} = 0. \end{gathered} \tag{8.3}$$

The optimal estimate of $x(t)$ using $z(t)$ for $t_0 \le \tau \le t$ is $\hat{x}(t)$ and is given by the filter:

$$\frac{d\hat{x}(t)}{dt} = F(t)\hat{x}(t) + P(t)H^T(t)R^{-1}(t)\{z(t) - H(t)\hat{x}(t)\}, \tag{8.4}$$

where $P(t)$ is given by the error variance algorithm:

$$\frac{dP}{dt} = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - P(t)H^T(t)R^{-1}(t)H(t)P(t) \tag{8.5}$$

for which the initial conditions are:

$$\hat{x}(t_0) = E\{x(t_0)\}\ P(t_0) = \operatorname{var}\{\tilde{x}(t_0)\}. \tag{8.6}$$

8.3.1 Continuous-Time Extended Kalman Filter

Because the applications made in the following chapter deal with observations which are constant linear combinations of the states, the observation matrix $H(t)$ is not a function time, thus $H(t)$ has been replaced by H. The filtering technique described in the previous subsection can be extended to consider nonlinear system models and the observations:

$$\dot{x}(t) = f\left[x(t),t\right](t) + G\left[x(t),t\right]w(t) \tag{8.7}$$

$$z(t) = Hx(t) + v(t), \tag{8.8}$$

where $w(t)$ and $v(t)$ are independent random processes with their statistics given by (8.3). If the value of $f\left[x(t),t\right]$ can be approximated by

$$f\left[x(t),t\right] \approx f\left[\hat{x}(t),t\right] + \frac{\partial f}{\partial \hat{x}}\left[x(t) - \hat{x}(t)\right],$$

where terms of order higher than one in the series expansion are neglected, the following filter can be used:

$$\frac{d\hat{x}(t)}{dt} = f\left[x(t),t\right] + P(t)H^T(t)R^{-1}(t)\left\{z(t) - H\hat{x}(t)\right\} \tag{8.9}$$

$$\frac{dP}{dt} = \frac{\partial f}{\partial \hat{x}}P(t) + P(t)\frac{\partial f^T}{\partial \hat{x}} + G\left[x(t),t\right]Q(t)G^T\left[x(t),t\right] - P(t)H^T(t)R^{-1}(t)HP(t) \tag{8.9}$$

$$\hat{x}(t_0) = E\left\{x(t_0)\right\} P(t_0) = \mathrm{var}\left\{\tilde{x}(t_0)\right\} \tag{8.10}$$

The reader referred to [92–102] for more information about this new method.

CHAPTER IX

TAREK Method

In this chapter, a new adaptive motion control scheme for robust performance control of robot manipulators is presented. The proposed scheme is designed by combining the fuzzy logic control with the sliding mode control based on extended Kalman filter. Fuzzy logic controllers have been applied successfully in many applications and were shown to be superior to the classical controllers for some nonlinear systems. Sliding mode control is a powerful approach for controlling nonlinear and uncertain systems. It is a robust control method and can be applied in the presence of model uncertainties and parameter disturbances provided that the bounds of these uncertainties and disturbances are known. In the previous chapters, a control scheme called sliding fuzzy logic control (SMFC) is proposed, in which the principles of fuzzy logic control and sliding mode control are combined. The main advantage of SMFC is the system stability. Therefore, we design a new adaptive SMFC method that requires only position measurements. These measurements and the input torques are used in an extended Kalman filter (EKF) to estimate the inertial parameters of the full nonlinear robot model as well as the joint positions and velocities. These estimates are used by the SMFC to generate the input torques. The combination of the EKF and the SMFC is shown to result in a stable adaptive control scheme called trajectory-tracking adaptive robot with extended Kalman (TAREK) method. The theory behind the TAREK method provides clear guidelines on the selection of the design parameters for the controller. The proposed controller is applied to a two-link robot manipulator. Computer simulations show the robust performance of the proposed scheme.

9.1 Robot Model Revisited

In this section, the dynamic model for the robotic manipulator is developed. This model is based on the assumption that the links are rigid bodies. As we derived in chapter two in equation (2.27), the manipulator's model can be formulated using Lagrangian dynamics as:

$$\tau = M(q)\ddot{q} + V(q,\dot{q}) + F(\dot{q}) + G(q), \tag{9.1}$$

where $q, \dot{q}$, and $\ddot{q}$ are the manipulator joint position, velocity and acceleration vectors respectively. $M(q)$ is the inertia matrix, $V(q,\dot{q})$ Coriolis and centrifugal force vector, $G(q)$ the gravitational force vector, $F(\dot{q})$ the vector function of frictional forces and τ is the output torque vector applied by the motor to the robot joints. Since this equation is linear in the parameters [92-94], we have

$$\tau = Y(q,\dot{q},\ddot{q})\theta.$$

Also, the manipulator acceleration is given by

$$\ddot{q} = M^{-1}(q)\{\tau - F(\dot{q}) - V(q,\dot{q}) + G(q)\}. \tag{9.2}$$

In equation (9.1), one assumes that all the parameters are known, but in practice this might not be true. One can form a parameter vector θ which includes the unknown parameters (e.g., mass of the load, viscous friction coefficient, inertia and mass of the links). The parameter θ vector is treated as constant vector so $\dot{\theta} = 0$. Let us define a state vector $x = \left[q^T, \dot{q}^T, \theta^T\right]$ and the input $u(t) = \tau$ so that:

$$\dot{x} = f(x,u) = \begin{bmatrix} \dot{q} \\ M^{-1}(q,\theta)\{u(t) - F(\dot{q},\theta) - V(q,\dot{q},\theta) - G(\dot{q},\theta)\} \\ 0 \end{bmatrix}. \tag{9.3}$$

These equations represent a system where there is no input disturbances and no change in the parameters. If we want to take into account the possibility of input disturbances and parameter variations, u should be replaced by $u + w_1$ and $\dot{\theta} = 0$ by $\dot{\theta} = w_2$ where w_1 and w_2 are random variables. This is the Bayesian approach where time-variant parameters are modeled as random walk processes[95]. This results in:

$$\dot{x} = f(x,u) + G(x)w \tag{9.4}$$

$$G(x) = \begin{bmatrix} 0 & 0 \\ M^{-1}(q,\theta) & 0 \\ 0 & I \end{bmatrix} \tag{9.5}$$

$$w = \begin{bmatrix} w_1^T & w_2^T \end{bmatrix}^T. \tag{9.6}$$

An extended Kalman filter will be applied to this model. Thus, a perturbation model will be needed. This perturbation model is based on first order Taylor series expansion of the system model with respect to an estimated trajectory $\hat{x}$. Let us define:

$$\dot{\hat{x}} = f(\hat{x},u) \tag{9.7}$$

$$\delta\dot{x} \approx \frac{\partial f(\hat{x},u)}{\partial x}\delta x + G(\hat{x})w \tag{9.8}$$

with

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & I & 0 \\ F_{21}(t) & F_{22}(t) & F_{23}(t) \\ 0 & 0 & 0 \end{bmatrix} \tag{9.9}$$

and

$$F_{21}(t) = -M^{-1}\left\{\frac{\partial M}{\partial q}\ddot{q} + \frac{\partial V}{\partial q} + \frac{\partial g}{\partial q}\right\} \tag{9.10}$$

$$F_{22}(t) = -M^{-1}\left\{\frac{\partial F}{\partial \dot{q}} + \frac{\partial V}{\partial \dot{q}}\right\} \tag{9.11}$$

$$F_{21}(t) = -M^{-1}\left\{\frac{\partial M}{\partial \theta}\ddot{q} + \frac{\partial F}{\partial \theta} + \frac{\partial V}{\partial \theta} + \frac{\partial g}{\partial \theta}\right\}, \tag{9.12}$$

where all the partial derivatives are evaluated along the estimated trajectory. These equations can be written as a linear time-variant system:

$$\delta \dot{x} = F(t)\delta x + G(t)w. \tag{9.13}$$

Equation (9.13) describes a linearization of the nonlinear manipulator dynamics about the estimated trajectory. Therefore, one can use the model dynamics in Equation (9.13) to derive the extended Kalman filter.

9.2 TAREK Method [103]

The new method, Trajectory-tracking Adaptive Robot with Extended Kalman is an indirect adaptive controller. The control is based on the Sliding Mode Fuzzy Control (SMFC) using position, velocity and parameter estimates. Therefore, the control torque vector is given by

$$\tau = \hat{M}(\hat{q})\left[\ddot{q}_d + 2\lambda\left(\dot{q}_d - \hat{\dot{q}}\right) + \lambda^2\left(q_d - \hat{q}\right)\right] + \hat{V}\left(\hat{q}, \hat{\dot{q}}\right) + \hat{G}(\hat{q}) + \hat{F}\left(\hat{\dot{q}}\right), \tag{9.14}$$

where 2λ and λ^2 is part of the sliding surface $s = \ddot{q}_d + 2\lambda\dot{q} + \lambda^2 q$ and the hats denote the use of the estimated values of $q, \dot{q}$, and θ. The estimates are provided by the application of the continuous-time extended Kalman filter described in chapter 8 by Equations (8.7)-(8-10) to the robot model based on Equations (9.4)-(9-12) with position measurements

only. The controller structure is shown in Figure (9.1) where SMFC denotes the Sliding Mode Fuzzy Control method and EKF the extended Kalman filter. The EKF has two purposes: to provide estimates of the parameters and to reconstruct the velocity from position measurements $\left(H=\begin{bmatrix} I & 0 & 0 \end{bmatrix}\right)$. The stochastic interpretation of the Kalman filter will allow us to select $R(t)$ and $Q(t)$ (the covariance matrices) from practical considerations.

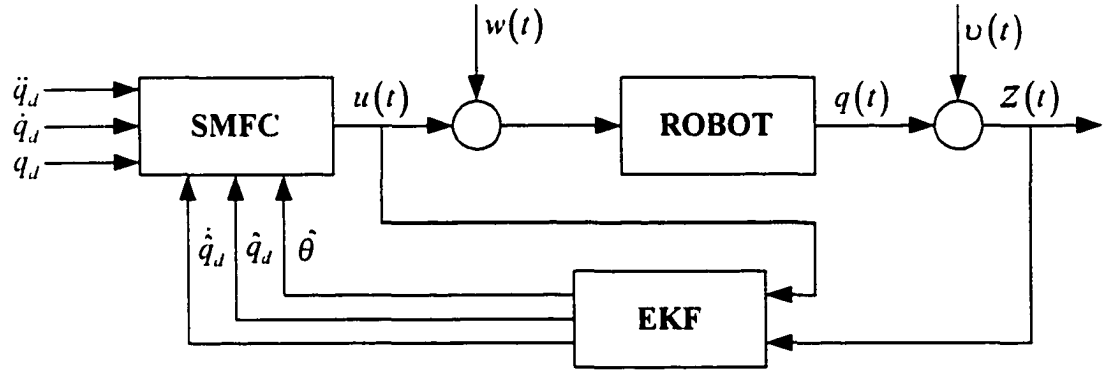


Figure 9.1 TAREK method controller structure

A modification will be introduced in order to improve the robustness of the system. This modification uses a priori bounds on the unknown parameters:

$$\theta_{min} < \theta < \theta_{max}.$$

When $\hat{\theta}$ satisfies these bounds, $\hat{M}(q)$ is guaranteed to be positive definite. This is not very restrictive since from physical considerations these bounds can be easily selected for most parameters (e.g., mass and inertia are positive and their maximum possible values can be easily guessed).

This modification can be considered as a σ-modification that uses a priori bounds on the unknown parameters. The state equations for the modified model are

$$\dot{x} = f(x,u) + \begin{bmatrix} 0 \\ 0 \\ \sigma(\theta) \end{bmatrix},$$

where $\sigma(\theta)$ is a vector function given by

$$\sigma_i(\theta) = \begin{cases} -\alpha(\theta_i - \theta_{i\min}) & \text{if} \quad \theta_i < \theta_{i\min} \\ 0 & \text{if} \quad \theta_{i\min} < \theta_i < \theta_{i\max} \\ -\alpha(\theta_i - \theta_{i\max}) & \text{if} \quad \theta_i < \theta_{i\max} \end{cases}$$

where $\theta_{i\min}$ and $\theta_{i\max}$ are the known bounds on the i-th parameters and $\alpha > 0$. This modification, when it is included in the filter, preserves the positive definiteness of $\hat{M}(q)$ and avoids parameter drift by using

$$f'(x,u) = f(x,u) + \begin{bmatrix} 0 \\ 0 \\ \sigma(\theta) \end{bmatrix}$$

$$\frac{\partial f'}{\partial x} = \frac{\partial f}{\partial x} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \dfrac{\partial \sigma(\theta)}{\partial \theta} \end{bmatrix},$$

where $\partial\sigma(\theta)/\partial\theta$ is a negative semi-definite diagonal matrix with:

$$\frac{\partial \sigma_i(\theta)}{\partial \theta_i} = \begin{cases} -\alpha & \text{if} \quad \theta_i < \theta_{i\min} \\ 0 & \text{if} \quad \theta_{i\min} < \theta_i < \theta_{i\max} \\ -\alpha & \text{if} \quad \theta_i < \theta_{i\max} \end{cases}.$$

Hence, this modification will come into effect when at least one parameter is outside the a priori bounds; thus, no parameter can increase without bounds since they are stabilized

as soon as they cross the bounds. The higher α is chosen, the lower the depth of the penetration of $\hat{\theta}$ across the boundaries defined by

$$\theta_{min} < \theta < \theta_{max}.$$

This modification uses the same a priori information as the resetting rule of Craig's method[104] and is similar to the σ-modification used in[105].

9.3 Choice of Design Parameters

9.3.1 Selection of EKF initial condition

From filtering theory the initial filtered state estimates are the expected values of these states at the beginning of control. Hence, for the robotic manipulator starting at rest and at a known position (up to the accuracy of the resolver), the initial filtered states are:

$$\hat{x}_0 = \begin{bmatrix} z(t_0) \\ 0 \\ \theta_0 \end{bmatrix},$$

where $z(t_0)$ is the first measurement of position and θ_0 the best estimate available of the unknown parameters.

Accordingly, the initial covariance matrix, is given by

$$P_0 = \begin{bmatrix} P_q & 0 & 0 \\ 0 & P_{\dot{q}} & 0 \\ 0 & 0 & P_\theta \end{bmatrix},$$

where P_q is a diagonal matrix of the estimated variances of the respective measurements (these values are lower bounded by the accuracy of the analog-to-digital A/D converters used to measure position), $P_{\dot{q}}$ is a diagonal matrix of the variances representing the

confidence that we have that the robot is indeed starting to operate from rest, and finally P_0 is the estimated values of the covariance of the possible errors in the estimated parameters. The higher P_0 is, the less we are sure of the accuracy of our initial estimates.

9.3.2 Selection of $Q(t)$ and $R(t)$

The matrices $Q(t)$ and $R(t)$ are other parameters that must be defined to simulate the robot model when using the extended Kalman filter. These matrices should be representative of the noise content of the measurements and control inputs[100].

- $R(t)$ is simply a diagonal matrix for the measurement variances.
- $Q(t)$ is composed of two diagonal matrices $Q_1(t)Q_2(t)$.

$$Q(t)=\begin{bmatrix} Q_1(t) & 0 \\ 0 & Q_2(t) \end{bmatrix}$$

  - $Q_1(t)$ reflects the magnitude of the disturbances caused by unmodeled dynamics, hence the confidence in our model. It also includes the effect of the truncation error in the series expansion, hence $Q_1(t)>0$.
  - $Q_1(t)$ represents the speed at which the parameter vector is estimated to vary in a random walk fashion.

9.4 Simulation and Results

In Figure (9.2), a structure of a two-link robot manipulator that will be used in the simulation of this controller. Defining the parameters:

$$p_1 = m_1\left(c_1^2 + o_1^2\right) + I_1 + m_2 l_1^2$$
$$p_2 = m_2\left(c_2^2 + o_2^2\right) + I_2$$
$$p_3 = m_2 l_1 c_2$$
$$p_3 = m_2 l_1 o_2$$

and applying Lagrange's equation to the system yields:

$$M(q) = \begin{bmatrix} p_1 + p_2 + 2p_3\cos q_2 - 2p_4\sin q_2 & sym \\ p_2 + p_3\cos q_2 - p_4\sin q_2 & p_2 \end{bmatrix}$$
$$V(q,\dot{q}) = \begin{bmatrix} -\left(p_3\sin q_2 + p_4\cos q_2\right)\dot{q}_2\left(2\dot{q}_1 + \dot{q}_2\right) \\ \left(p_3\sin q_2 + p_4\cos q_2\right)\dot{q}_1^2 \end{bmatrix}$$
$$g(q) = F(\dot{q}) = 0.$$

The same robot parameters presented in chapter seven will be used here, hence

$$\theta = \begin{bmatrix} 0.24 & 0.06 & 0.09 & 0 \end{bmatrix}^T \text{ kgm}^2.$$

All simulation will be started with the parameter estimates:

$$\theta = \begin{bmatrix} 0.3 & 0.2 & 0.1 & 0 \end{bmatrix}^T \text{ kgm}^2.$$

The desired path given in chapter seven will be used together with all parameters of the SMFC controller. The simulation will be with both cases without measurement noise. In both cases, the covariance matrices were set as

$$R(k) = 0.0001I, \quad Q_1(k) = 0.01I, \quad Q_2(k) = 0.0025I, \quad P(0) = 0.0001I.$$

After we run the simulation for the sliding mode fuzzy control based on extended Kalman filter, we come up with the following result as shown in figure (9.3) and (9.4). This result indicated that TAREK method could work almost like SMFC and guaranteed its stability performance.

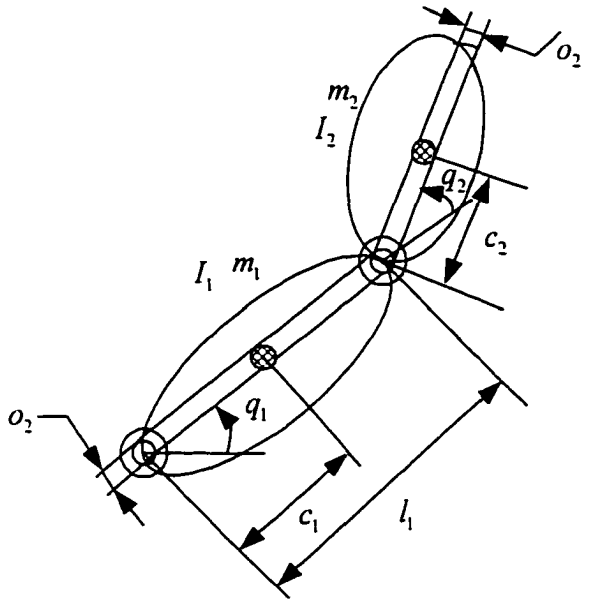


Figure 9.2 Robot Structure for SMFC with EKF

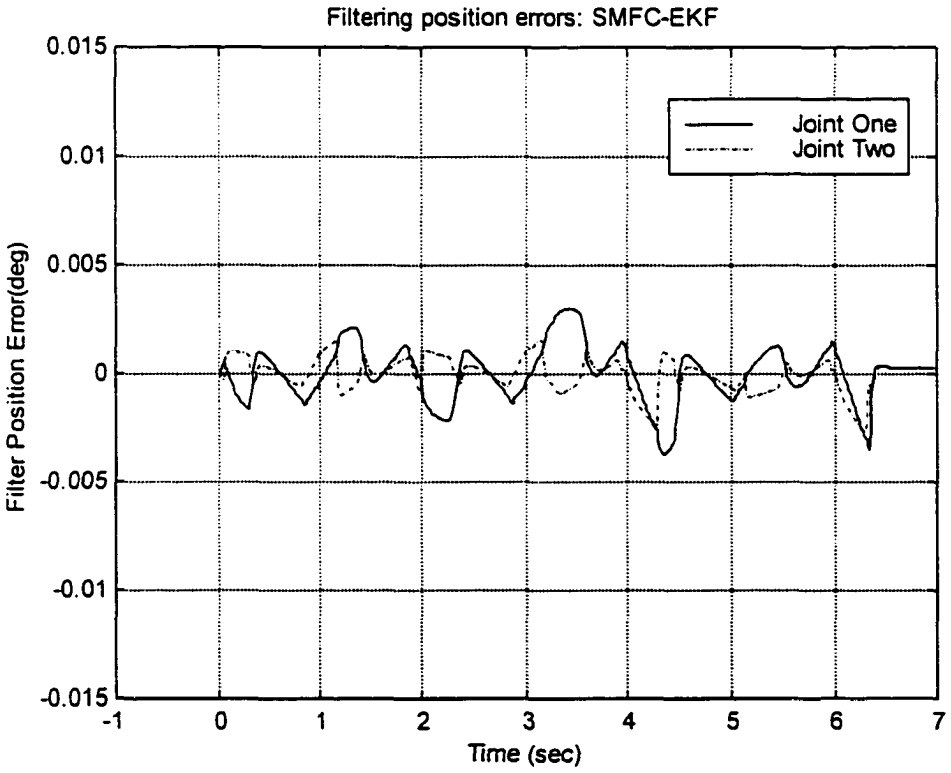


Figure 9.3 Filtering position errors: $\hat{q}_1 - q_1 =$ solid line, $\hat{q}_2 - q_2 =$ dashed line

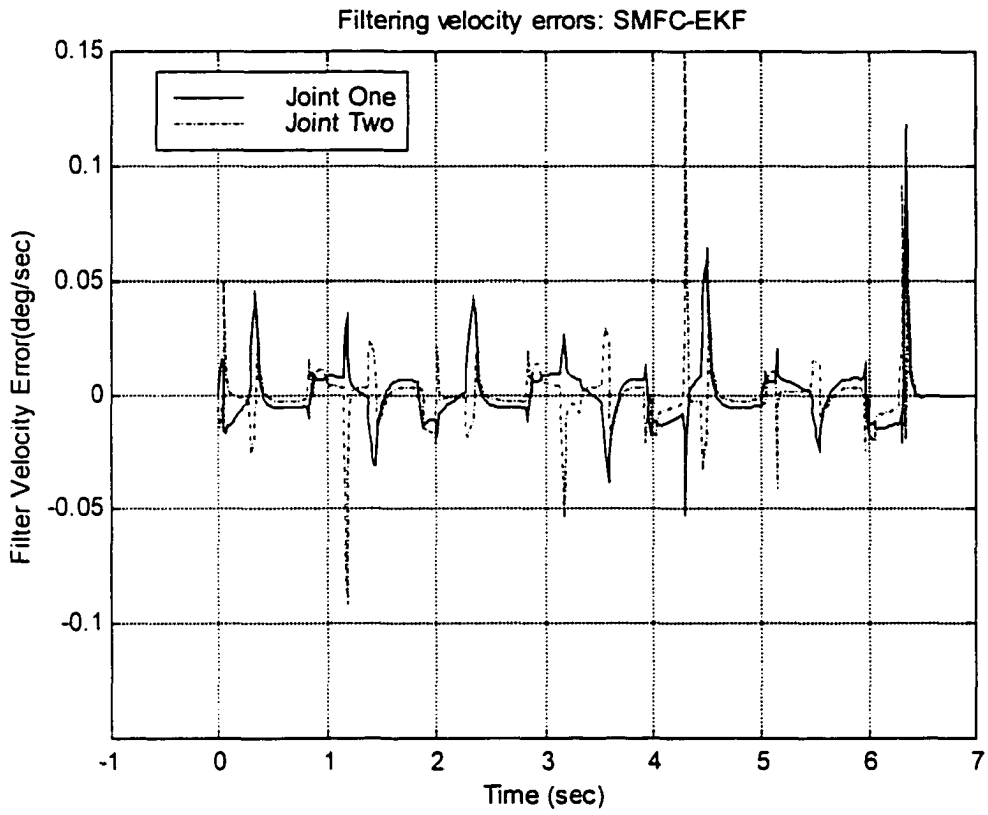


Figure 9.4 Filtering velocity errors: $\hat{\dot{q}}_1 - \dot{q}_1$ = solid line, $\hat{\dot{q}}_2 - \dot{q}_2$ = dashed line

CHAPTER X

Conclusions and Future Work

A new adaptive motion control scheme for robust performance control of robot manipulators is presented. The proposed scheme is designed by combining the fuzzy logic control with the sliding mode control based. Fuzzy logic controllers have been applied successfully in many applications and were shown to be superior to the classical controllers for some nonlinear systems. Sliding mode control is a powerful approach for controlling nonlinear and uncertain systems. It is a robust control method and can be applied in the presence of model uncertainties and parameter disturbances, provided that the bounds of these uncertainties and disturbances are known. In this work, a control scheme called sliding mode fuzzy logic control (SMFC) is proposed in which the principles of fuzzy logic control and sliding mode control are combined.

In addition, a new adaptive motion control scheme for robust performance control of robot manipulators is also presented. The proposed scheme is designed by SMFC based on extended Kalman filter. The new design adaptive SMFC method requires only position measurements. These measurements and the input torques are used in an extended Kalman filter (EKF) to estimate the inertial parameters of the full nonlinear robot model as well as the joint positions and velocities. These estimates are used by the SMFC to generate the input torques.

Through computer simulation and results, we prove that Sliding Mode Fuzzy Control is a stable and robust controller that can be used with application to robot manipulators. Compression between PID Controller, Sliding Mode Controller (SMC),

and Sliding Mode Fuzzy Controller (SMFC) showed that SMFC gives better performance than the other controllers. Even with increase of the payload, the SMFC gives better performance than the SMC.

Moreover, the combination of the EKF and the SMFC is shown to result in a stable adaptive control scheme, called trajectory-tracking adaptive robot with extended Kalman (TAREK) method. The theory behind TAREK method provides clear guidelines on the selection of the design parameters for the controller. When it is applied to a two-link robot manipulator, computer simulations show the robust performance of proposed schemes.

Sliding Mode Fuzzy Controller shows that it is a promising controller to be used with robotics systems. Its error free validity can be reached in the near future. With new technology, this robot controller can be implement to perform brain surgery. This can not be done without experimental research and high tech facilities.

BIBLIOGRAPHY

1 Uicker, J. J., "On the Dynamic Analysis of Spatial Linkages using 4 × 4 Matrices," *Transaction of ASME, Journal of Applied Mechanics*, Vol. 31, Series E, pp.309-314, 1965.

2 Bejczy, A. K., "Robot Arm Dynamics and Control," Technical Memo 33-669, Jet Propulsion Laboratory, Pasadena, CA, 1974.

3 Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Transaction on Systems, Man, Cybernetic*, Vol. SMC-10, No. 11, pp.730-736, 1980.

4 Luh, J. Y, Walker, M. W., and Paul, R. P., "On-Line Computational Scheme for Mechanical Manipulators," *Transaction of ASME, Journal of Dynamic Systems, Measurements and Control*, Vol. 120, pp. 69-76, 1980.

5 Lee, C. S., Lee, B. H., and Nigam, R, "Development of the Generalized d'Alembert Equations of Motion for Mechanical Manipulators," Proceedings of 22nd Conference on Decision and Control, San Antonio, TX, 1983, pp.1205-1210,.

6 Symon, K. R., *Mechanics*, Addison-Wesley, New York, 1971.

7 Marion, J. B., *Classical Dynamics*, Academic Press, New York, 1965.

8 Paul, R. P., *Robot Manipulators*, MIT Press, Cambridge, MA, 1981.

9 Lee, C. S., Gonzalez, R. C., and Fu, K. S., *Tutorial on Robotics*, IEEE Press, New York, 1983.

10 Asada, H., and Slotine, J.-J. E., *Robot Analysis and Control*, Wiley, New York, 1986.

11 Spong, M. W., and Vidyasagar, M., *Robot Dynamics and Control*, Wiley, New York, 1989.

12 Graig, J. J., *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, Reading, MA, 1988.

13 Schilling, R. J., Fundamentals of Robotics, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

14 Seraji, H., "Design of Digital Two and Three Term Controller for Discrete Time Multivariable Systems," *International Journal of Control*, Vol. 38, No. 4, pp.843-865, 1983.

15 Kovico, H.N., Tanttu, J.T., "Tuning of PID Controllers: Survey of SISO and MIMO Techniques," Intelligent Tuning and Adaptive Control, Selected Papers from the IFAC Symposium, 1994, pp. 57-80.

16 Unar, M.A., *Multi-loop PID Control Design*, M.Sc. Thesis, University of Glasgow, 1995.

17 Ziegler, J.G., "Those Magnificent Men and Their Controlling Machines," *Transaction of ASME, Journal of Dynamics Systems, Measurement and Control*, Vol. 97, pp.297-303, 1997.

18 Astrom, K.J., Hagglund, T., "Automatic Tuning of Simple Regulators with Specification on Phase and Amplitude Margins," *Automatica*, Vol. 20, pp.645-65, 1984.

19 Astrom, K.J., *PID Controllers*, Instruments Society of America, Research Triangle Park, 1995.

20 Astrom, K.J., Hagglund, T., *The Control Handbook* by Levine, editor, pp.817-826, CRC Press, Boca Raton, 1996.

21 Franklin, G., Powell, J., Workman, *Digital Control of Dynamics systems*, Addison-Wesley, Reading, MA, 1994.

22 Brogan, W.L., *Modern Control Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

23 Qu, Z., Dorsey, J., "Robust Tracking Control of Robots by Linear Feedback Law," *IEEE, Transaction on Automatic Control*, Vol. AC-36, No. 9, pp. 1081-1084, September 1991.

24 Hunt, L., Su, R., Meyer, G., "Global Transformation of Nonlinear Systems," *IEEE, Transaction on Automatic Control*, Vol. AC-28, No. 1, pp. 24-31, January 1983.

25 Gilbert, E., Ha, I., "An Approach to Nonlinear Feedback Control with Applications to Robotics," *IEEE, Transaction on System Man and Cybernetics*, Vol. SMC-14, No. 6, pp.879-884, November/December 1994.

26 Lee, C., Chen, M., "A Suboptimal control design for Mechanical Manipulators," *Proceedings of American Control Conference*, June 1983, pp.1056-1061.

27 Krutz, K., "On Manipulator Control by Exact Linearization," *IEEE, Transaction on Automatic Control*, Vol. AC-34, No. 7, pp. 763-767, July 1989.

28 Craig, J. J., *Introduction To Robotics: Mechanics and Control*, Addison-Wesley, Reading, MA, 1989.

29 Rocco, P., "Stability of PID Control for Industrial Robot Arms," *IEEE, Transaction on Robotics and Automation*, Vol. 12, No. 4, pp. 606-614, August 1989.

30 Tzafest, S., Dritsas, L., "Combined Computed Torque and Model Reference Adaptive Control of Robot Systems," *Journal of Franklin Institute*, Vol. 327, No. 2, pp.273-294, 1990.

31 Dawson, D., Qu, Z. Dorsey, J., "On Learning Control of A Robot Manipulator," Proceedings of the 28th Conference on Decision and Control, Tampa, FL, December 1989, pp.2632-2634.

32 Utkin, V. I., "Variable Structure Systems with Sliding Modes," *IEEE, Transaction on Automatic Control*, Vol. AC-22, No. 2, pp.212-222, April 1977.

33 Utkin, V. I., "Variable Structure Systems: Present and Future," *Automation and Remote Control*, Vol. 44, No. 9, Part 1, pp.1105-1120, 1983.

34 Utkin, V. I., "Discontinuous Control system: State of Art in Theory and Applications," *Selected Papers from the 10th Triennial World Congress of the IFAC*, Vol. 1, Munich, West Germany, July 1987, pp25-45.

35 DeCarlo, R., Zak, S., Matthews, G., "Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial," *Proceedings of the IEEE*, Vol. 76, No. 3, March 1988, pp. 212-232.

36 Walcott, B. Zak, S., "Laboratory Investigations in Microprocessor-Based Variable Structure Control," *IEEE, Transaction on Education*, Vol. E-30, No. 4, pp.227-243, November 1987.

37 Yong, K. D., "Controller Design for Manipulator Using Theory of Variable Structure Systems," *IEEE, Transaction on System, Man, and Cybernetics*, Vol. 8, pp.101-109, 1994.

38 Slotine, 3.3., and Sastry, S. S., "Tracking Control of Non-Linear Systems using Sliding Surfaces, with Application to Robot Manipulator," *International Journal of Control*, Vol.38, No.2, pp.465-492, 1983.

39 Ambrosino, G., Celentano, G., and Garfalo, F., "Variable Structure Model Reference Adaptive Control Systems," *International Journal of Control*, Vol. 39, No.6, pp.1339-1349, 1984.

40 Burton, 3. A., and Zinober, A. S. I., "Continuous Approximation of Variable Structure Control," *International Journal of Systems Science*, Vol. 17, No.5, pp.875-885, May 1986.

41 Hashimoto, H., Maruyama, K., and Harashima, F., "A Microprocessor-Based Robot Manipulator Control with Sliding Mode," *IEEE, Transactions on Industrial Electronics*, Vol.34, No. 1, pp.11-18, February 1987.

42 Morgan, R. G. and Ozguner, U., "A Decentralized Variable Structure Control Algorithm for Robotic Manipulators," *IEEE Journal of Robotics and Automation*, Vol.1, No.1, pp.57-65, March 1985.

43 Choi, S., and Jayasuriya, S., "A Sliding Mode Controller Incorporating Matching Conditions Applied to Manipulators," *Selected papers from the 10th Triennial World Congress of the IFAC*, Vol.4, Munich, West Germany, pp.277-282. July 1987.

44 Myszkorowski, P., "A Feedforward Sliding Controller for a Robot Manipulator," *Journal of Intelligent and Robotic Systems*, Vol.2, No.1, pp.43-52, 1989.

45 Leung, T. P., "Practical Trajectory Control of Robot Manipulator using Adaptive Sliding Control Scheme," *Proceedings of the 28th IEEE Conference on Decision and Control*, Vol.3, Tampa, FL, 13-15 Dec.1989, pp. 2647-2651.

46 Chen, Y., Wakui, S., and Mita, T., "A New Algorithm of Trajectory Control of Robot Manipulator using Sliding Mode," *Proceedings of the IEEE int. Workshop on Intelligent Robots and Systems*, 31 October - 2 November 1988, Tokyo, Japan, pp.119-124.

47 Chen, Y., Ikeda, H., Mita, T., and Wakui, S., "Trajectory Control of a Robot Ann using Sliding Mode Control," *Advanced Robotics*, Vol. 5, No.4, pp.385-395. 1991.

48 Ning-Su, L., and Chun-Bo, F., "A New Method for Suppressing Chattering in Variable Structure Control Systems," *Selected papers from the 3rd IFAC Symposium on Nonlinear Control Systems*, Capri, Italy, 1989, pp.279-284.

49 Furuta, K., "Sliding Mode Control of a Discrete System," *Systems and Control Letters*, Vol.14, No.2, pp.145-152, 1990.

50 Yeung, K. S., and Chen, Y. P., "A New Design for Manipulators Using the Theory of Variable Structure Systems," *IEEE Trans. on Automatic Control*, Vol. 33, No. 2, pp.200-206, February 1988.

51 Slotine, J.J., "Robust Control of Robot Manipulators," *International Journal of Robotics Research*, Vol. 4, No. 2, pp.49-64, summer 1985.

52 Zadeh, L.A., "Fuzzy Sets," Information Control, Vol. 8, pp.338-353, 1965.

53 Yasunobu, S., Myamoto, S., "Automatic Train Operation by Predictive Fuzzy control," *Industrial Applications of Fuzzy Control*, by Sugeno editor, Amsterdam, North Holland, 1985.

54 Nissan, "New auto Systems use Fuzzy Logic," *New York Times*, pp.23-25, July 1989.

55 Kasal, Y., and Morimoto, Y., "Electronically Controlled Continuously Variable Transmission" *Proceedings of Int. Congress on Transportation Electronics*, Dearborn, MI, 1988.

56 Mitsubishi, "First Fuzzy Logic Controlled Commercial Air Conditioner Begins Volume Shipping," News-Togai Infralogic Inc. December 1989.

57 Preprints of the Second Congress of the International Fuzzy Systems Association, Tokyo, Japan, 1987.

58 King, L E., "Fuzzy Logic Control of A Cement Kiln Precalciner Flash Furnace," *Proceedings of IEEE Conference on Applications of Adaptive and Multivariable Control*, Hull. 1982, pp.56-59.

59 Sugeno, M., and Nishida, M., "Fuzzy Control Model Car," *Fuzzy Sets and Systems*, Vol. 16, pp.103-113, North-Holland 1985.

60 Fujitec, F., "FLEX-8800 Series Elevator Group Control System," Fujitec Company, Limited, Osaka, Japan 1988.

61 Bernard, J. k., "Use of Rule-based System for Process Control," *IEEE System Magazine*, vol. 8, No.5, pp.3-13, 1988.

62 Kinoshita, M., Fukuzaki, T., Satoh, T., and Miyake, M., "An Automatic Operation Method for Control Rods in BWR Plants," *Proceedings of Specialists' Meeting on In-Core Instrumentation and Reactor Core Assessment*, Cadarache, France, 1988.

63 Yamakawa, T., "High Speed Bury Controller Hardware System," *Proceedings of 2nd Fuzzy T system Symposium*, Japan, 1986, pp.122-130.

64 Yamakawa, T., "Fuzzy Controller Hardware System," *Proceedings of 2nd International Fuzzy Systems Association Congress*, Tokyo, Japan, July 1987.

65 Yamakawa, T., "A Simple Fuzzy Computer Hardware System Employing Min and Max Operations- A Challenge to 6th Generation Computer," Proceedings of 2nd International Fuzzy Systems Association Congress, Tokyo, Japan, July 1987.

66 Aledhaibi, A., " Design of One Level Wireless Mobile LEGO Robot," Research Skill Competition, Old Dominion University, 1998.

67 Lee, C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I, II," *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 20, No. 2, pp.404-435, March/April 1990.

68 Cox, E., "Fuzzy Fundamentals," *IEEE Spectrum*, pp.58-61, October 1992.

69 Ragot, J., and Lamotte, M., "Fuzzy Logic Control," *International Journal on System Science*, Vol. 24, No. 19, pp.1825-1848, 1993.

70 Mamdani, E., "application of Fuzzy Algorithms for Simple Dynamic Plant," *Proceedings of IEE, Part D*, Vol. 121, 1974, pp.1585-1588.

71 Kim, S., and Lee, J., "Design of a Fuzzy Controller with Fuzzy Sliding Surface," *Fuzzy Set and Systems*, Vol. 71, pp.359-367, 1995.

72 Wu, J. and Liu, T., "A Sliding-mode Approach to Fuzzy Control Design," *IEEE Transaction on Control System Technology*, Vol. 4, No. 2, pp.141-150, 1996.

73 Lin, S., and Kung, C., "Linguistic Fuzzy-Sliding Mode Controller," *Proceedings of 1992 American Control Conference*" Chicago, pp.1904-1905.

74 Slotine, J., Li, W., *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

75 Mamadani, E., "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant," *Instruments of Electrical Engineering*, Vol.121, pp.1589-1588, 1974.

76 Procyk, T., and Mamdani, E "A Linguistic Self-organizing Process Controller," *Automatica*, Vol. 15, pp.15-30, 1979.

77 Linkens, L. and Hasnain, S., "Self-organizing Fuzzy Logic Control and Application to Muscle Relaxant Anesthesia," *IEE Proceedings-D*, Vol. 188, No.3, pp.274-284, 1991.

78 Daley, S., and Gill, K, "A Design Study of A Self-organizing Fuzzy Logic Controller," *Proceedings Institute of Mechanical Engineering*, Vol.200, pp.59-69, 1986.

79 Sugiyama, Rule-Based Self-organizing Controller, *Fuzzy Computing*, North- Holland, pp.341-353, 1988.

80 Zhang, B. and Edmunds, J., "Self-Organizing Fuzzy Logic Controller," *IEE Proceedings - D*, Vol. 139, No.5, pp.460-464, 1992.

81 Chung, C and Oh, J. "Autotuning Method of Membership Function in a Fuzzy Learning Controller," *Journal of Intelligent and Fuzzy Systems*, Vol. 1, pp.835-349, 1994.

82 Hung, J., Gao, W., and Hung, S., "Variable Structure Control: A Survey," *IEEE Trans. on Industrial Electronics,* Vol. 40, pp.2-21, 1998.

83 Hwang, G., and Lin, S. "A Stability Approach to Fuzzy Control Design for Nonlinear Systems," *Fuzzy Sets and Systems*, Vol. 48, pp.279-287, 1992.

84 Emami, M., Goldenberg, A., Turksen, I., "A Robust Model-Based Fuzzy Logic Controller for Robot Manipulators," *Proceedings of the 1998 IEEE International Conference on Robotics & Automation,* Leuven, Belgium, May 1998, pp2500-2505.

85 Chen, J., Liu, C., Wang, Y., "Control of Robot Manipulator Using A Fuzzy Model-Based Sliding Control Scheme," *Proceedings of the 33rd IEEE Conference on Decision and Control*, Lake Buena Vista, Florida, Vol. 4, December 1994, pp.3506-3511.

86 Kawaji, S., Matsunaga, M., "Fuzzy Control of VSS Type and Its Robustness," *Proceedings of the 3rd IFSA Congress*, Brussels, 1991, pp-81-88.

87 Ljung, L., *System Identification - Theory for the User*, Prentice Hall, Englewood Cliffs, New Jersey, 1987.

88 Juang, J.-N., *Applied system Identification*, PRT, Prentice Hall, Englewood Cliffs, New Jersey, 1987.

89 Juang, J.-N., Phan, M., "Linear System Identification via Backward Observer Models," *NASA Technical Memorandum* TM-107632, May 1992.

90 Huang, J.-K., "Linear System Identification" Lecture Notes, Old Dominion University, Norfolk, 1997.

91 Kalman, R. P., and Bucy, R. S., "New Results in Linear Filtering' and Prediction Theory," *ASME Journal of Basic Engineering,* Vol. 83, pp.95-108, March 1961.

92 Atkeson, C. 0., An, C. H., and Hollerbach, S. M., "Estimation of Inertial Parameters of Manipulator Loads and Links," *Proceedings of the Third International Symposium on Robotic Research,* 1985.

93 Hsia, T. C., "Adaptive control of Robot Manipulators-A Review," *Proceedings 'of the IEEE Conference on Robotics and Automation,* 1986, pp.183-189.

94 Khosla, P. K., and Kanada, T., "Parameter Identification of Robot Dynamics," *Proceedings of the IEEE Conference on Decision and Control,* 1985, pp.4754-1760.

95 Ljung, L., and' Soderstrom, T., *Theory and Pra6tice of Recursive Identification,* Cambridge, Mass; MIT Press, 1983.

96 Middleton, R.H., and Goodwin, G.C., "Adaptive Computed Torque Control for Rigid Link Manipulators," *Proceedings of the IEEE Conference on Decision and Control,* 1986, pp.68-73.

97 Nicosia, S., and Tomei, P., "Robot Control by Using Only Joint Position Measurements," *IEEE Trans. on Automatic Control,* Vol. AC-35, pp.1058-1061, September 1990.

98 Ortega, R., and Spong, M. W., "Adaptive Motion Control of Rigid Robots: A Tutorial," *Proceedings of the IEEE Conference on Decision and Control,* 1988, pp.1575-1584.

99 Reed, S. S., and Ioannou, P. A., "Instability Analysis and Robust Adaptive Control of Robotic Manipulators," *Proceedings of the IEEE Conference on Decision and Control,* 1988, pp.1607-1612.

100 Schwartz, H. M. , Warshaw, 0., and Janabi, T., "Issues in Robot Adaptive13 Control," *Proceedings of the American Control Conference,* 1990.

101 Slotine, S. E., and Li, W., "Adaptive Manipulator Control: A Case Study," *IEEE Trans. on Automatic Control,* Vol. AC-33, pp.995-1003. November 1988.

102 Thornton, C. L., and Bierman, 0.5., "Gram-Schmidt Algorithms for Covariance Propagation," *Proceedings of the IEEE Conference on Decision and Control,* 1975, pp.489-498.

103 Aledhaibi, A., Huang, J.-K., "Design Of Adaptive Sliding Mode Fuzzy Control For Robot Manipulator Based On Extended Kalman Filter," *STCEX 2000,* November 2000, Riyadh, Saudi Arabia.

104 Craig, S., Hsu, P., and Sastry, S.., "Adaptive Control of Mechanical Manipulators," *The International Journal of Robotics Research,* Vol.6, pp.49-59.1987.

CURRICULM VITA
for
Abdelrahman Aledhaibi

NAME: Abdelrahman M. Aledhaibi
DATE OF BIRTH: November 15, 1962

DEGREES:

Doctor of Philosophy (Mechanical Engineering), Old Dominion University, Norfolk, VA , August 2000.
Master of Science (Mechanical Engineering), Murray State University, Murray, KY, December 1995.
Bachelor of Science (Mechanical Engineering), Southern Illinois University, Carbondale, IL , May 1992.

PROFESSIONAL CHRONOLOGY:

Department of Mechanical Engineering, Old Dominion University, Norfolk, Virginia.
Instructor, January 1998- Present

COURSES TAUGHT DURING LAST FIVE YEARS:

Murray State University:
ME560: Intro. to Electro-hydraulic Motion Control, ME550: Into. to Robotics, ME260: Dynamic, ME250: Static, ME261: Fluid Power.

Old Dominion University:
ME496/692: Manufacturing Process Auto. and Control (Robotics), ME438/538: Control System Design, ENGR110/111: Freshman Engineering.

SCHOLARLY ACTIVITIES COMPLETED:

Refereed Conference Articles

1. Aledhaibi, A., Huang, J.-K., "Design of Adaptive Sliding Mode Fuzzy Control for Robot Manipulators," *SES98*, September 1998, Pullman, WA.
2. Aledhaibi, A., Abdelmotagaly, K., Huang, J.-K., "Design of Self-Organizing Fuzzy Control for Robot Manipulator," *JCIS'98*, October 1998, Research Triangle Park, NC.
3. Aledhaibi, A., Huang, J.-K., "Design of Adaptive Sliding Mode Fuzzy Control for Robot Manipulator Based on Extended Kalman Filter," *STCEX 2000*, November 2000, Riyadh, Saudi Arabia.

Refereed Journal Articles

1. Aledhaibi, A., Huang, J.-K., "Design of Adaptive Sliding Mode Fuzzy Control for Robot Manipulator Based on Extended Kalman Filter," *Arabian Journal for Science and Engineering* (to be published)