AN ABSTRACT OF THE THESIS OF

JUNKU YUH for the degree of DOCTOR OF PHILOSOPHY

in Mechanical Engineering presented on March 12, 1986.

Title: DISCRETE-TIME EXPLICIT MODEL REFERENCE ADAPTIVE CONTROL FOR ROBOTIC MANIPULATORS

Redacted for privacy

Abstract approved: ______________________

Dr. William E. Holley

In this dissertation a direct approach to discrete-time model reference adaptive control (MRAC) based on hyperstability theory is proposed to control industrial robotic manipulators.

For industrial robots and manipulators, which usually have highly nonlinear and complex dynamic equations and often have unknown inertia characteristics, it is very difficult to achieve high performance with conventional control strategies. This desired high performance in terms of speed and accuracy can be obtained by adaptive control techniques.

Considering the effects of gravity, process noise and payload uncertainty the MRAC approach is investigated

using simulation for a three degree of freedom industrial robot. These simulation results show that adaptive control techniques can provide robust properties in spite of poor a priori information regarding the robot dynamics and operating circumstances.

Discrete-time Explicit Model Reference Adaptive Control for Robotic Manipulators

by

Junku Yuh

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed March 12, 1986

Commencement June 1986

APPROVED:

Redacted for privacy

Professor of Mechanical Engineering in charge of major

Redacted for privacy

Head of Department of Mechanical Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented March 12, 1986

ACKNOWLEDGEMENTS

I am deeply indebted to Dr. William E. Holley for all the help and guidance that I received from him during my graduate education at OSU. Without his consistant support, this work would not have been possible.

Gratitude is also due to Dr. James R. Welty, chairman of the department of Mechanical Engineering for continuous financial aid and advice, and to all the staff members of the department for the great opportunity and helpful assistance they provided.

I also would like to thank Dr. Charles E. Smith for the excellent course he taught in Advanced Dynamics and his constant interest in my work, and Dr. Ronald B. Guenther for his valueable advice during my years at OSU. It is a pleasure to thank Dr. Timothy C. Kennedy for his encouragement, and Dr. Harold I. Laursen for taking his time to serve on my committee. I appreciate the offering of Mr. John Hartin for suggesting better expressions in English.

I owe my parents special graditude for their support and absolute love toward me, and my fiancee, Jong-il for her patience and sincere encouragement.

I know not how to express my grateful thanks to the Lord who created this world.

TABLE OF CONTENTS

LIST OF FIGURES

DISCRETE-TIME EXPLICIT MODEL REFERENCE ADAPTIVE CONTROL FOR ROBOTIC MANIPULATORS

I. INTRODUCTION

In the present work a direct approach using a discrete-time model reference adaptive control (MRAC) system based on hyperstability theory is described in order to control industrial robotic manipulators.

When the parameters of the plant dynamic model (in this case a robotic manipulator) and the associated disturbances are poorly known or vary in time, it is very difficult to achieve high performance with conventional control strategies. A more robust and sophisticated control scheme which gives high performance in spite of poor information regarding the plant and operating circumstances is therefore desired. It has been reported that adaptive control techniques can provide this robust property in several applications [1-3].

The main problem of adaptive control schemes is to find a proper way to adjust the controller parameters to follow changes in the plant parameters and disturbances without losing stability of system. Depending on the method for adjusting the parameters of the controller, various approaches to adaptive control can be classified in the following three schemes: gain scheduling [4,5], model reference adaptive control [6-9], and self-tuning regulators [10,11]. Among these the model reference adaptive control method is discussed here. This scheme was originally introduced by Whitaker, Yamron and Kezer(1958) [12] for the servo problem, and its objective is to find an adaptation mechanism which assures that the difference between the output of the reference model and the output of the plant tends to zero as time goes to infinity for any initial conditions.

From the stability point of view, two basic approaches to the design of MRAC systems can be considered. The first approach is to use a suitable Lyapunov function and derive an adaptation mechanism assuring the global asymptotic stability for the whole system. This approach can be found in the earlier works of Park and others [13-17]. The second approach is to use the hyperstability and positivity concepts of Popov [18] which provide sufficient stability conditions for a feedback system which can be formed by a linear time-invariant feedforward block and a nonlinear time-varying feedback block. The error equation of the MRAC system can be represented by this equivalent feedback system. Thus the adaptation structure is determined by the stability conditions for this equivalent feedback system. This approach has been proposed by several authors [19-21]. In the present work the second approach is used since it has been shown by Landau [22] that the use of hyperstability theory gives a larger class of adaptation algorithms than is obtained when using the direct Lyapunov method.

In the last few years, several industrial robots and manipulators have been introduced as applications of the MRAC approach [23-27]. Conventional linear control techniques usually give only limited dynamic performance for robotic manipulators. This poor performance is due to the highly nonlinear and complex dynamic equations which often have unknown inertia characteristics. In most previous work, continuous-time MRAC schemes for robotic manipulators have been proposed. However, since practical implementation of the manipulator control is performed on a digital computer which is low-cost and reliable, a discrete-time MRAC scheme is desired. One approach for a discrete-time MRAC for manipulators has been proposed by R. Horowitz [28]. In his work, only the linear terms for the manipulator were considered, allowing simplified dynamic equations to be obtained. The discrete-time MRAC system was then developed using only velocity feedback. The position feedback was implemented in a conventional way. Using this approach, step-responses for the system were reported. In the present work, the complete nonlinear manipulator dynamic equations are considered and both position and velocity feedback are utilized in the MRAC system. Considering the effects of gravity, process noise and payload uncertainty the MRAC scheme is investigated using computer simulation. Also the reference input is computed using a trapezoidal speed law along a desired trajectory.

The remaining chapters in the thesis are organized as follows: In chapter 2 the definitions and basic theorems of stability theory, the concept of a positive system, and hyperstability theory are introduced. In chapter 3 a parameter adaptation algorithm (PAA) based on hyperstability theory is determined for single-input single-output plant systems and discussed in terms of stability. In order to provide more understanding of this PAA, an adaptation scheme based on a Lyapunov function is compared to the adaptation scheme based on hyperstability theory. In chapter 4 a linear model following control (LMFC) system for tracking and regulation of linear plants with known parameters is determined. Based on the structure of the liner model following control law, a MRAC system is determined using the PAA of chapter 3 and is studied with a time-varing adaptation gain. In chapter 5 a three degree of freedom industrial robot similar to the SEIKO Model PN-700 is modelled. The discrete-time MRAC obtained in chapter 4 is used to control this manipulator with a given trajectory and speed law. Considering the effects of gravity, process noise and payload uncertainty, this approach is investigated by simulation. Chapter 6 summarizes this work and gives some comments on directions for further research.

II. THEORETICAL BACKGROUND

2.1 Introduction

Adaptive control systems have been introduced to achieve high control performance when the plant dynamic characteristics are poorly known or when large and unpredictable variations occur.

To answer the question introduced by Landau [29] regarding the difference between conventional feedback control and adaptive control, we propose that adaptive control is focused on the elemination of the effects of parameter variations in the plant to be controlled, while conventional feedback control is focused on the elemination of the effect of modeled disturbances. For example, a D.C. servo motor used in robotic applications is subject to important variations in the load moment of inertia. Thus to achieve high performance in such a system, adaptive control techniques are attractive.

Among the various types of adaptive control techniques, we consider the model reference adaptive control (MRAC) systems. The main hypothesis for MRAC systems is that in order to achieve desired performance when plant parameters vary, only the values of the parameters of the controller have to be changed and not the structure of the controller.

Basically a MRAC system is formed by:

(1) A reference model which gives the desired performance.

(2) A plant whose performance should be as close as possible to that of the reference model.

(3) A differencing element which forms the error between the output of the reference model and of the plant.

(4) An adaptation mechanism which processes the error in order to modify the parameters of the controller.

Figure 2.1.1 shows the basic structure of the MRAC system.

A MRAC system can be equivalently represented as a nonlinear time-varying feedback system which will be shown in chapter 3. Thus we can use results on the stability of this type of feedback system for the design of the adaptation algorithm. In the work presented here, the results of hyperstability theory whose concept was first

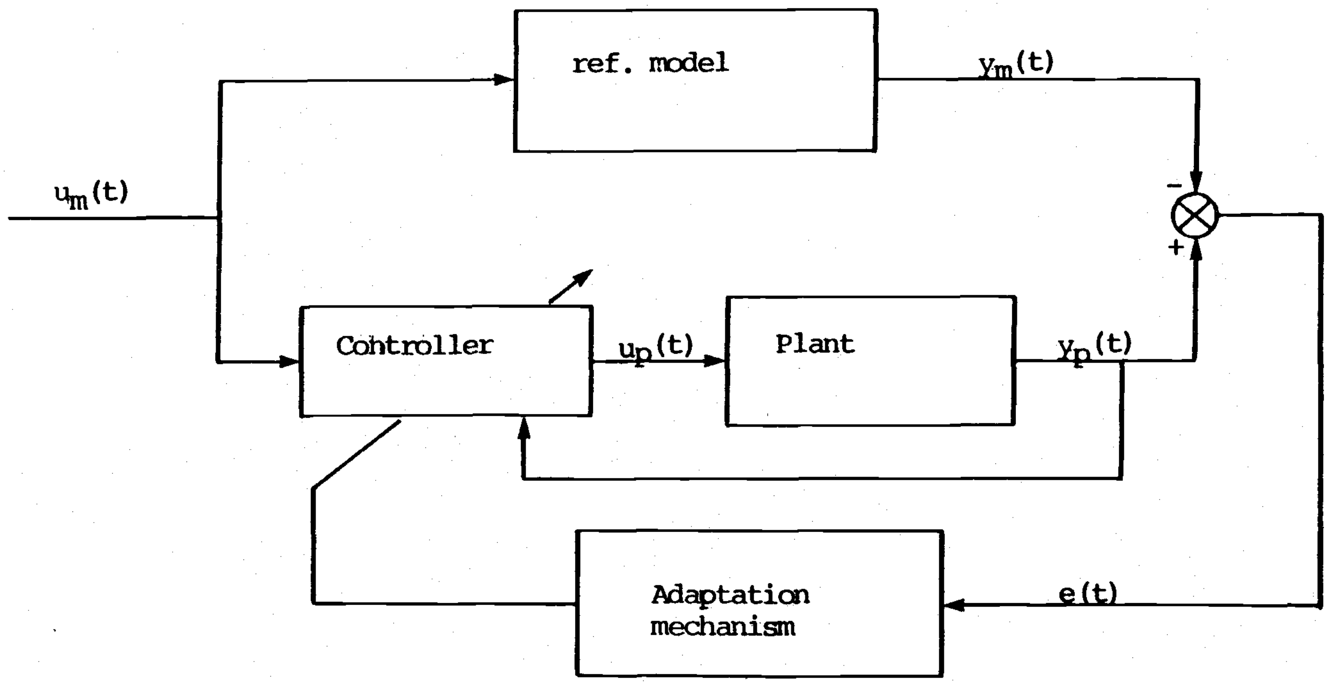

Figure 2.1.1 Basic Structure of a MRAC System

introduced by Popov [30] is used for the design of the adaptation algorithm.

The concept of hyperstability theory is closely related to the positivity concept. Thus in section 2.2, 2.3, and 2.4 of this chapter, we will discuss some definitions and basic theorems of the following subjects:

(1) System stability

(2) Concept of a positive system

(3) Hyperstability

The reader should realize that only the discrete time version of each theorem will be included here since our interest lies solely with discrete-time systems. For readers interested in a deeper study of the above subjects, references are indicated in each section.

2.2 System Stability

Stability refers to the boundedness of the variables of a system as time goes to infinity. There are two major points of view with respect to the stability. First, the system is presumed to possess an equlibrium state and the concern is with the ability of the system to maintain its state in the vicinity of this equlibrium in the absence of any inputs. Another point of view concerns the behavior of the state variables when the system is subjected to a bounded input.

In this section a Lyapunov function is defined and the stability of discrete-time systems is described. A basic reference for this section is Freeman [31].

Consider a sampled-data system to which no input is applied:

$$x(t_{k+1}) = F(x(t_k), t_k) \qquad (2.2.1)$$

where x is the n-dimensional vector, t_k is the independent, discrete-time variable, $t_{k+1} > t_k$ for all integer k, $-\infty \le k \le \infty$, and $t \to \infty$ as $k \to \infty$.

Consider the vector fuction $P(t_k,x_o,t_o)$, a solution of equation (2.2.1) for an initial state x_o and an initial time t_o :

$$x(t_k) = P(t_k,x_o,t_o) \quad \text{for all } t_k \geq t_o \tag{2.2.2}$$

We assume that for the systems under consideration the vector function P is always continuous in all its arguments. Clearly,

$$P(t_o,x_o,t_o) = x_o \tag{2.2.3}$$

$$P(t_a,x_c,t_c) = P(t_a,P(t_b,x_c,t_c),t_b) \quad \text{for all } t_a \geq t_b \geq t_c \tag{2.2.4}$$

A state x_e of the system given by equation (2.2.1) is called an equlibrium state if

$$F(x_e,t_k) = x_e \quad \text{for all } t_k \tag{2.2.5}$$

Definition 2.2.1

The equlibrium state x_e is said to be L-stable (stable in the sense of Lyapunov) if, for any t_o and any $\varepsilon > 0$, there corresponds a $\delta(\varepsilon, t_o) > 0$ such that if

$||x_o - x_e|| < \delta(\varepsilon, t_o)$,

then $||P(t_k,x_o,t_o) - x_e|| < \varepsilon$ for all $t_k \geq t_o$.

Definition 2.2.2

The equlibrium state x_e is said to be uniformly L-stable if, for any t_o and any $\varepsilon > 0$, there corresponds a $\delta(\varepsilon) > 0$

such that if $||x_o - x_e|| < \delta(\varepsilon)$, then $||P(t_k, x_o, t_o) - x_e|| < \varepsilon$ for all $t_k \geq t_o$.

Definition 2.2.3

The equlibrium state x_e is said to be asymptotically stable if it is L-stable and if there exists a $\eta(t_o) > 0$ such that

$$\lim_{k \to \infty} ||P(t_k, x_o, t_o) - x_e|| = 0 \text{ for all } ||x_o - x_e|| < \eta(t_o) \tag{2.2.6}$$

If η is independent of t_o, the state is uniformly asymptotically stable.

Definition 2.2.4

The equilibrium x_e is said to be globally asymptotically stable if it is asymptotically stable for any initial state x_o.

Definition 2.2.5

A scalar function $V(x, t_k)$ is said to be positive definite in a neighborhood N of the point x_e if $V(x_e, t_k) = 0$ and if there exists a continuous nondecreasing, scalar function W such that

$$W(0) = 0 \tag{2.2.7}$$

$$V(x,t_k) \geq W(||x||) \text{ for all } t_k \text{ and all } x \text{ in } N \quad (2.2.8)$$

Definition 2.2.6

A positive definite function $V(x,t_k)$ is said to be decrescent in a neighborhood N if there exists a continuous nondecreasing scalar function S such that

$$S(0) = 0 \quad (2.2.9)$$

$$V(x,t_k) \leq S(||x||) \text{ for all } t_k \text{ and all } x=x_e \text{ in } N \quad (2.2.10)$$

Definition 2.2.7

A positive definite function $V(x,t_k)$ is said to be infinitely large if $|V(x,t_k)| \to \infty$ as $||x|| \to \infty$ for all t_k.

Let $V(x,t_k)$ be a positive definite function with continuous first partial derivatives with respect to x. Let $V(x,t_k)$ denote the first forward difference in $V(x,t_k)$ along the positive time axis, that is,

$$V(x,t_k) = \frac{V(x(t_{k+1}),t_{k+1}) - V(x(t_k),t_k)}{t_{k+1} - t_k} \quad (2.2.11)$$

when $x(t_k) = x$

Consider the system given by equation(2.2.1) where

$$F(0,t_k) = 0 \quad (2.2.12)$$

that is, an equlibrium state x_e=0.

Theorem 2.2.1

The equlibrium state $x_e=0$ is L-stable if there exists a positive definite function $V(x,t_k)$ possessing a nonpositive forward difference $\Delta V(x,t_k)$.

Proof: We refer to definitions 2.2.1 and 2.2.5. Given a particular $\varepsilon>0$, we select a $\delta(\varepsilon,t_o)>0$ such that for $||x_o||<\delta(\varepsilon,t_o)$ we obtain both $||x_o||<\varepsilon$ and $V(x_o,t_o)<W(\varepsilon)$. Such a choice of δ is possible because of the continuity in x of $V(x\ ,t_k)$. Since $V(x,t_k)$ is nonpositive,

$$V(x_o,t_o) \geq V(P(t_k,x_o,t_o),t_k) \tag{2.2.13}$$

$$\geq W(||P(t_k,x_o,t_o)||) \tag{2.2.14}$$

and therefore

$$W(\varepsilon) > V(x_o,t_o) \geq W(||P(t_k,x_o,t_o)||) \tag{2.2.15}$$

Since, however, W is a nondecreasing function, it follows that

$$||P(t_k,x_o,t_o)|| < \varepsilon \text{ for all } t_k>t_o \text{ and all } ||x_o||<\delta(\varepsilon,t_o) \tag{2.2.16}$$

Q.E.D.

Theorem 2.2.2

The equlibrium $x_e=0$ is asymptotically stable if there exists a decrescent positive definite function $V(x,t_k)$

possessing a negative definite forward difference $\Delta V(x,t_k)$.

Proof: From the proof of the L-stablility theoreom 2.2.1 we know that the positive definite function $V(x,t_k)$ has a non-negative limit as $t_k \to \infty$. We denote this limit by V_L. Since $V(x,t_k)$ is decrescent by hypothesis,

$$V(x,t_k) \leq S(||x||) \qquad (2.2.17)$$

Hence $V_L > 0$ implies that the state magnitude of $||x(t_k)||=||P(t_k,x_o,t_o)||$ will always larger than some positive number μ. Since $\Delta V(x,t_k)$ is negative definite,

$$\Delta V(x,t_k) \leq -r(||x||) \qquad (2.2.18)$$

where r is a continuous, nondecreasing scalar function such that r(0)=0, and where we have for simplicity assumed that $t_{k+1}-t_k=1$ in equation (2.2.11). Then $V_L>0$ implies

$$\Delta V(x,t_k) \leq -r(\mu) < 0 \qquad (2.2.19)$$

We now write $V(x,t_k)$ in terms of its forward difference $\Delta V(x,t_k)$:

$$V(x(t_k),t_k) = V(x_o,t_o) + \sum_{i=0}^{k-1} V(x(t_i),t_i)$$
$$\leq V(x_o,t_o) - kr(\mu) \qquad (2.2.20)$$

Since $V(x,t_k)$ is positive definite, the right-hand side of equation (2.2.20) may not become negative. The only way this can be satisfied for large k is to have $r(\mu)=0$. Hence

$\mu=0$ and $||P(t_k,x_o,t_o)||\to\infty$ as $K\to\infty$.

Q.E.D.

Theorem 2.2.3

The equlibrium state $x_e=0$ is globally asymptotically stable if it is asymptotically stable and if $V(x,t_k)$ is such that $W(||x||)\to\infty$ as $||x||\to\infty$.

Proof of theorem 2.2.3 is not included here since it follows directly from the proofs of the foregoing theorems.

We note that the stability (in the sense of Lyapunov) of an equlibrium point depends on the existence of a positive definite function $V(x,t_k)$ possessing a nonpositive first difference $\Delta V(x,t_k)$. This function is referred to as a Lyapunov function. It is clear that this function will normally not be unique for a given equlibrium point.

Theorem 2.2.4

The equlibrium state $x_e=0$ of the linear discrete-time autonomous dynamic system:

$$x(k+1) = Ax(k) \tag{2.2.21}$$

is asymptotically stable if and only if given any positive definite matrix Q there exists a symmetric positive definite matrix P which is the unique solution of the matrix equation

$$A^TPA - P = -Q \tag{2.2.22}$$

and $V(x) = x(k)^TPx(k)$ (2.2.23)

is a Lyapunov function for the system of equation (2.2.21).

Proof: Let Q be a symmetric positive definite matrix such that

$$\Delta V(x(k)) = -x(k)^TQx(k) \tag{2.2.24}$$

Then $\Delta V(x(k)) = V(x(k+1)) - V(x(k))$

$$= V(Ax(k)) - V(x(k)) \tag{2.2.25}$$

Substituting a Lyapunov function (2.2.23) and making use of the fact that $(Ax(k))^T = x(k)^TA^T$, we find

$$x(k)^TA^TPAx(k) - x(k)^TPx(k) = -x(k)^TQx(k) \tag{2.2.26}$$

and hence

$$A^TPA - P = -Q \tag{2.2.27}$$

If we can solve equation (2.2.27) for the matrix P and then P is a symmetric positive definite matrix, the system (2.2.21) will be asymptotically stable by theorem 2.2.2.

Q.E.D.

Now we investigate the stability of linear discrete-time systems which have a non-zero input. Since we shall permit a nonzero input for all future time, we must modify our definition of stability to take into account a boundness condition on the input. Roughly speaking, we shall regard a system as stable if a bounded input produces a bounded output. The following definition which is applicable to continuous-time and discrete-time systems is introduced.

Definition 2.2.8

A linear system is stable if and only if at any time t , with the system in any initial state $x(t_o)$, every input u that satisfies the condition $||u(t)||<\eta_1$ for all $t_o<t<\infty$, yields a state x and an output y such that $||x(t)||<\eta_2$ and $||y(t)||<\eta_3$ for all $t_o<t<\infty$ where η_1, η_2 and η_3 are positive finite constants.

2.3 Concept of a Positive System

The concept of positive systems was introduced by Popov, Kalman, and Yakubovitch. This concept will be used to develop hyperstability theory in section 2.4. Basic references for this section are [32-34].

Consider the linear discrete-time system

$$x(k+1) = Ax(k) + Bu(k) \tag{2.3.1}$$

$$y(k) = Cx(k) + Du(k) \tag{2.3.2}$$

where x is an n-dimensional state vector, u and y are m-dimensional vectors representing the input and the output, respectively and A, B, C and D are matrices of appropriate dimension. We assume that (A,B) is completely controllable and that (C,A) is completely observable. The system of equations (2.3.1) and (2.3.2) is also characterized by the discrete square transfer matrix,

$$H(z) = D + C(zI - A)^{-1}B \tag{2.3.3}$$

Definition 2.3.1

An m x m discrete matrix H(z) of real rational functions is positive real if

1. All elements of H(z) are analytic outside the unit circle, that is, they do not have poles in $|z|>1$.

2. The eventual poles of any element of $H(z)$ on the unit circle $|z|=1$ are simple, and the associated residue matrix is a positive semidefinite Hermitian.

3. The matrix

$$F(w) = H(e^{jw}) + H(e^{-jw})^T \qquad (2.3.4)$$

is a positive semidefinite Hermitian for all real values of w which are not a pole of any element of $H(e^{jw})$, that is, for all z on the unit circle $|z|=1$ which are not a pole of $H(z)$.

Definition 2.3.2

An $m \times m$ discrete matrix $H(z)$ of real rational functions is strictly positive real if

1. all the elements of $H(z)$ are analytic in $|z|\geq 1$.
2. The matrix

$$F(w) = H(e^{jw}) + H(e^{-jw})^T \qquad (2.3.4)$$

is a positive definite Hermitian for all w, that is, for all z on the unit circle $|z|=1$.

Definition 2.3.3

The discrete matrix kernal $c(k, i)$ is termed positive definite if for each interval $[k_0,k_1]$ and for all the discrete vectors $f(k)$ bounded in $[k_0,k_1]$ the following inequality holds:

$$\sum_{k=k_0}^{k_1} f(k)^T[\sum_{i=k_0}^{k} c(k,i)f(i)] \geq 0 \text{ for all } k_1 \geq k_0 \qquad (2.3.5)$$

Definition 2.3.4

The system of equations (2.3.1) and (2.3.2) is positive if the sum of the input-output scalar products over the interval $[0,k_1]$ can be expressed by

$$\sum_{k=k_0}^{k_1} y(k)^T u(k) = \zeta(x(k_1+1)) - \zeta(x(0)) + \sum_{k=0}^{k} \lambda(x(k),u(k)) \qquad (2.3.6)$$

for all $k_1 \geq 0$ with $\lambda(x(k),u(k)) \geq 0$ for all $x(k) \in R^n$, $u(k) \in R^m$. Some of the equivalent formulations of the properties of the linear discrete-time positive systems are given in the following theorem 2.3.1. These properties are necessary and sufficient conditions for a linear discrete-time system to be positive.

Theorem 2.3.1

The following properties concerning the system of equations (2.3.1) and (2.3.2) are equivalent to each other:

1. The system of equations (2.3.1) and (2.3.2) is positive (Definition 2.3.4).
2. H(z) given by equation (2.3.3) is a positive real discrete transfer matrix.

3. There exist a symmetric positive definite matrix P, a symmetric positive semidefinite matrix Q, and matrices S and R such that

$$A^T PA - P = -Q \tag{2.3.7}$$

$$B^T PA + S^T = C \tag{2.3.8}$$

$$D + D^T - B^T PB = R \tag{2.3.9}$$

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0 \tag{2.3.10}$$

4. [Kalman-Szego-Popov Lemma] There exist a symmetric positive definite matrix P and matrices K and L such that

$$A^T PA - P = -LL^T \tag{2.3.11}$$

$$B^T PA + K^T L^T = C \tag{2.3.12}$$

$$K^T K = D + D^T - B^T PB \tag{2.3.13}$$

5. Every solution x(k) of equations (2.3.1) and (2.3.2) satisfies the following equality:

$$\sum_{k=0}^{k_1} y(k)^T u(k) = \frac{1}{2} x(k_1+1)^T Px(k_1+1) - \frac{1}{2} x(0)^T Px(0) + \frac{1}{2} \sum_{k=0}^{k_1} [x(k)^T Qx(k) + 2u(k^T S^T x(k) + u(k)^T Ru(k)] \tag{2.3.14}$$

where P is a positive definite matrix and the matrices P, Q, S, and R satisfy equations (2.3.7) and (2.3.10).

6. [Impulse response matrix] The discrete matrix kernal,

$$c(k-i) = D\delta(k-i) + CA^{-(i-1)} B\mu(k-i) \tag{2.3.15}$$

where $\mu(k)$ means a unit step function, C is a positive definite discrete kernel.

Proof of theorem 2.3.1 will be shown in Appendix A.

Lemma 2.3.1

The discrete transfer matrix H(z) given by equation (2.3.3) is strictly positive real if there exists a symmetric positive definite matrix P, a symmetric positive definite matrix Q, and matrices K and L such that

$$A^TPA - P = -LL^T - Q \tag{2.3.16}$$

$$B^TPA + K^TL^T = C \tag{2.3.17}$$

$$K^TK = D + D^T - B^TPB \tag{2.3.18}$$

Proof of lemma 2.3.1 is not included here but using the continuous version [35] of the strictly positive real lemma, it can be proven by the same procedure of the proof of proposition 4 of theorem 2.3.1, which is shown in Appendix A.

2.4 Hyperstability

Let us consider the feedback system represented in Figure 2.4.1 which is formed by a linear time-invariant feedforward block and a nonlinear time-varying feedback block. When a feedback system shown in Figure 2.4.1 is globally stable for all the feedback blocks satisfying the inequality:

$$n(k_0,k_1) \triangleq \sum_{k=0}^{k_1} w(k)^T y(k) \geq -r_0^2 \tag{2.4.1}$$

where r_o^2 is a finite positive constant, this feedback system will be said to be hyperstable.

In this section, hyperstability is defined and basic results of hyperstability theory are introduced. Basic references for this section are [36-38].

Consider the closed-loop system having a feedforward block:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.4.2}$$

$$y(k) = Cx(k) + Du(k) \tag{2.4.3}$$

and a feedback block:

$$u(k) = -w(k) \tag{2.4.4}$$

$$w(k) = f(y,k) \tag{2.4.5}$$

where x is an n-dimensional state vector, u and y are m-

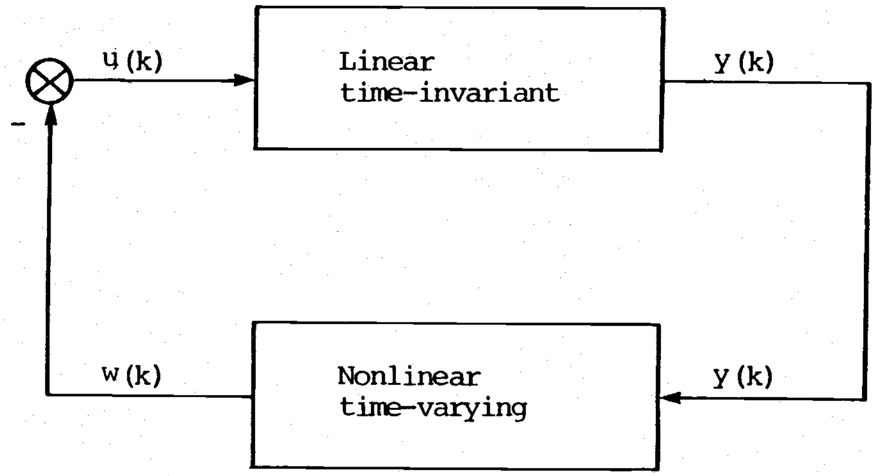

Figure 2.4.1 Nonlinear time-varying feedback system

dimensional input and output vectors respectively, A, B, C and D are matrices of appropriate dimension, (A,B) is completely controllable, (C,A) is completely observable, and f is a functional-vector.

Definition 2.4.1

The closed-loop system of equations (2.4.2) to (2.4.5) is hyperstable if there exists a positive constant $\delta>0$ and a positive constant $r_0>0$ such that all the solutions $x(x(0),k)$ of equations (2.4.2) and (2.4.3) satisfy the inequality

$$||x(k)|| < \delta[||x(0)|| + r_0] \quad \text{for all } k \geq 0 \qquad (2.4.6)$$

for any feedback block of equation (2.4.5) satisfying the inequality of equation (2.4.1).

Definition 2.4.2

The closed-loop system of equation (2.4.2) to (2.4.5) is asymptotically hyperstable if it is globally asymptotically stable for all the feedback blocks given by equation (2.4.5) which satisfies the inequality of equation (2.4.1).

Theorem 2.4.1

The necessary and sufficient condition for the feedback system described by equations (2.4.1) to (2.4.5) to be (asymptotically) hyperstable is as follows:

The discrete transfer matrix

$$H(z) = D + C(zI - A)^{-1}B \qquad (2.4.7)$$

must be a (strictly) positive real discrete transfer matrix.

Proof of theorem 2.4.1 can be found in ref. [39].

III. THE PARAMETER ADAPTATION ALGORITHM

3.1 Introduction

In this chapter, a parameter adaptation algorithm (PAA) is determined and its stability is discussed. This discussion centers on the problem of output prediction since the feedback structure for this system is simpler than the structure of the model reference adaptive control (MRAC) system. This simplicity allows a clearer understanding of the stability properties of the hyperstability and positivity concepts. The application to the MRAC problem will be discussed in chapter 4.

In section 3.2 a simple PAA is determined using hyperstability theory with the following procedure:

(1) Define the plant equation and an adjustable predictor equation.

(2) From (1) define the error equation between the output of the plant and the output of predictor.

(3) Determine the equivalent feedback system for the error equation, which has a time-invariant feedforward block and a time-varying feedback block.

(4) Using hyperstability theory and the positivity concept, determine a PAA giving a feedback system which is globally asymptotically stable.

In section 3.3 the general PAA is introduced and analyzed from the stability point of view. This PAA will be used to develop the explicit MRAC system in chapter 4.

In section 3.4 a PAA is determined using a Lyapunov function and is compared to the PAA of section 3.2. This approach is described in order to help in understanding the stability properties of the PAA. The advantages of the hyperstability and positivity concepts are readily contrasted with the direct Lyapunov approach for this prediction problem.

3.2 PAA based on Hyperstability Theory

In this section the structure of the adjustable predictor is defined and then a PAA is developed so that the adaptation error is globally asymptotically stable for any initial conditions.

We will consider the following dynamic model for the plant:

$$y(k+1) = A(q^{-1})y(k) + B(q^{-1})u(k) \qquad (3.2.1)$$

where y is a scalar output, u is a scalar input, A and B are the scalar operators involving time delays and q^{-1} is a single time step delay operator, and the a posteriori output of the adjustable predictor can be described by:

$$\hat{y}(k+1) = \hat{A}(k+1,q^{-1})\hat{y}(k) + \hat{B}(k+1,q^{-1})u(k) \qquad (3.2.2)$$

where

$$\hat{A}(k,q^{-1}) = \hat{a}_1(k) + \ldots\ldots + \hat{a}_n(k)q^{-n+1} \qquad (3.2.3)$$

$$\hat{B}(k,q^{-1}) = \hat{b}_o(k) + \ldots\ldots + \hat{b}_m(k)q^{-m} \qquad (3.2.4)$$

The a posteriori output of the adjustable predictor of equation (3.2.2) can be rearranged so that

$$\hat{y}(k+1) = \hat{\Theta}(k+1)^T\emptyset(k) \qquad (3.2.5)$$

where

$$\hat{\Theta}(k)^T = [\hat{a}_1 \; \ldots \; \hat{a}_n \; \hat{b}_o \; \ldots \; \hat{b}_m] \tag{3.2.6}$$

$$\emptyset(k)^T = [\hat{y}(k) \; \ldots \; \hat{y}(k-n+1) \; u(k) \; \ldots \; u(k-m)] \tag{3.2.7}$$

This form gives the structure of a parallel model reference adaptive system which is shown in Figure 3.2.1.

The a priori output of the adjustable predictor will be given by:

$$\hat{y}_o(k+1) = \hat{\Theta}(k)^T \emptyset(k) \tag{3.2.8}$$

A priori here refers to conditions priori to the error feedback adjustment to the predictor. Thus, the a priori prediction error is defined by:

$$\begin{aligned} e_o(k+1) &\triangleq y(k+1) - \hat{y}_o(k+1) \\ &= y(k+1) - \hat{\Theta}(k)^T \emptyset(k) \end{aligned} \tag{3.2.9}$$

and the a posteriori prediction error is defined by:

$$\begin{aligned} e(k+1) &\triangleq y(k+1) - \hat{y}(k+1) \\ &= y(k+1) - \hat{\Theta}(k+1)^T \emptyset(k) \end{aligned} \tag{3.2.10}$$

Using equations (3.2.1) and (3.2.5), we can rewrite the equation (3.2.10):

$$\begin{aligned} e(k+1) &= A(q^{-1})y(k) - \hat{A}(k+1,q^{-1})\hat{y}(k) \\ &\quad + [B(q^{-1}) - \hat{B}(k+1,q^{-1})]u(k) \\ &= A(q^{-1})e(k) + [A(q^{-1}) - \hat{A}(k+1,q^{-1})]\hat{y}(k) \\ &\quad + [B(q^{-1}) - \hat{B}(k+1,q^{-1})]u(k) \end{aligned} \tag{3.2.11}$$

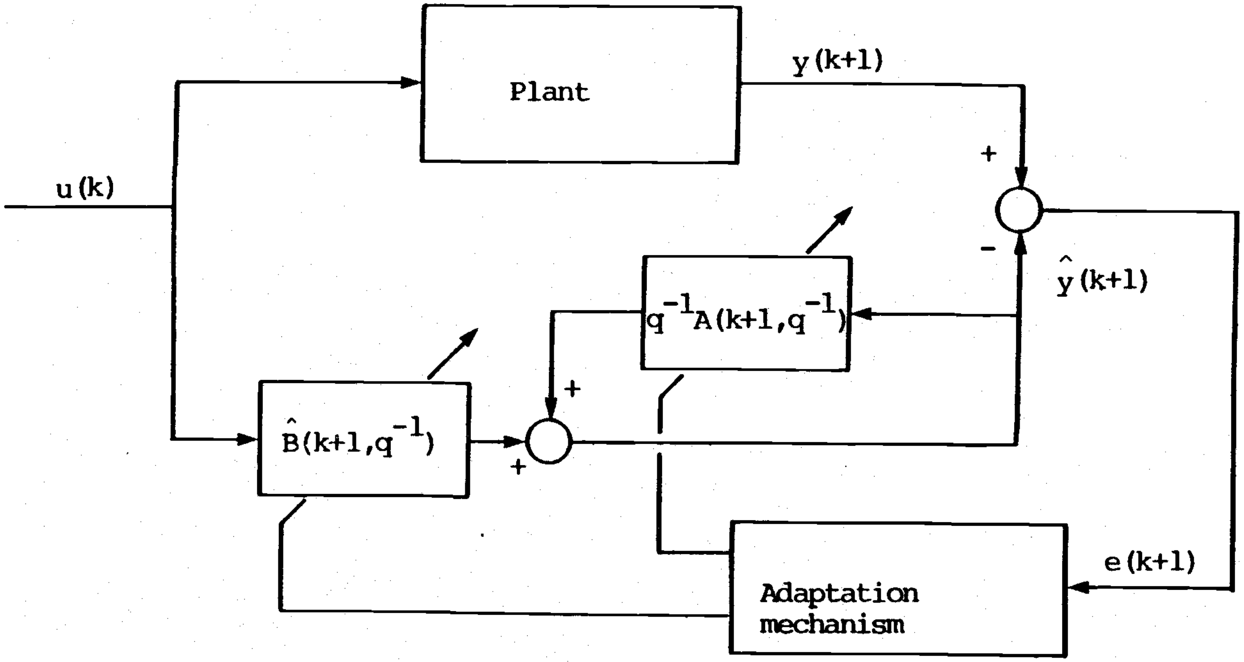

Figure 3.2.1 Structure of Parallel MRAC System

Thus, the a posteriori prediction error equation is given by:

$$
\begin{aligned}
e(k+1) &= A(q^{-1})e(k) - [\hat{\theta}(k+1) - \theta]^T \phi(k) \\
&= A(q^{-1})e(k) - \tilde{\theta}(k+1)^T \phi(k) \\
&= -\frac{1}{[1-q^{-1}A(q^{-1})]}\tilde{\theta}(k+1)^T \phi(k) \qquad (3.2.12)
\end{aligned}
$$

where

$$\theta^T = [a_1 \ \ldots \ a_n \ b_o \ \ldots \ b_m] \qquad (3.2.13)$$

$$\tilde{\theta}(k) = \hat{\theta}(k) - \theta \qquad (3.2.14)$$

The design objective is to find a PAA of the form:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + f_\theta[e(k+1)] \qquad (3.2.15)$$

$$e(k+1) = f_e[e_o(k+1)] \qquad (3.2.16)$$

such that $\lim_{k\to\infty} e(k+1)=0$ for any initial conditions $e(0)$ and $\hat{\theta}(0)$.

From equation (3.2.15) using equation (3.2.14) we obtain the following equations:

$$\tilde{\theta}(k+1) = \tilde{\theta}(k) + f_\theta[e(k+1)] \qquad (3.2.17)$$

$$\phi(k)^T \tilde{\theta}(k+1) = \phi(k)^T \tilde{\theta}(k) + \phi(k)^T f_\theta[e(k+1)] \qquad (3.2.18)$$

From the equations (3.2.12), (3.2.17), and (3.2.18), we have an equivalent feedback representation shown in Figure 3.2.2 for this parallel adaptive predictor. Equation

(3.2.12) corresponds to a linear time-invariant feedforward block, with transfer function $[1-z^{-1}A(z^{-1})]^{-1}$. The input is $-\tilde{\Theta}(k+1)^T\emptyset(k)$ and output is $e(k+1)$. Equations (3.2.17) and (3.2.18) correspond to a nonlinear time-varying feedback block with input $e(k+1)$ and output $\tilde{\Theta}(k+1)^T\emptyset(k)$.

Using hyperstability theory, a function $f_\Theta[e(k+1)]$ for the adaptation algorithm can be found such that the feedback system (Figure 3.2.2) is globally asymptotically stable.

First, the Popov inequality condition will be checked for the feedback block of this system and then the stability conditions for the feedforward block (ie, a strictly positive real transfer function) will be checked.

From equations (3.2.17) and (3.2.18) the Popov inequality for the feedback block is defined by:

$$n(0,k_1) = \sum_{k=0}^{k_1} e(k+1)\emptyset(k)^T\tilde{\Theta}(k+1)$$

$$= \sum_{k=0}^{k_1} e(k+1)\emptyset(k)^T[\sum_{i=0}^{k} f_\Theta[e(i+1)] + \tilde{\Theta}(0)] > -r_o^2 \qquad (3.2.19)$$

To find the $f_\Theta[e(k+1)]$ which satisfies the Popov

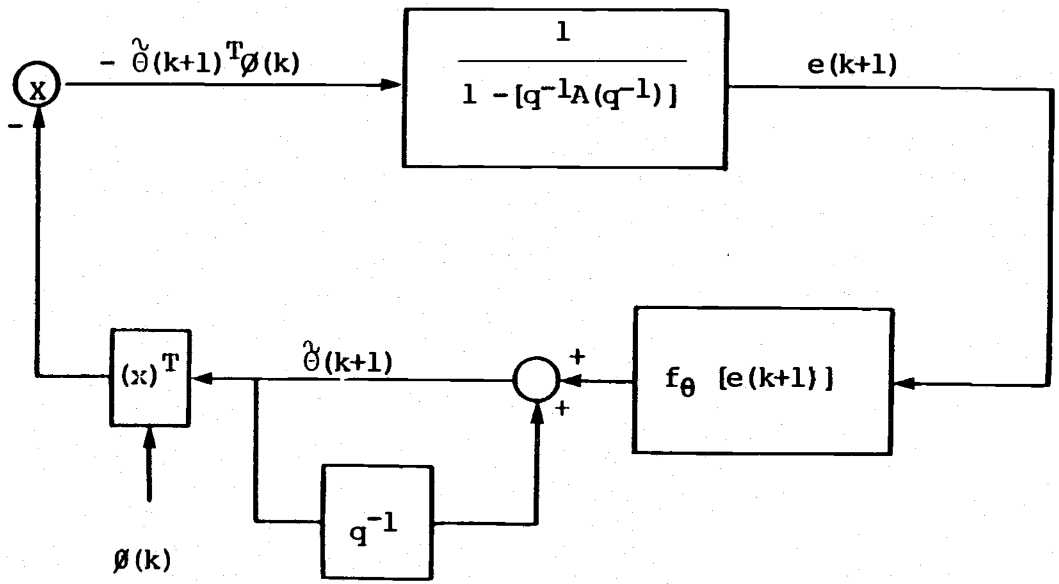

Figure 3.2.2 Equivalent Feedback System Representation

inequality (3.2.19) the following lemma 3.2.1 is introduced.

Lemma 3.2.1

Consider a sequence of real vectors x(k) and a constant vector c. The following relation is true:

$$\sum_{k=0}^{k_1} x(k)^T\left[\sum_{i=0}^{k} x(i) + c\right] = \frac{1}{2}\left[\sum_{k=0}^{k_1} x(k) + c\right]^T\left[\sum_{k=0}^{k_1} x(k) + c\right] + \frac{1}{2}\sum_{k=0}^{k_1} x(k)^T x(k) - \frac{1}{2}c^T c \geq -\frac{1}{2}c^T c \qquad (3.2.20)$$

Proof: Assuming that the relation (3.2.20) is true up to (k_1-1), one gets at k_1:

$$\sum_{K=0}^{k_1} x(k)^T\left[\sum_{i=0}^{k} x(i) + c\right] = \sum_{k=0}^{k_1-1} x(k)^T\left[\sum_{i=0}^{k} x(i) + c\right] + x(k_1)^T\left[\sum_{i=0}^{k_1} x(i) + c\right]$$

$$= \frac{1}{2}\left[\sum_{k=0}^{k_1-1} x(k) + c\right]^T\left[\sum_{k=0}^{k_1-1} x(k) + c\right] + \frac{1}{2}\sum_{k=0}^{k_1-1} x(k)^T x(k) - \frac{1}{2}c^T c + \frac{1}{2}x(k_1)^T x(k_1) + \frac{1}{2}x(k_1)^T x(k_1)$$

$$+ x(k_1)^T[\sum_{i=0}^{k_1-1} x(i) + c]$$

(3.2.21)

However

$$\frac{1}{2}[\sum_{k=0}^{k_1} x(k)+c]^T[\sum_{k=0}^{k_1} x(k)+c] = \frac{1}{2}[\sum_{k=0}^{k_1-1} x(k)+c]^T[\sum_{k=0}^{k_1-1} x(k)+c]$$

$$+ \frac{1}{2}x(k_1)^T x(k_1)$$

$$+ x(k_1)^T[\sum_{k=0}^{k_1-1} x(k) + c]$$

(3.2.22)

Thus, using equations (3.2.22) and (3.2.21) it results that equation (3.2.20) is true by induction.

Q.E.D.

Using lemma 3.2.1, the following solution for f_Θ [e(k+1)] results:

$$f_\Theta[e(k+1)] = F(k)\varnothing(k)e(k+1), \quad F(k) > 0 \qquad (3.2.23)$$

Thus,

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + F(k)\varnothing(k)e(k+1), \quad F(k) > 0 \qquad (3.2.24)$$

In simple form a positive constant adaptation gain matrix, F(k)=F>0 can be used instead of a time-varying adaptation

gain matrix. The general adaptation algorithm which includes a time-varying adaptation gain matrix F(k) will be discussed in next section.

We now have to find the relationship between the a priori adaptation error $e_0(k+1)$ and the a posteriori adaptation error e(k+1) to implement this PAA. Using equation (3.2.24) from equations (3.2.9) and (3.2.10), we have

$$\begin{aligned} e(k+1) - e_0(k+1) &= -[\hat{\Theta}(k+1) - \hat{\Theta}(k)]^T \emptyset(k) \\ &= -\emptyset(k)^T F \emptyset(k) e(k+1) \qquad (3.2.25) \end{aligned}$$

Thus, the relationship between e(k+1) and $e_0(k+1)$ will be given by:

$$e(k+1) = \frac{e_0(k+1)}{[1 + \emptyset(k)^T F \emptyset(k)]} \qquad (3.2.26)$$

We have obtained a feedback block which satisfies the Popov inequality. Therefore if the linear time-invariant feedforward transfer function $[1-z^{-1}A(z^{-1})]^{-1}$ is strictly positivity real, global asymptotic stability will be assured. This positivity condition on the feedforward block can be made less restrictive by filtering the error as shown in the next section. Also we must make sure that $\emptyset(k)$ is asymptotically bounded in equation (3.2.26) so that $\lim_{k \to \infty} e_0(k)=0$ when $\lim_{k \to \infty} e(k)=0$. The positivity condition on the feedforward block and the boundedness of $\emptyset(k)$ will be discussed again in section 3.3 and section 4.3.

3.3 Stability Analysis of the General Time Varying PAA

In this section an approach due to Landau and Lozano [40] is introduced to design and analyze a general time varying PAA for asymptotic stability of the equivalent feedback system representation shown in Figure 3.3.1.

Theorem 3.3.1

Assume that the following adaptation algorithm is used to update the parameter vector $\hat{\theta}(k)$:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k)\phi(k)\nu(k+1) \tag{3.3.1}$$

where

$$F(k+1)^{-1} = \lambda_1(k)F(k)^{-1} + \lambda_2(k)\phi(k)\phi(k)^T \tag{3.3.2}$$

with $F(0)>0$, $0<\lambda_1(k)\leq 1$, $0\leq\lambda_2(k)<2$; for all k.

Assume that the relation between $\phi(k)$ and $\nu(k)$ is given by:

$$\nu(k+1) = G(q^{-1})[\theta - \hat{\theta}(k+1)]^T\phi(k) \tag{3.3.3}$$

where $\phi(k)$ is a bounded or unbounded vector sequence, $G(z^{-1})$ is a rational discrete transfer function normalized under the form:

$$G(q^{-1}) = \frac{G_1(q^{-1})}{G_2(q^{-1})} \tag{3.3.4}$$

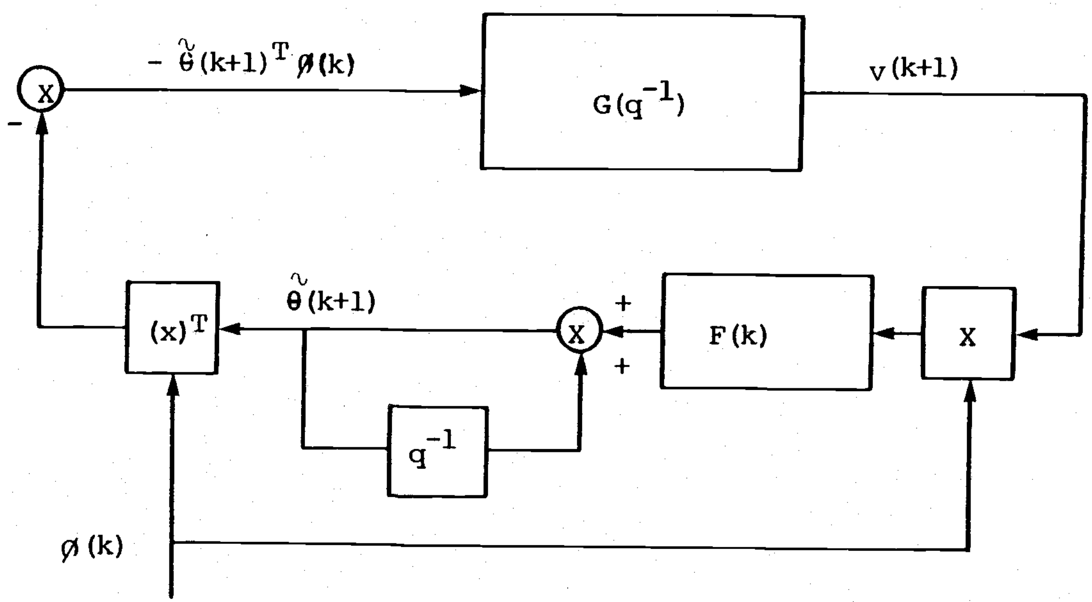

Figure 3.3.1 General Equivalent Feedback System Representation

with $G_j(q^{-1}) = 1 + \sum_{i=1}^{n_j} g_i^j q^{-i}$

$$= 1 + q^{-1}G_j^*(q^{-1}), \quad j=1,2 \qquad (3.3.5)$$

and Θ is a constant parameter vector. Then if it exists :

$$2 > \lambda \geq \max[\lambda_2(k)], \quad 0 \leq k \leq \infty \qquad (3.3.6)$$

such that $G(z^{-1}) - \frac{\lambda}{2}$ (3.3.7)

is strictly positive real, one has, for any $\nu(0)$ and $\emptyset(0)$, bounded:

$$\lim_{k\to\infty} \nu(k) = 0 \qquad (3.3.8)$$

The other related results and proof of theorem 3.3.1 can be found in ref. [40,41].

Thus, with the positivity condition (3.3.7) we can summarize the general parameter adaptation algorithm in the following form:

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + F(k)\emptyset(k)\nu(k+1) \qquad (3.3.1)$$

$$\nu(k+1) = \frac{\nu_0(k+1)}{[1 + \emptyset(k)^T F(k)\emptyset(k)]} \qquad (3.3.9)$$

$$F(k+1)^{-1} = \lambda_1(k)F(k)^{-1} + \lambda_2(k)\emptyset(k)\emptyset(k)^T \qquad (3.3.2)$$

$$0 < \lambda_1(k) \leq 1, \; 0 \leq \lambda_2(k) < 2, \quad F(0) > 0$$

where $\hat{\Theta}(k)$ is an adjustable parameter vector, $F(k)$ is the adaptation gain matrix, $\emptyset(k)$ is the measurement vector, $\nu_0(k+1)$ is the a priori filtered adaptation error which is defined by:

$$\nu_0(k+1) = G(q^{-1})[\Theta - \hat{\Theta}(k)]^T\emptyset(k) \qquad (3.3.10)$$

and $\nu(k+1)$ is the a posteriori filtered adaptation error which satisfies the equation (3.3.3).

The adaptation gain matrix $F(k+1)$ is computed recursively from equation (3.3.2) using the matrix inversion lemma 3.3.1.

Lemma 3.3.1

Consider a nonsingular nxn-dimensional matrix F a nonsingular mxm-dimensinal matrix R and a mxn-dimensional matrix H of maximum rank, then the following indentity holds:

$$(F^{-1} + HR^{-1}H^T)^{-1} = F - FH(R + H^TFH)^{-1}H^TF \qquad (3.3.11)$$

Proof: By right multiplication of equation (3.3.11) with $(F^{-1} + HR^{-1}H^T)$, one obtains:

$$I = I + FHR^{-1}H^T$$
$$- FH(R + H^TFH)^{-1}H^T - FH(R + H^TFH)^{-1}H^TFHR^{-1}H^T$$
$$= I + FHR^{-1}H^T - FH(R + H^TFH)^{-1}(R + H^TFH)R^{-1}H^T$$
$$= I \qquad (3.3.12)$$

Q.E.D.

Thus, from equation (3.3.11) making the following substitutions

$$F^{-1} + HR^{-1}H = [\lambda_1 F(k+1)]^{-1} \qquad (3.3.13)$$

$$R = \frac{\lambda_1}{\lambda_2} \qquad (3.3.14)$$

$$F = F(k) \qquad (3.3.15)$$

$$H = \emptyset(k) \qquad (3.3.16)$$

F(k+1) is given in the following form:

$$F(k+1) = \frac{1}{\lambda_1(k)} [F(k) - \frac{F(k)\emptyset(k)\emptyset(k)^T F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)} + \emptyset(k)^T F(k)\emptyset(k)}] \qquad (3.3.17)$$

$\lambda_1(k)$ and $\lambda_2(k)$ have opposite effects: $\lambda_1(k)$ tends to increase the adaptation gain while $\lambda_2(k)$ tends to decrease the adaptation gain. By the different choices of $\lambda_1(k)$ and $\lambda_2(k)$, different types of adaptation algorithm are obtained. For the following choices of $\lambda_1(k)$ and $\lambda_2(k)$, different adaptation algorithms were discussed by the several authors [42, 43, 44]:

(1) $\lambda_1(k)=1$, $\lambda_2(k)=0$: a constant adaptation gain

(2) $\lambda_1(k)=1$, $\lambda_2(k)=\lambda$: a time decreasing adaptation gain

(3) $\lambda_1(k)=\lambda_1$, $\lambda_2(k)=\lambda_2$: a time varying adaptation gain which is useful for slowly time varying plants

(4) $\lambda_1(k)$, $\lambda_2(k)$ such that trace $[F(k)]$ = constant: a real time adaptation algorithm for tracking time varying plants.

3.4 Design of the PAA based on the Direct Use of a Lyapunov Function

In this section a Lyapunov function is introduced to derive a simple PAA assuring global asymptotic stability for the error model described by equation (3.2.11). To do this the following lemma 3.4.1 is introduced.

Lemma 3.4.1

Consider the linear time-invariant system:

$$x(k+1) = Ax(k) + Bu(k) \tag{3.4.1}$$

$$y(k) = Cx(k) + Du(k) \tag{3.4.2}$$

$$u(k) = p(k+1)^T \emptyset(k) \tag{3.4.3}$$

and the adaptation law:

$$p(k+1) = p(k) - F\emptyset(k)y(k)^T \tag{3.4.4}$$

where x is an n-dimensional vector, y and u are m_1-dimensional vectors, $\emptyset$ is an m_2-dimensional vector, A,B,C and D are matrices of approciate dimension, p is a time-varying matrix of appropriate dimension, and F is an $m_2 x m_2$ dimensional matrix.

Given an nxn-dimensional matrix A with all its eigenvalues within the unit circle, a symmetric positive definite matrix F, (A,B) is completely controllable, (C,A) is completely observable and a bounded vector $\emptyset(k)$, the equilibrium state of the set of $n+m_2 x m_1$ difference equations (3.4.1) and (3.4.4) is stable and $\lim_{k\to\infty} x(k)=0$ if the transfer function $H(z)=D+C(zI-A)^{-1}B$ is strictly positive real.

Proof: From the strictly positive real lemma 2.3.1 , it is known that if H(z) is strictly positive real, there exist a matrix $P=P^T>0$, a matrix $Q=Q^T>0$, and matrices K and L such that:

$$A^TPA - P = -LL^T - Q \tag{3.4.5}$$

$$B^TPA + K^TL^T = C \tag{3.4.6}$$

$$K^TK = D + D^T - B^TPB \tag{3.4.7}$$

Defining a Lyapunov function candidate for the set of difference equations (3.4.1) and (3.4.4) as

$$V(x(k),p(k)) = x(k)^TPx(k) + tr[p(k)^TF^{-1}p(k)] \tag{3.4.8}$$

we obtain

$$\begin{aligned}\Delta V(x(k),p(k)) &\triangleq \Delta V(k)\\ &= V(k+1) - V(k)\\ &= x(k+1)^TPx(k+1) + tr[p(k+1)^TF^{-1}p(k+1)]\\ &\quad -x(k)^TPx(k) - tr[p(k)^TF^{-1}p(k)]\end{aligned} \tag{3.4.9}$$

Using equation (3.4.1),

$$\Delta V(k) = x(k)^T[A^TPA - P]x(k) + x(k)^TA^TPBu(k) + u(k)^TB^TPAx(k) + u(k)^TB^TPBu(k) + tr[p(k+1)^TF^{-1}p(k+1)] - tr[p(k)^TF^{-1}p(k)] \tag{3.4.10}$$

Using equations (3.4.5) through (3.4.7),

$$\Delta V(k) = -[L^Tx(k) + Ku(k)]^T[L^Tx(k) + Ku(k)] - x(k)^TQx(k) + y(k)^Tu(k) + u(k)^Ty(k) + tr[p(k+1)^TF^{-1}p(k+1)] - tr[p(k)^TF^{-1}p(k)] \tag{3.4.11}$$

Using adaptation law (3.4.4)

$$\Delta V(k) = -[L^Tx(k) + Ku(k)]^T[L^Tx(k) + Ku(k)] - x(k)^TQx(k) - tr[y(k)\emptyset(k)^TF^{-1}\emptyset(k)y(k)^T] \le 0 \tag{3.4.12}$$

Thus, the system of equations (3.4.1) and (3.4.4) is stable and x(k) and p(k) are bounded if x(0) and p(0) are bounded.

Q.E.D.

From equations (3.2.12) and (3.2.24), making substitutions

$$x(k)^T = [e(k-n+1)\ e(k-n+2)\ \ldots\ e(k)] \tag{3.4.13}$$

$$u(k) = -\gamma(k+1)\emptyset(k) \tag{3.4.14}$$

$$y(k) = e(k+1) \tag{3.4.15}$$

we obtain a system of the form equations (3.4.1) through (3.4.4) where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 \cdots & 0 \\ \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & 0 \\ & & & & 1 \\ a_n & a_{n-1} & \cdots & & a_1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} a_n & a_{n-1} & \cdots & a_1 \end{bmatrix}$$

$$D = 1$$

and $p(k) = -\tilde{\Theta}(k)$. From equation (3.4.8) a Lyapunov function for this system is obtained. Thus from equation (3.4.4) the following adaptation law is obtained for the error equation (3.2.12):

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + F\emptyset(k)e(k+1) \qquad (3.4.16)$$

This adaptation law results in a stable error equation (3.2.12) in which $e(k)$ and $\hat{\Theta}(k)$ remain uniformly bounded and $\lim_{k\to\infty} e(k)=0$ if $\emptyset(k)$ is uniformly bounded.

However from equation (3.4.12) we can have the following equation

$$\left|\sum_{k=0}^{\infty} V(k)\right| = |V(\infty) - V(0)| < \infty \qquad (3.4.17)$$

since $\Delta V(k) \leq 0$.

Thus, having established the global stability of the error equation we now state some of its other stability properties:

(1) The boundness of $e(k)$ and $\hat{\Theta}(k)$ are assured even when $\emptyset(k)$ is not bounded.

(2) $e(k)$ tends to zero whether $\emptyset(k)$ is bounded or not.

(3) $\lim_{k \to \infty} [\Delta \hat{\Theta}(k) = \hat{\Theta}(k+1) - \hat{\Theta}(k)] = 0$

But we cannot conclude directly that $\hat{\Theta}(k)$ tends to a constant vector Θ.

Using a Lyapunov function we have obtained the same adaptation law as the simple form of sectin 3.2. However this approach is limited since it is difficult to choose a suitable Lyapunov functions in order to widen the class of adaptation laws which lead to globally stable MRAC systems.

In the next chapter, we will design a MRAC system for tracking and regulation control objectives, using the PAA of section 3.3.

IV. DESIGN OF THE DISCRETE-TIME EXPLICIT MODEL REFERENCE ADAPTIVE CONTROL SYSTEM

4.1 Introduction

A parameter adaptation algorithm (PAA) has been developed and analyzed from the stability point of view in chapter 3. This PAA can be generalized for a plant with a delay greater than 1. Introducing such a plant time delay into the previous PAA, we will design a model refernce adaptive control (MRAC) system based on a unification of the discrete-time explicit schemes proposed by Landau and Lozano [44,45].

In section 4.2 the parameters of the plant are assumed to be known and a reference model is defined. Then a linear model following control (LMFC) system is designed for independent tracking and regulation control objectives.

In section 4.3, based on the structure of the control law of section 4.2 we obtain a control law which has adjustable parameters which compensate for the unknown parameters of plant. A PAA is then designed to adjust the controller. In order to improve the performance of the MRAC system in the stochastic environment, an asymptotically stable filter is introduced, and the MRAC system is redesigned using filtered variables.

4.2 Design of the Linear Control System

Consider a discrete-time linear time-invariant plant described by:

$$y_p(k+d) = A_p(q^{-1})y_p(k+d-1) + B_p(q^{-1})u_p(k) \qquad d>0,\ y_p(0)=0 \qquad (4.2.1)$$

where

$$A_p(q^{-1}) = a_1^p + \dots + a_n^p q^{-n+1} \qquad (4.2.2)$$

$$B_p(q^{-1}) = b_o^p + b_1^p q^{-1} + \dots + b_m^p q^{-m} \qquad (4.2.3)$$

q^{-1} is the unit time delay operator, $u(k)$ is input, $y(k)$ is output and d is the plant time delay. The zeros of $B(z^{-1})$ are all assumed to be inside the unit circle, therefore they can be cancelled without leading to an unbounded control input.

The following control objectives are considered:

(1) Tracking objective

The output of plant must be equal to the output of reference model

$$y_p(k) - y_M(k) = 0 \qquad (4.2.4)$$

The reference model is given by

$$y_M(k+d) = A_M(q^{-1})y_M(k+d-1) + B_M(q^{-1})u_M(k) \qquad (4.2.5)$$

where $A_M(q^{-1})$ is asymptotically stable, $u_M(k)$ is a bounded reference input sequence and $A_M(q^{-1})$ and $B_M(q^{-1})$ are defined by:

$$A_M(q^{-1}) = a_1^M + \dots + a_n^M q^{-n+1} \quad (4.2.6)$$

$$B_M(q^{-1}) = b_o^M + b_1^M q^{-1} + \dots + b_M^M q^{-m} \quad (4.2.7)$$

(2) Regulation objective

An initial disturbance must be eliminated with the following dynamics:

$$C_R(q^{-1})y_p(k+d) = 0, \qquad k > 0 \quad (4.2.8)$$

when the reference model is zero,

where

$$C_R(q^{-1}) = 1 + c_1^R q^{-1} + \dots + c_n^R q^{-n} \quad (4.2.9)$$

is an asymptotically stable linear operator.

The control law is designed to achieve the above two objectives which can be summarized by the equation

$$C_R(q^{-1})[y_p(k+d) - y_M(k+d)] = 0, \quad k > 0 \quad (4.2.10)$$

Lemma 4.2.1 is introduced to represent $C_R(q^{-1})$ in a useful equivalent form.

Lemma 4.2.1

Consider the following equation

$$C(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d}R(q^{-1}) \qquad (4.2.11)$$

where

$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots. + c_{nc} q^{-nc}$$
$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots. + a_{nA} q^{-nA}$$
$$S(q^{-1}) = 1 + s_1 q^{-1} + \ldots. + s_{ns} q^{-ns}$$
$$R(q^{-1}) = r_o + r_1 q^{-1} + \ldots. + r_{nR} q^{-nR}$$

For given $C(q^{-1})$ and $A(q^{-1})$ a unique solution $S(q^{-1})$, $R(q^{-1})$ exists for $n_s = d-1$ and $n_R = \max(n_A-1, n_c-d)$.

Proof: Let us rewrite equation (4.2.11) as a set of equations corresponding to the various power of q^{-1}:

$$\begin{aligned} 1 &= 1 \\ c_1 &= a_1 + s_1 \\ c_2 &= a_2 + a_1 s_1 + s_2 \\ &\cdot \\ &\cdot \\ &\cdot \\ c_{d-1} &= a_{d-1} + \ldots. + s_{d-1} \\ c_d &= a_d + \ldots. + s_d + r_o \end{aligned} \qquad (4.2.12)$$

The number of coefficients to be computed is $n_s + n_R + 1$. On the other hand, the number of equations in (4.2.12) is

$\max(n_A + n_s, n_R + d)$ and in order to have a unique solution, we must verify:

$$n_s + n_R + 1 = \max(n_A+n_s, n_R+d) > n \qquad (4.2.13)$$

We can also see from equation (4.2.12) that n_s must verify:

$$n_s \geq d - 1 \qquad (4.2.14)$$

Let us take the smallest number $n_s = d - 1$, then from equation (4.2.13) we have

$$n_s = d - 1$$
$$n_R = \max(n_A - 1, n_C - d) \qquad (4.2.15)$$

Thus taking into account equation (4.2.15), equation (4.2.11) can be written by:

$$\begin{bmatrix} 1 & & & & & & & & \\ a_1 & 1 & & & & \underline{0} & & & \\ a_2 & a_1 & 1 & & & & & & \\ \cdot & \cdot & & & & & & & \\ \cdot & \cdot & & & & & & & \\ a_{d-1} & a_{d-2} & \cdots & a_1 & 1 & & & & \\ a_d & a_{d-1} & \cdots & & a_1 & 1 & & & \\ a_{d+1} & & & & a_2 & 0 & 1 & & \\ \cdot & & & & \cdot & \cdot & \cdot & & \\ \cdot & & & & \cdot & \cdot & \cdot & & \\ \cdot & & & & & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ s_1 \\ s_2 \\ \cdot \\ \cdot \\ s_{d-1} \\ r_o \\ r_1 \\ \cdot \\ \cdot \\ r_{nR} \end{bmatrix} = \begin{bmatrix} 1 \\ c_1 \\ c_2 \\ \cdot \\ \cdot \\ c_{d-1} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ c_{nc} \end{bmatrix}$$

and since the determinant of a triangular matrix is always different from zero, the solution exists and is unique.

Q.E.D.

Using the lemma 4.2.1, write $C_R(q^{-1})$ in the form

$$C_R(q^{-1}) = [1-q^{-1}A_p(q^{-1})]S(q^{-1}) + q^{-d}R(q^{-1}) \qquad (4.2.17)$$

where

$$S(q^{-1}) = 1 + s_1q^{-1} + \dots + s_{d-1}q^{-d+1} \qquad (4.2.18)$$

$$R(q^{-1}) = r_o + r_1q^{-1} + \dots + r_{n-1}q^{-n+1} \qquad (4.2.19)$$

Thus, using equation (4.2.17) we can rewrite the equation (4.2.10):

$$\begin{aligned} & C_R(q^{-1})[y_p(k+d) - y_M(k+d)] \\ &= B_p(q^{-1})S(q^{-1})u_p(k) + R(q^{-1})y_p(k) - C_R(q^{-1})y_M(k+d) \\ &= B_s(q^{-1})u_p(k) + R(q^{-1})y_p(k) - C_R(q^{-1})y_M(k+d) \\ &= 0 \end{aligned} \qquad (4.2.20)$$

where

$$\begin{aligned} B_s(q^{-1}) &= B(q^{-1})S(q^{-1}) \\ &= b_o + q^{-1}B_s^*(q^{-1}) \\ &= b_o + b_1^sq^{-1} + \dots + b_{m+d-1}^sq^{-m-d+1} \end{aligned} \qquad (4.2.21)$$

From equation (4.2.20) we can find $u_p(k)$ in the following form:

$$u_p(k) = \frac{1}{b_o}[C_R(q^{-1})y_M(k+d) - R(q^{-1})y_p(k) - B_s^*(q^{-1})u_p(k-1)] \qquad (4.2.22)$$

The block diagram of the LMFC system is shown in Figure 4.2.1.

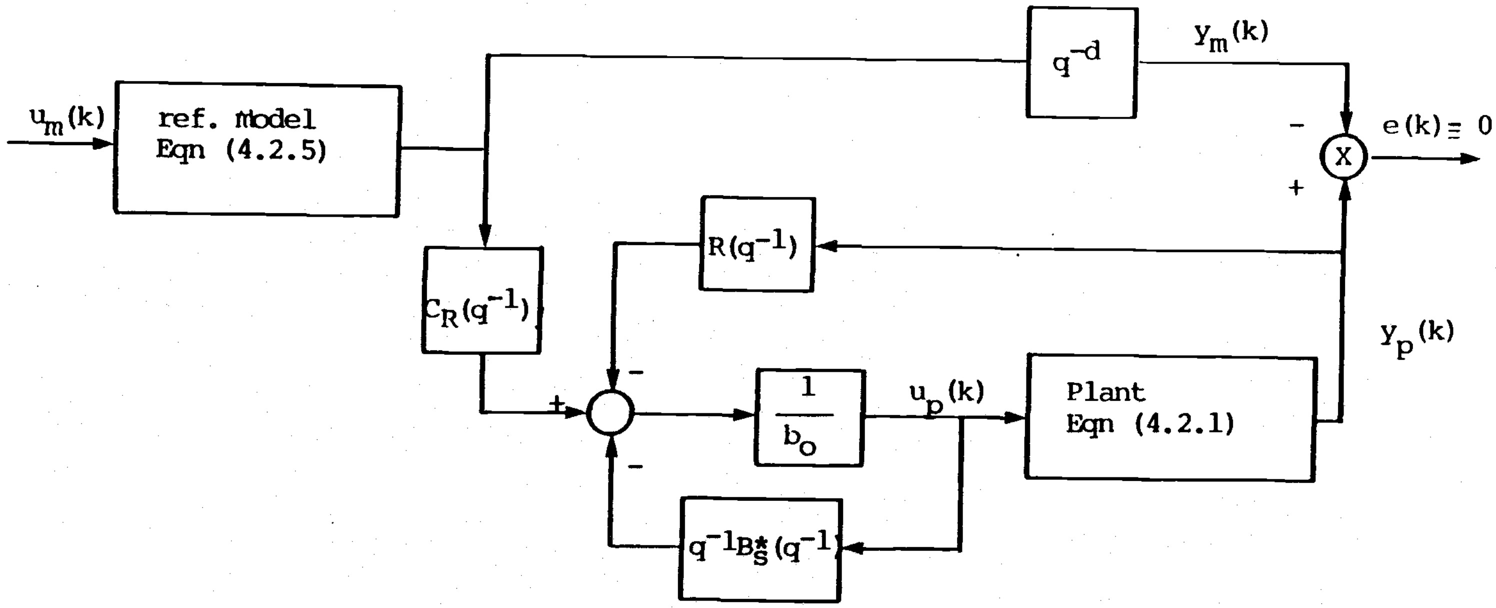

Figure 4.2.1 Diagram of LMFC System

If we consider the following plant dynamics with special disturbance, $w(k)$:

$$y_p(k+d) = A_p(q^{-1})y_p(k+d-1) + B_p(q^{-1})u(k) + w(k+d) \quad (4.2.23)$$

where

$$w(k) = C_R(q^{-1})v(k) \quad (4.2.24)$$

$v(k)$ is a sequence of independent random variables with zero mean $N(0,\delta)$, the same control law of equation (4.2. 22) is obtained for minimum variance tracking and regulat ion objectives.

4.3 Design of the Adaptive Control System

When the plant parameter values are unknown, a controller with the same model following structure as in the previous section but whose parameters are adjustable can be introduced.

Based on the structure of equation (4.2.22) of section 4.2, we get the following control law which has adjustable parameters:

$$u_p(k) = \frac{1}{\hat{b}_o(k)}[C_R(q^{-1})y_M(k+d) - \hat{R}(k,q^{-1})y_p(k) - \hat{B}_s^*(k,q^{-1})u_p(k-1)] \tag{4.3.1}$$

The filtered plant model error is therefore given by:

$$\begin{aligned} e_o^f(k+d) &= C_R(q^{-1})[y_p(k+d) - y_M(k+d)] \\ &= [\Theta - \hat{\Theta}(k)]^T \emptyset(k) \end{aligned} \tag{4.3.2}$$

where

$$\Theta^T = [b_o \; b_1^s \; \ldots \; b_{m+d-1}^s \; r_o \; \ldots \; r_{n-1}] \tag{4.3.3}$$

$\hat{\Theta}(k)$ is an adjustable approximation for the parameters at k-th time step, and

$$\emptyset(k) = [u_p(k) \; u_p(k-1) \; \ldots \; u_p(k-m+d-1) \; y_p(k) \; y\,(k-1) \; \ldots \; \ldots \; y_p(k-n+1)] \tag{4.3.4}$$

The design objective is to find an adaptation algorithm such that

$$\lim_{k \to \infty} e_o^f(k+d) = 0 \tag{4.3.5}$$

and $||\emptyset(k)|| \le M < \infty$ for all k. (4.3.6)

Considering theorem 3.3.1, an asymptotically stable adaptive system can be obtained if the following PAA is used:

$$\hat{\Theta}(k+d) = \hat{\Theta}(k+d-1) + F(k)\emptyset(k)\nu(k+d) \tag{4.3.7}$$

where the a posteriori adaptation error $\nu(k+d)$ is governed by:

$$\nu(k+d) = [\Theta - \hat{\Theta}(k+d)]^T\emptyset(k) \tag{4.3.8}$$

with

$$F(k+1) = \frac{1}{\lambda_1(k)}\left[F(k) - \frac{F(k)\emptyset(k)\emptyset(k)^T F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)} + \emptyset(k)^T F(k)\emptyset(k)}\right] \tag{4.3.9}$$

where $0 < \lambda_1(k) \le 1$, $0 \le \lambda_2(k) < 2$, $F(0) > 0$.

Since the discrete transfer function, $G(z^{-1}) = 1$ in equation (4.3.8) the positivity condition (3.3.7) is satisfied.

Using equations (4.3.2), (4.3.7), and equation (4.3.8) $\nu(k+d)$ can be described by:

$$\nu(k+d) = e_o^f(k+d) + [\hat{\Theta}(k) - \hat{\Theta}(k+d)]^T \emptyset(k)$$
$$= e_o^f(k+d) - \emptyset(k)^T \sum_{i=1}^{d-1} F(k-d+i)\emptyset(k-d+i)\nu(k+i)$$
$$- \emptyset(k)^T F(k)\emptyset(k)\,\nu(k+d) \qquad (4.3.10)$$

Thus,

$$\nu(k+d) = \frac{\nu_o(k+d)}{[1 + \emptyset(k)^T F(k)\emptyset(k)]} \qquad (4.3.11)$$

where the a priori adaptation error $\nu_o(k+d)$ is governed by:

$$\nu_o(k+d) = e_o^f(k+d) - \emptyset(k)^T \sum_{i=1}^{d-1} F(k-d+i)\emptyset(k-d+i)\nu(k+i) \qquad (4.3.12)$$

The block diagram of this adaptive control system is shown in Figure 4.3.1.

In order to improve the performance of the adaptive control system in a stochastic environment, the measurement vector $\emptyset(k)$ and $y_M(k)$ can be filtered by an asymptotially stable filter such that:

$$L(q^{-1})\emptyset^f(k) = \emptyset(k) \qquad (4.3.13)$$

$$L(q^{-1})y_M^f(k) = y_M(k) \qquad (4.3.14)$$

where

$$L(q^{-1}) = \ell_o + \ell_1 q^{-1} + \ldots. + \ell_n q^{-n} = 1 + q^{-1}L^*(q^{-1})$$
$$(4.3.15)$$

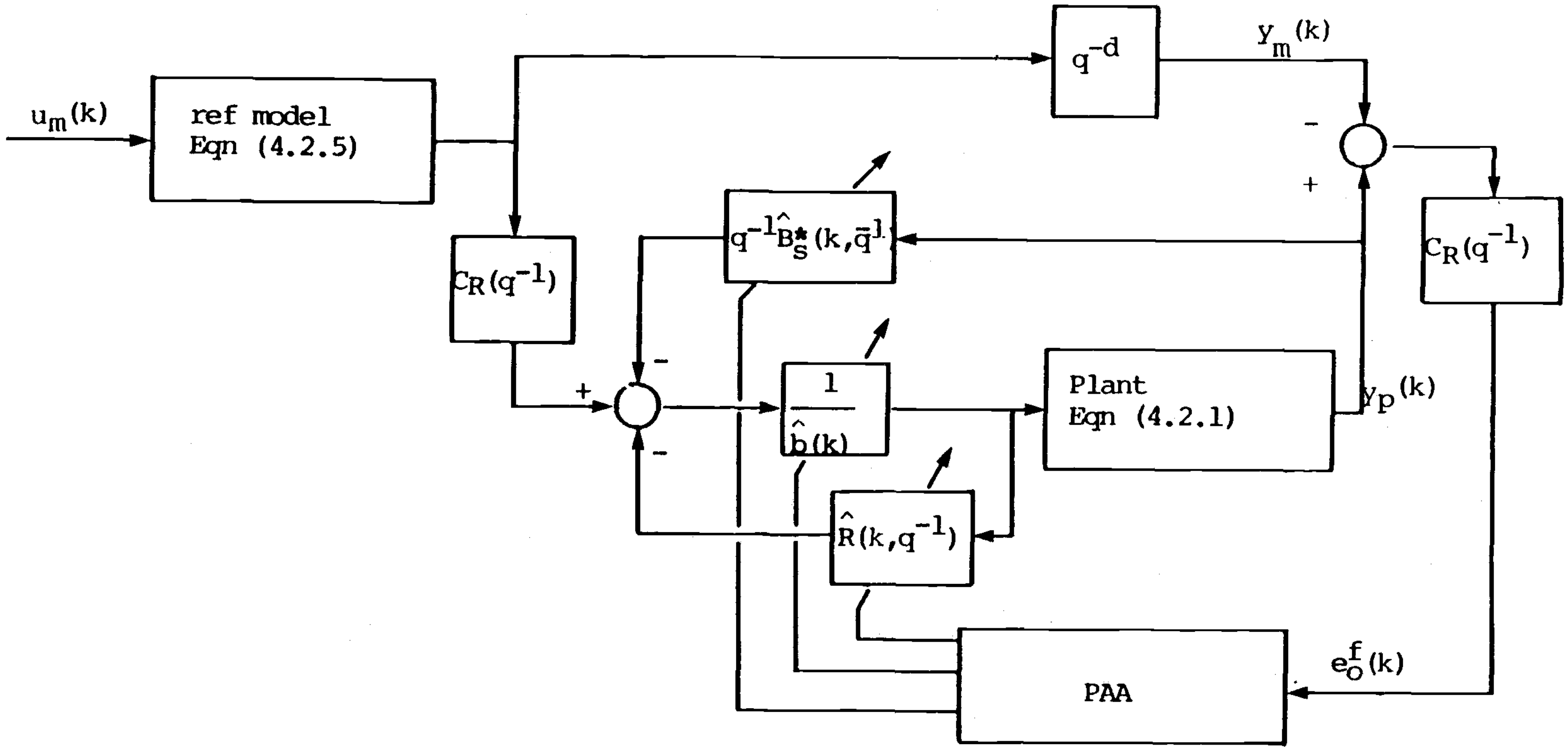

Figure 4.3.1 Diagram of Explicit MRAC System

Therefore, the filtered vector $\phi^f(k)$ is given by:

$$\phi^f(k)^T = \frac{1}{L(q^{-1})}[u_p(k) \; \ldots \; u_p(k-m-d+1) \; y_p(k) \; \ldots \; y_p(k-n+1)]$$
$$= [u_p^f(k) \; \ldots \; u_p^f(k-m-d+1) \; y_p^f(k) \; \ldots \; y_p^f(k-n+1)] \quad (4.3.16)$$

Using equations (4.3.14) through (4.3.16), the control law (4.3.1) can be rewritten in following form:

$$u_p(k) = L(q^{-1})u_p^f(k) \quad (4.3.17)$$

$$u_p^f(k) = \frac{1}{\hat{b}_o(k)}[C_R(q^{-1})y_M^f(k+d) - \hat{B}_s^*(k,q^{-1})u_p^f(k-1) - \hat{R}(k,q^{-1})y_p^f(k)] \quad (4.3.18)$$

Using equation (4.3.13) the filtered plant model error of equation (4.3.2) is redefined as:

$$e_L^f(k+d) = C_R(q^{-1})[y_p(k+d) - y_M(k+d)]$$
$$= L(q^{-1})[\Theta - \hat{\Theta}(k)]^T\phi^f(k) \quad (4.3.19)$$

In general e_L^f can be further filtered before it is used for adaptation. Thus the filtered error is defined as:

$$e^f(k+d) = \frac{G_1(q^{-1})}{G_2(q^{-1})} e_L^f(k+d) \quad (4.3.20)$$

where $G_1(q^{-1})$ and $G_2(q^{-1})$ are monic polynomials:

$$G_j(q^{-1}) = 1 + q^{-1}G_j^*(q^{-1}), \quad j=1,2 \tag{4.3.21}$$

Using equation (4.3.19), from equation (4.3.20), the filtered error is described by:

$$e^f(k+d) = G(q^{-1})[\Theta - \hat{\Theta}(k)]^T\phi(k) \tag{4.3.22}$$

where

$$G(q^{-1}) = \frac{G_1(q^{-1})L(q^{-1})}{G_2(q^{-1})} \tag{4.3.23}$$

Applying theorem 3.3.1, asymptotic stability of the adaptive system can be obtained if we use a PAA of the following form:

$$\hat{\Theta}(k+d) = \hat{\Theta}(k+d-1) + F(k)\phi^f(k)\,\nu(k+d) \tag{4.3.24}$$

where the a posteriori adaptation error ν (k+d) is governed by:

$$\nu(k+d) = G(q^{-1})[\Theta - \hat{\Theta}(k+d)]^T\phi^f(k) \tag{4.3.25}$$

with

$$F(k+1)^{-1} = \lambda_1(k)F(k)^{-1} + \lambda_2(k)\phi^f(k)\phi^f(k)^T \tag{4.3.26}$$

where $0 < \lambda_1(k) \leq 1$, $0 \leq \lambda_2(k) < 2$, $F(0) > 0$.

$G_1(q^{-1})$, $G_2(q^{-1})$ and $L(q^{-1})$ are any finite dimensional asymptotically stable linear operators satisfying the positivity condition (3.3.7). It was shown by Dugard and

Landau [46] that these operators have an important role in a stochastic environment.

Using equations (4.3.20), (4.3.22), and (4.3.23) $\nu(k+d)$ can be described by

$$\nu(k+d) = e^{f}(k+d) + G(q^{-1})[\hat{\Theta}(k) - \hat{\Theta}(k+d)]^{T}\emptyset^{f}(k)$$

$$= e^{f}(k+d) - G(q^{-1})\emptyset^{f}(k)\sum_{i=1}^{d-1} F(k-d+i)\emptyset^{f}(k-d+i)\nu(k+i)$$

$$- G(q^{-1})\emptyset^{f}(k)^{T}F(k)\emptyset^{f}(k)\nu(k+d) \qquad (4.3.27)$$

Using equations (4.3.15), (4.3.20), (4.3.21), and (4.3.23)

$$\nu(k+d) = - G_{2}^{*}(q^{-1})\nu(k+d-1) + G_{1}(q^{-1})e_{o}^{f}(k+d)$$

$$- G_{1}(q^{-1})L(q^{-1})\emptyset^{f}(k)\sum_{i=1}^{d-1} F(k-d+i)\emptyset(k-d+i)\nu(k+i)$$

$$- [G_{1}^{*}(q^{-1}) + L(q^{-1}) + q^{-1}G_{1}^{*}(q^{-1})L^{*}(q^{-1})]\emptyset^{f}(k+1)^{T}$$

$$F(k-1)\emptyset^{f}(k-1)\nu(k+d-1) - \emptyset^{f}(k)^{T}F(k)\emptyset^{f}(k)\nu(k+d)$$

$$(4.3.28)$$

Thus,

$$\nu(k+d) = \frac{\nu_{o}(k+d)}{[1 + \emptyset^{f}(k)^{T}F(k)\emptyset^{f}(k)]} \qquad (4.3.29)$$

where the a priori adaptation error $\nu_{o}(k+d)$ is governed by:

$$\nu_o(k+d) = - G_2^*(q^{-1})\,\nu(k+d-1) + G_1(q^{-1})e_o^f(k+d)$$

$$- G_1(q^{-1})L(q^{-1})\emptyset^f(k)^T\sum_{i=1}^{d-1} F(k-d+i)\emptyset(k-d+i)\nu(k+i)$$

$$- [G_1^*(q^{-1}) + L^*(q^{-1}) + q^{-1}G_1^*(q^{-1})L^*(q^{-1})]\emptyset^f(k-1)^T$$

$$F(k-1)\emptyset^f(k-1)\nu(k+d-1) \qquad (4.3.30)$$

As mentioned in section 3.2, $\emptyset^f(k)$ of equation (4.3.29) must be asymptotically bounded so that $\lim_{k\to\infty} \nu_o(k+d)=0$ when $\lim_{k\to\infty} \nu(k+d)=0$. if $\emptyset^f(k)$ is asymptotically bounded and $\lim_{k\to\infty}\nu(k+d)=0$ with asymptotically stable polynomials $C_R(q^{-1})$ and $L(q^{-1})$, we can say that $\lim_{k\to\infty} e(k)=0$ where $e(k)=y_p(k)-y_M(k)$ and $\emptyset(k-d)$ is bounded. The boundedness of $\emptyset(k)$ and convergence to zero of $e(k)$ have been discussed and proved in ref. [47].

V. APPLICATION OF MODEL REFERENCE ADAPTIVE CONTROL TO INDUSTRIAL ROBOTIC MANIPULATOR SYSTEMS

5.1 Dynamic Model

In this section, a three degree of freedom (one rotation and two translations) industrial robot similar to SEIKO Model [48] is modelled as an application of the MRAC system structure described in general in the previous chapters.

The configuration of this robot is shown in Figure 5.1.1. The horizontal arm is translated inside the horizontal sleeve which is fixed to the upright column. The column moves vertically and rotates about an axis in z-direction.

For simplicity of the dynamic analysis, we assume that:

(1) Arm, sleeve, and column are rigid,

(2) Friction forces are negligible, and

(3) The dynamics of the actuators are neglected.

The dynamic equations can be derived using the Lagrangian approach:

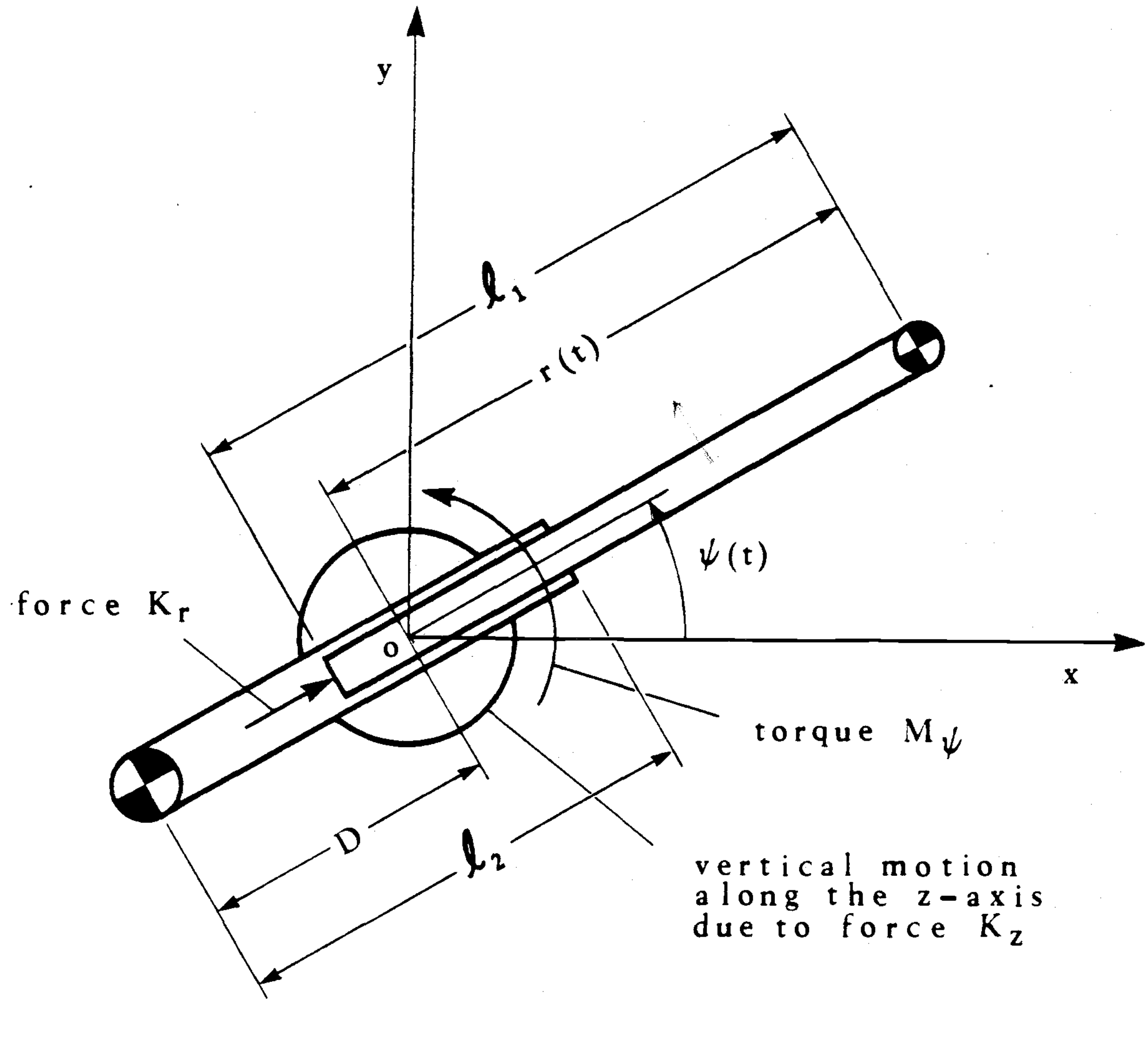

Figure 5.1.1 Three degree-of-freedom Robotic Manipulator

$$Q(t) = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} \qquad (5.1.1)$$

where Q is m-dimensional vector of the generalized forces to be supplied by the actuators, q and $\dot{q}$ are respectively the m-dimensional vector of the rotations or translations and their velocities and L is the Lagrange function described in terms of the kinetic energy T(q,q) and of the potential energy V(q):

$$L = T(q,\dot{q}) - V(q) \qquad (5.1.2)$$

For this robot, the kinetic energy T and the potential energy V are given by equations (5.1.3) and (5.1.4) respectively:

$$T = \frac{1}{2} M_r\dot{r}^2 + \frac{1}{2} I_z\dot{\psi}^2 + \frac{1}{2} M_z\dot{z}^2 \qquad (5.1.3)$$

$$V = M_z gz \qquad (5.1.4)$$

where r(t) and z(t) describe translational motions, ψ(t) describes the angle of rotation, M_r, I_z, and M_z are the inertia terms corresponding to r, ψ and z, and g is the gravity coefficient.

Since real robot systems usually consist of the complicated mechanical parts (for instance, gears, chains etc.), it is almost impossible to describe the inertia terms by exact values or analytic functions. However, we can see that I_z varies depending on the arm stroke $r(t)$ (ie, I_z is a function of $r(t)$). Thus, using the Lagrangian approach, the dynamic equations are given by:

$$M_r \ddot{r} - \frac{1}{2}\dot{\psi}[\dot{\psi}(\frac{\partial I_z}{\partial r})] = K_r(t) \tag{5.1.5}$$

$$I_z \ddot{\psi} + [\frac{d}{dt}(I_z)]\dot{\psi} = M_\psi(t)$$

$$M_z \ddot{z} + M_z g = K_z(t)$$

where K_r (t), $K_z(t)$, and $M_\psi(t)$ are the forces and the torque corresponding to the degrees of freedom, $r(t)$, $z(t)$, and $\psi(t)$.

Introducing the following vectors:

plant state vector, $x_p^T = [x_{p1}\ x_{p2}\ x_{p3}\ x_{p4}\ x_{p5}\ x_{p6}]$

$$= [r\ \psi\ z\ \dot{r}\ \dot{\psi}\ \dot{z}] \tag{5.1.6}$$

plant input vector, $u_p^T = [u_{p1}\ u_{p2}\ u_{p3}]$

$$= [K_r\ M_\psi\ K_z] \tag{5.1.7}$$

measurable disturbance vector, $W = [0\ W_1]$ (5.1.8)

the equation (5.1.5) can be written in the state space model:

$$\dot{x}_p(t) = A(x_p,t)x_p(t) + B(x_p,t)u_p(t) + W(x_p,t)$$

$$= \left[\begin{array}{c|c} 0 & I \\ \hline A_1(x_p,t) & A_2(x_p,t) \end{array}\right] x_p(t) + \left[\begin{array}{c} 0 \\ \hline B_1(x_p,t) \end{array}\right] u_p(t)$$

$$+ \left[\begin{array}{c} 0 \\ \hline W_1(x_p,t) \end{array}\right] \tag{5.1.9}$$

where

$$A_1 = [\ 0\ ] \tag{5.1.10}$$

$$A_2 = \begin{bmatrix} 0 & a_{12} & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix}$$

$$W_1 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

It is shown in Appendix B that all natural systems [49] can be described in the form of equation (5.1.9).

With the following further assumptions:

(1) Arm, sleeve, and column have purely cylinderical shapes.

(2) These three parts, and the balance weight and payload are simply connected to each other as shown in Figure 5.1.1.

(3) Additional parts except the above stated parts are negligible for dynamic motion of the robot system.

We can describe the elements of matrices A_2, B_1 and vector W_1 as follows:

$$a_{12} = x_{p5}h(x_{p1})/(m_R + m_L) \qquad (5.1.11)$$

$$a_{22} = -2x_{p4}h(x_{p1})/I_Z$$

$$b_{11} = 1/(m_R + m_L)$$

$$b_{22} = 1/I_Z$$

$$b_{33} = 1/(m_R + m_L + m_S + m_T + m_C)$$

where

$$I_Z = [m_R(3a_R^2+3b_R^2+\ell_1^2) + m_S(3a_S^2+3b_S^2+\ell_2^2) + m_C(a_C^2+b_C^2)]/2 + m_R(r-\ell_1/2)^2 + m_S(D-\ell_2/2)^2 + m_L r^2 + m_T D^2 \qquad (5.1.12)$$

$$h(x_{p1}) = m_R(x_{p1} - \ell_1/2) + m_L x_p \qquad (5.1.13)$$

mi, i = R, S, C, T, L: masses of arm, sleeve, column, balance weight, and payload

ai and bi, i = R, S, C: the inner radius and the outer radius of arm, sleeve, and column

D, ℓ_1, ℓ_2 are as defined in Figure 5.1.1.

As can be seen from equation (5.1.11), the elements of matrix A are complicated nonlinear functions of x_p. Moreover the elements of matrices A and B are also functions of the mass of the payload, m_L which may be unknown and may change during the repititive task of the manipulator. Thus, the dynamic characteristics of the manipulator will change during a given task.

In order to obtain high performance in terms of speed and accuracy, it is necessary to use a control system which will consider the changes in the dyanamic characteristics of the manipulator. For this purpose, we apply the discrete-time MRAC system developed in the previous chapter. To accomplish this goal, we need a discrete-time model of the manipulator.

In order to design the specific structure of the adaptive controller a discrete-time model can be approximated by applying the Euler method to the continuous time model in equation (5.1.9). It is further assumed that:

(1) A and B are slowly varing compared to the adaptation speed.

(2) All inputs and disturbances acting on the manipulator are constants or staircase inputs.

Thus, an approximate discrete-time model is given by the following difference equation:

$$x_p(k+1) = (I + TA)x_p(k) + T[Bu_p(k) + W(k)] \qquad (5.1.14)$$

where T is the sampling time interval and k means the k-th sampling time step.

Defining the matrices A_p and B_p, and the vector W^* as follows:

$$A_p = \left[\begin{array}{c|c} I & TI \\ \hline TA_1 & (I+TA_2) \end{array}\right]$$

$$B_p = \left[\begin{array}{c} 0 \\ \hline Bp_1 \end{array}\right] \qquad \text{where } Bp_1 = TB_1 \qquad (5.1.15)$$

$$W^* = \left[\begin{array}{c} 0 \\ \hline W_1^* \end{array}\right] \qquad \text{where } W_1^* = TW_1$$

equation (5.1.14) can be written as

$$x_p(k+1) = A_p x_p(k) + B_p u_p(k) + W^*(k) \qquad (5.1.16)$$

5.2 Discrete-time Explicit MRAC Law

In the previous section, we have obtained an approximate discrete-time model equation for the robotic manipulator. We are now ready to design the discrete-time MRAC system using the design procedure shown in chapter 4.

It should be noted that in comparing equation (5.1.16) to equation (4.2.1), the former is written as a relation between the state x_p and the input u_p in vector form and also has the added term W^* due to the measurable disturbance W with plant time delay d=1. Considering these points, the discrete-time MRAC system will be described.

The tracking dynamics are given by a linear reference model and since it has to be structurally similar to the manipulator model, it is chosen to be:

$$x_M(k+1) = A_M x_M(k) + B_M u_M(k) \tag{5.2.1}$$

where

$$A_M = \left[\begin{array}{c|c} I & TI \\ \hline A_{M1} & A_{M2} \end{array}\right] \qquad B_M = \begin{bmatrix} 0 \\ B_{M1} \end{bmatrix} \tag{5.2.2}$$

A linear control can be found to satisfy the following objective:

$$C_R(q^{-1})[x_p(k+1) - x_M(k+1)] = 0 \qquad (5.2.3)$$

where

$$C_R(q^{-1}) = I + C_R^{*}q^{-1} \qquad (5.2.4)$$

which defines the regulation dynamics and has a determinant with all zeros inside the unit circle.

Considering the structure of the matrix A_p, we can design independent tracking and regulation control laws for the manipulator system where the matrix C_R^{*} in the following form:

$$C_R^{*} = \left[\begin{array}{c|c} -I & -TI \\ \hline \mathrm{diag}(c_{1i}) & \mathrm{diag}(c_{2i}) \end{array}\right] \quad i=1,2,3 \qquad (5.2.5)$$

Thus, the linear control law is given by:

$$u_p(k) = B_0[C_R(q^{-1})x_M(k+1) - Rx_p(k) - w^{*}(k)] \qquad (5.2.6)$$

where

$$B_0 = [0 \mid B_{p1}^{-1}]$$

$$R = C_R^{*} + A_p \qquad (5.2.7)$$

Using this control law, the filtered error,

$$e_0^{f}(k+1) = C_R(q^{-1})[x_p(k+1) - x_M(k+1)] \qquad (5.2.8)$$

is equal to zero when the parameters of the manipulator dynamic equation are known and the controller parameters have the values given by this equation. However, since A_p, B_p and W^* are time-varying and include the terms involving payload inertia which is often unknown, the filtered error $e_0^f(k+1)$ will not actually be zero anymore. This value of filtered error is used as the input to an adaptation mechanism for the controller parameters with the following purpose:

$$\lim_{k\to\infty} e_0^f(k+1) = 0 \qquad (5.2.9)$$

Defining $W^* = -V^* d^*$ where $V^* = [\ 0\ 0\ 0\ v_1^*\ v_2^*\ v_3^*\ ]$ and d^* are arbitrary factors, the control law in the adaptive case will be chosen as:

$$u_p(k) = \hat{B}_0(k)[C_R(q^{-1})x_M(k+1) - \hat{R}(k)x_p(k) + \hat{V}^*(k)d^*] \qquad (5.2.10)$$

where ^ denotes evaluation using estimates of the parameters.

From equation (5.2.3) and (5.2.10),

$$e_0^f(k+1) = C_R(q^{-1})[x_p(k+1) - x_M(k+1)]$$
$$= [\Theta - \hat{\Theta}(k)]^T \phi(k) \qquad (5.2.11)$$

where

$$\Theta^T = [B_p \; R \; V^*]$$

$$\phi(k)^T = [u_p(k)^T \; x_p(k)^T \; d^*] \tag{5.2.12}$$

Note that

$$e_0^f(k+1) = -\tilde{\Theta}(k)^T \phi(k) \tag{5.2.13}$$

where

$$\tilde{\Theta}(k) = \hat{\Theta}(k) - \Theta \tag{5.2.14}$$

is the a priori adaptation error which is assumed measurable and

$$e^f(k+1) = -\tilde{\Theta}(k+1)^T \phi(k) \tag{5.2.15}$$

is the a posteriori adaptation error. These error equations (5.2.13) and (5.2.15) have the same forms as equations (4.3.2) and (4.3.8) except the additional terms V^* and d in and $\phi(k)$. Therefore from equations (4.3.7), (4.3.9) and (4.3.11) the following adaptation algorithm is obtained directly:

$$\hat{\Theta}(k+1) = \tilde{\Theta}(k) + F(k)\phi(k)e^f(k+1)^T \tag{5.2.16}$$

with the relationship between $e_0^f(k+1)$ and $e^f(k+1)$:

$$e^f(k+1) = \frac{e_0^f(k+1)}{[1 + \phi(k)^T F(k)\phi(k)]} \tag{5.2.17}$$

where the adaptation gain matrix,

$$F(k+1) = \frac{1}{\lambda_1(k)}\left[F(k) - \frac{F(k)\phi(k)\phi(k)^T F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)} + \phi(k)^T F(k)\phi(k)} \right]$$

$$F(0) > 0,\ 0 < \lambda_1(k) \le 1,\ 0 \le \lambda_2(k) < 2$$

(5.2.18)

In ref. [50], various types of adaptation gain matrices were discussed by the different choices of $\lambda_1(k)$ and $\lambda_2(k)$ It has been shown that by using an adaptation gain matrix obtained by choosing $\lambda_1(k)$ and $\lambda_2(k)$ so that trace[F(k)] = constant, an improved performance of the MRAC system can be achieved for a plant which has time-varying parameters. From equation (5.2.18) we can get the trace[F(k+1)] in the following form:

$$\text{trace}[F(k+1)] = \frac{1}{\lambda_1(k)}\text{trace}\left[F(k) - \frac{F(k)\phi(k)\phi(k)^T F(k)}{\delta(k) + \phi(k)^T F(k)\phi(k)}\right]$$

(5.2.19)

where

$$\delta(k) = \frac{\lambda_1(k)}{\lambda_2(k)} \qquad (5.2.20)$$

From equation (5.2.19) choosing $\delta(k)$ = const > 0, $\lambda_1(k)$ can be computed at each step such that trace[F(k+1)] has the desired value.

Since as mentioned in section 5.1 the manipulator equation (5.1.14) has time-varying and unknown parameters, the above stated approach was used to compute the adaptation gain matrix, F(k) in the simulation.

Further discussion of different adaptation gain matrices will not be included here. An interested reader may refer to reference [47].

We can summarize this discrete-time explicit MRAC system by the following equations:

Manipulator: $x_p(k+1) = A_p x_p(k) + B_p u_p(k) + W^*(k)$ (5.1.16)

Ref. model: $x_M(k+1) = A_M(k) x_M + B_M u_M(k)$ (5.2.1)

Control law: $u_p(k) = \hat{B}_0(k)[C_R(q^{-1})x_M(k+1) - \hat{R}(k)x_p(k) + \hat{V}^*(k)d^*]$ (5.2.10)

Adaptation mechanism:

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + F(k)\emptyset(k)e^f(k+1)^T \tag{5.2.16}$$

$$e^f(k+1) = \frac{e_o^f(k+1)}{[1 + \emptyset(k)^T F(k)\emptyset(k)]} \tag{5.2.17}$$

$$F(k+1) = \frac{1}{\lambda_1(k)}[F(k) - \frac{F(k)\emptyset(k)\emptyset(k)^T F(k)}{\delta + \emptyset(k)^T F(k)\emptyset(k)}] \tag{5.2.21}$$

$$\text{trace}[F(k+1)] = \frac{1}{\lambda_1(k)}\text{trace}[F(k) - \frac{F(k)\emptyset(k)\emptyset(k)^T F(k)}{\delta + \emptyset(k)^T F(k)\emptyset(k)}]$$

(5.2.22)

The diagram of this MRAC system is shown in Figure 5.2.1.

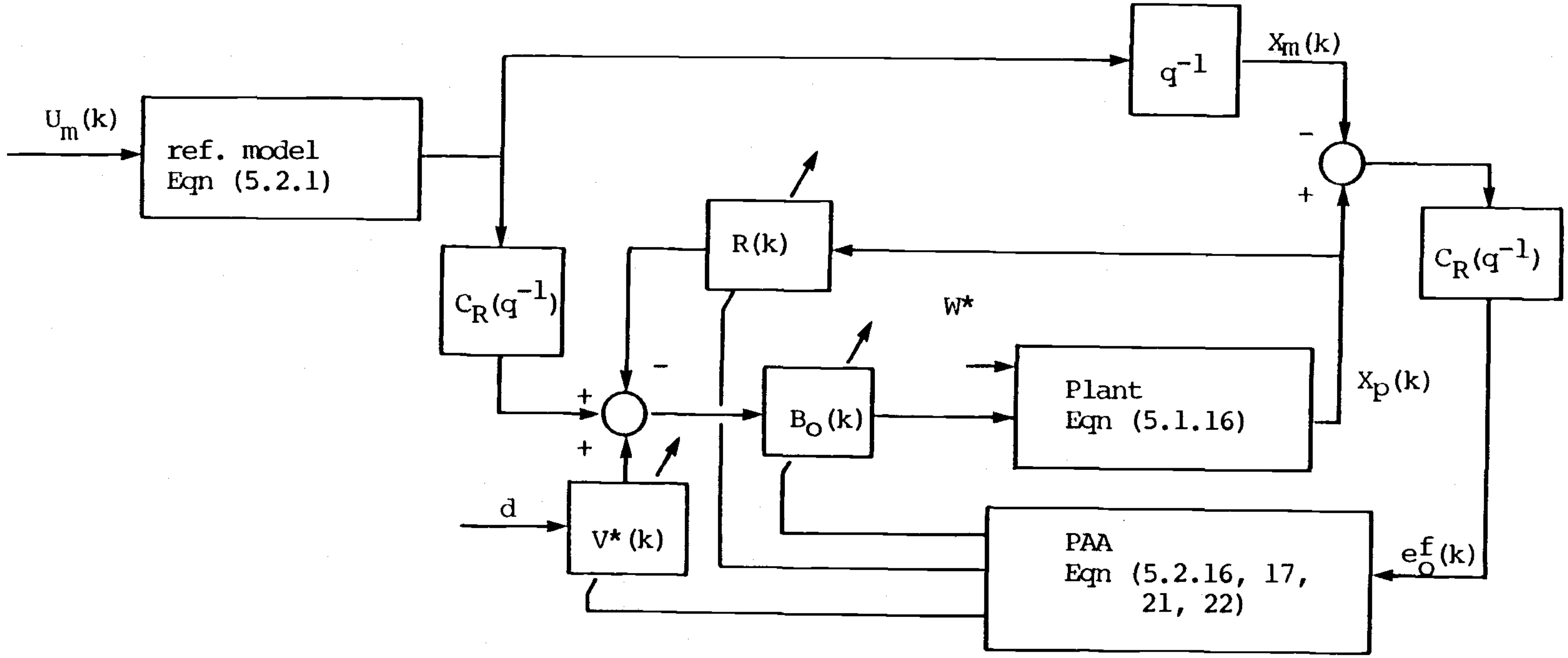

Figure 5.2.1 Diagram of MRAC System for Robotic Manipulator

5.3 Results

In order to investigate the effects of process noise and payload uncertainty on the MRAC system developed for the manipulator model (5.1.9), the manipulator was redescribed by the following equation which has the additional term W_S compared to equation (5.1.9):

$$\dot{x}_p(t) = A(x_p,t)x_p(t) + B(x_p,t)u_p(t) + W(x_p,t) + W_S \tag{5.3.1}$$

where $W_S = [0 \mid W_{S1}]$, the 3-dimensional vector W_{S1} represents an independent stochastic process disturbance with zero mean, and the adjustable controller of equation (5.2.10) was obtained initially assuming a 1 kg payload. The robot system described by equation (5.3.1) was then simulated using a 20 kg payload.

In this simulation:

(1) The noise vector W_{S1} was generated by a pseudo-random number generator [51] which gives nearly Gaussian random number.

(2) The reference input $u_M(k)$ was computed using a trapezoidal speed law (Figure 5.3.1) along the desired trajectory (Figure 5.3.2).

(3) The controller sampling time was 0.01 sec.

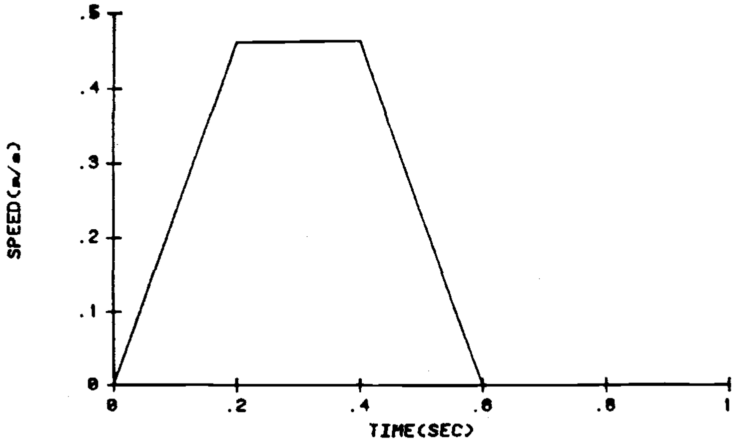

Figure 5.3.1 Speed Law (V_O=0.4616 m/s, a=2.308 m/s^2)

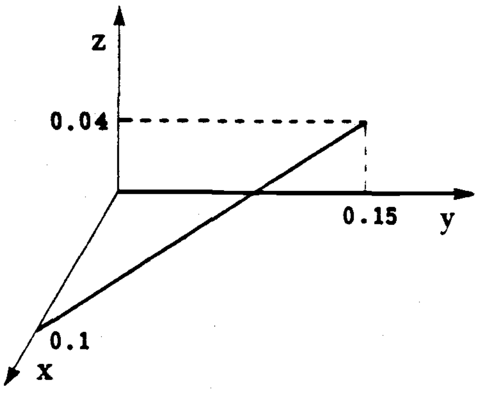

Figure 5.3.2 Desired Trajectory (unit: m)

(4) The manipulator dynamics were simulated by numerically integrating the complete set of nonlinear equations (5.3.1) with time interval 0.002 sec using a numerical technique found suitable for stiff nonlinear dynamic system equations [52].

The following numerical values were assumed from the specifications of the SEIKO Model PN-700:

Mass(kg): mass of sleeve, $m_S = 1.26$

mass of arm, $m_R = 0.37$

mass of upright column, $m_C = 1.97$

mass of balance weight, $m_T = 0.4$

Inner radius(m): $a_S = 0.0125$, Outer radius(m): $b_S = 0.019$

$a_R = 0.0075$ $b_R = 0.011$

$a_C = 0.02$ $b_C = 0.03$

Length of horizontal arm: $\ell_1 = 0.23$m

Length of horizontal sleeve: $\ell_2 = 0.25$m

Distance between center and balance weight: D = 0.17m

The marices and constants of equations (5.2.1), (5.2.5) and (5.2.18) used in the simulation are as follows:

$A_{M1} = (-188.01s^{-1})I$

$A_{M2} = -0.98I$

$B_{M1} = -A_{M1}$

$c_{1i} = (100/3)s^{-1}$

$c_{2i} = 1/6$

$\delta = 0.5$

$d^* = 1$ m/sec

$v_1^*(0) = v_2^*(0) = 0$

$v_3^*(0) = 0.098$

$F(0) = \text{diag}(f_i(0))$

where $f_1 = 0.0001\ (N^{-2})$

$f_2 = 0.01\ (N^{-2}m^{-2})$

$f_3 = 0.0001\ (N^{-2})$

$f_8 = 0.0001\ (s^2/rad^2)$

$f_{10} = 0.0001\ (s^2/m^2)$

$f_4 = f_5 = f_6 = f_7 = f_9 = 0$

With these values, the reference model (5.2.1) has the repeated eigenvalue z=0.01, which gives a small trajectory error, and regulation dynamics have the eigenvalues z=-0.5 and z=-0.3. With a different initial adaptation gain matrix F(0), no significant difference has been observed in the dynamic response. However, small values in the F(0) matrix are preferred since this provides slow but smooth parameter adaptation. See Appendix C for detailed description of the simulation model.

For the following three different operating tasks of the robot system with respect to payload and stochastic process noise:

Case 1: payload 1kg, no stochastic process noise

Case 2: payload 20kg, stochastic process noise N(0,0.1)

Case 3: payload 20kg, stochastic process noise N(0,1)

the following simulation results were obtained:

(1) Outputs (r, ψ, z) of the robot system shown in figures 5.3.4 to 12.

(2) Relative errors (EM1, EM2, EM3) between the outputs (r, ψ, z) of the reference model and outputs of robot system shown in figures 5.3.13 to 21.

(3) Absolute error (ET1) between trajectory and manipulator shown in figures 5.3.22 to 24, absolute error (ET2) along the trajectory shown in figures 5.3.25 to 27, and absolute error (ET3) on the perpendicular to the trajectory shown in figures 5.3.28 to 30.

(4) Input forces and torques (K_r, M_ψ, K_z) shown in figures 5.3.31 to 39.

In these figures, the solid lines show the results with adaptation while the dot lines show the results without adaptive control. Figure 5.3.3 shows the elements of the stochastic process noise vector which has zero mean and unity standard deviation, N(0,1).

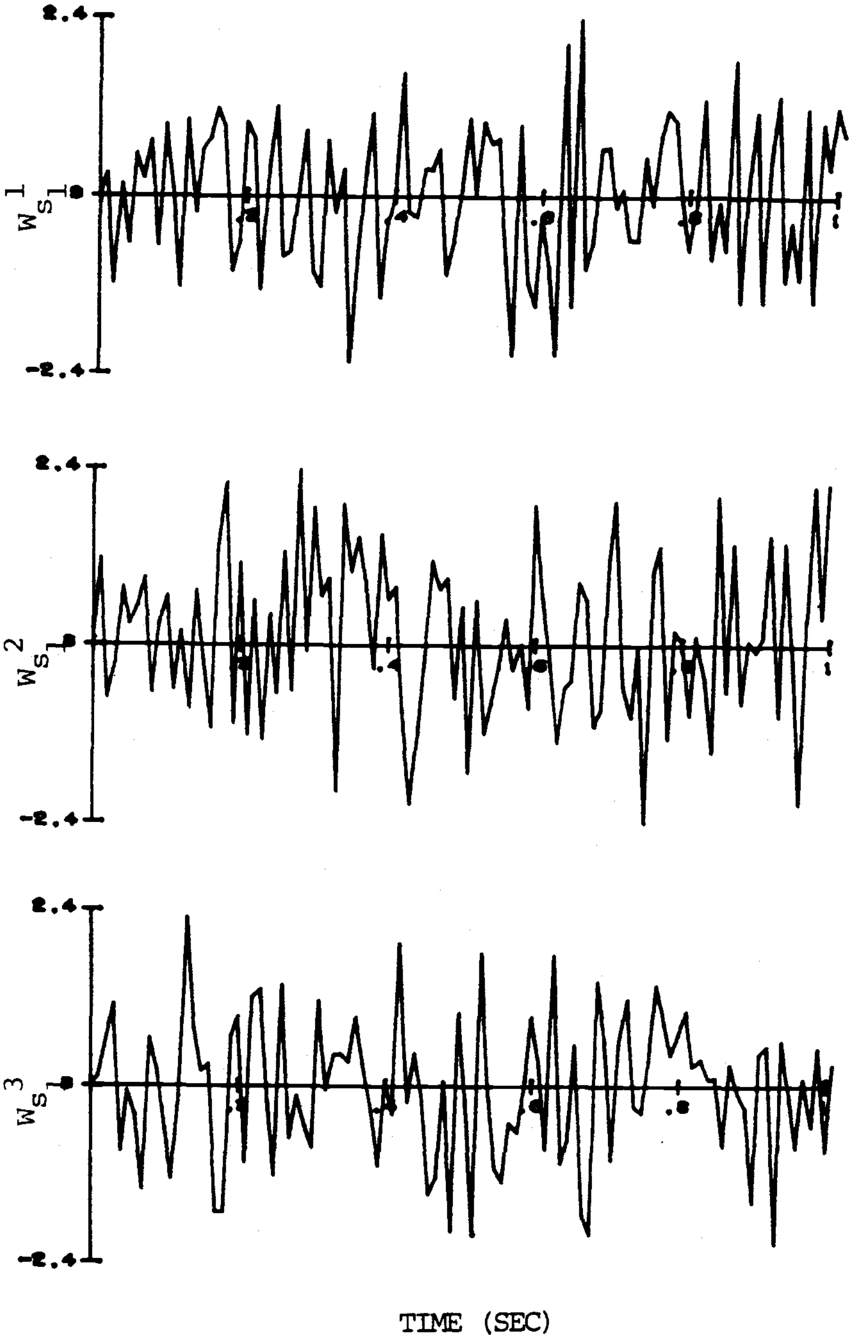

Figure 5.3.3 Stochastic Process Noise Vector W_{s_1}, N(0,1)

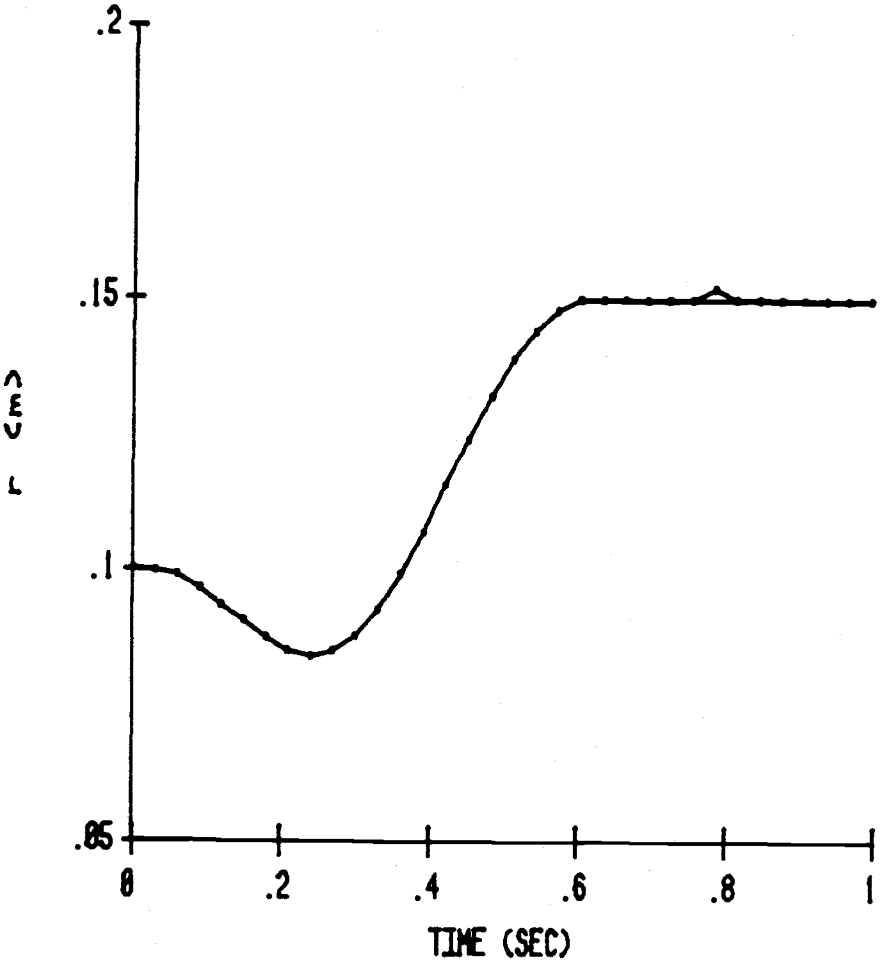

Figure 5.3.4 Arm Length (r) of Robot (case 1)

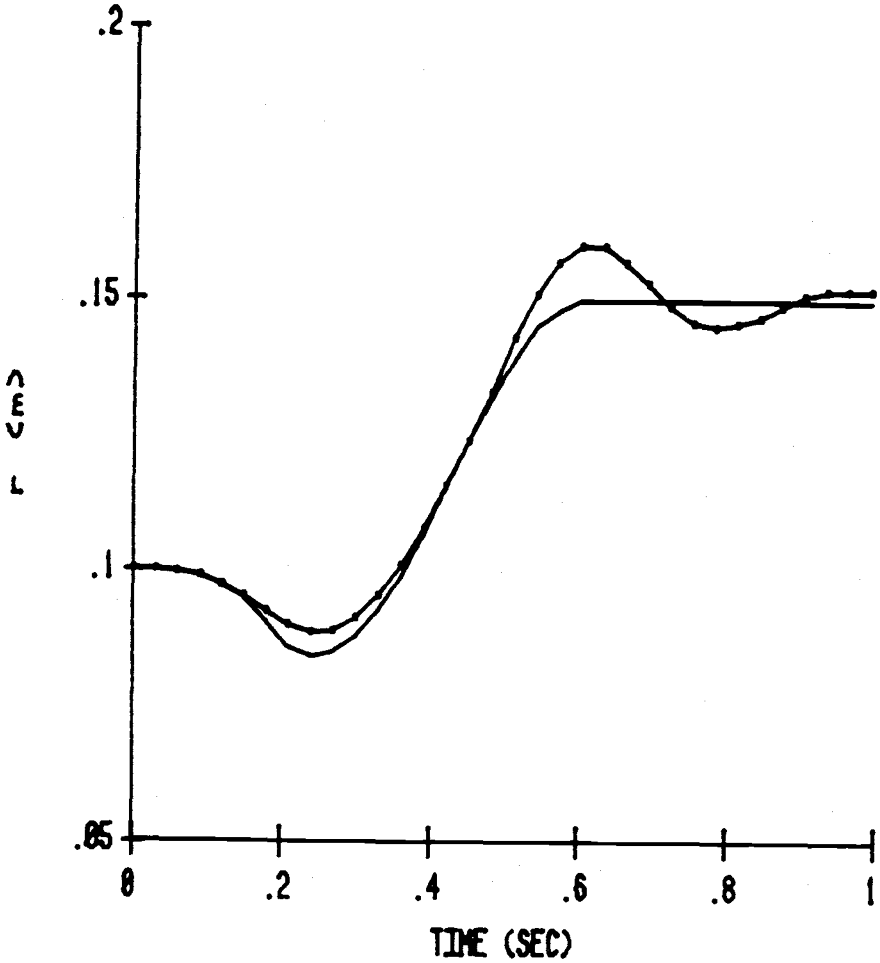

Figure 5.3.5 Arm Length (r) of Robot (case 2)

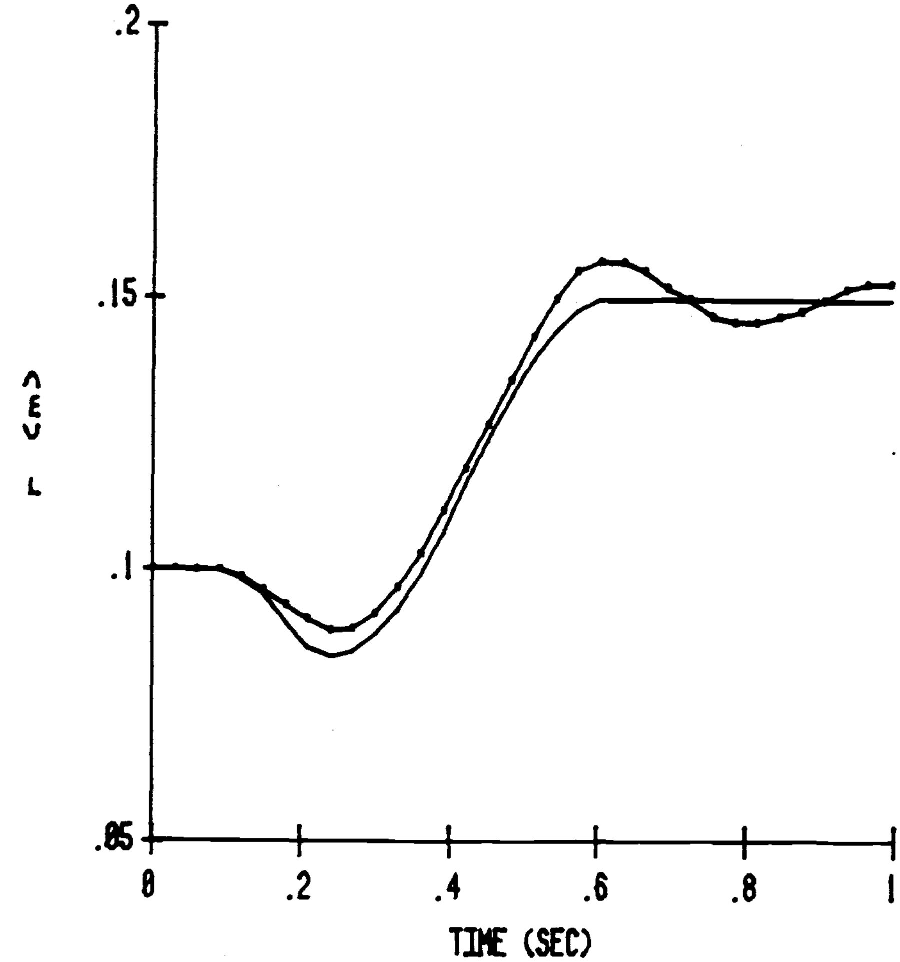

Figure 5.3.6 Arm Length (r) of Robot (case 3)

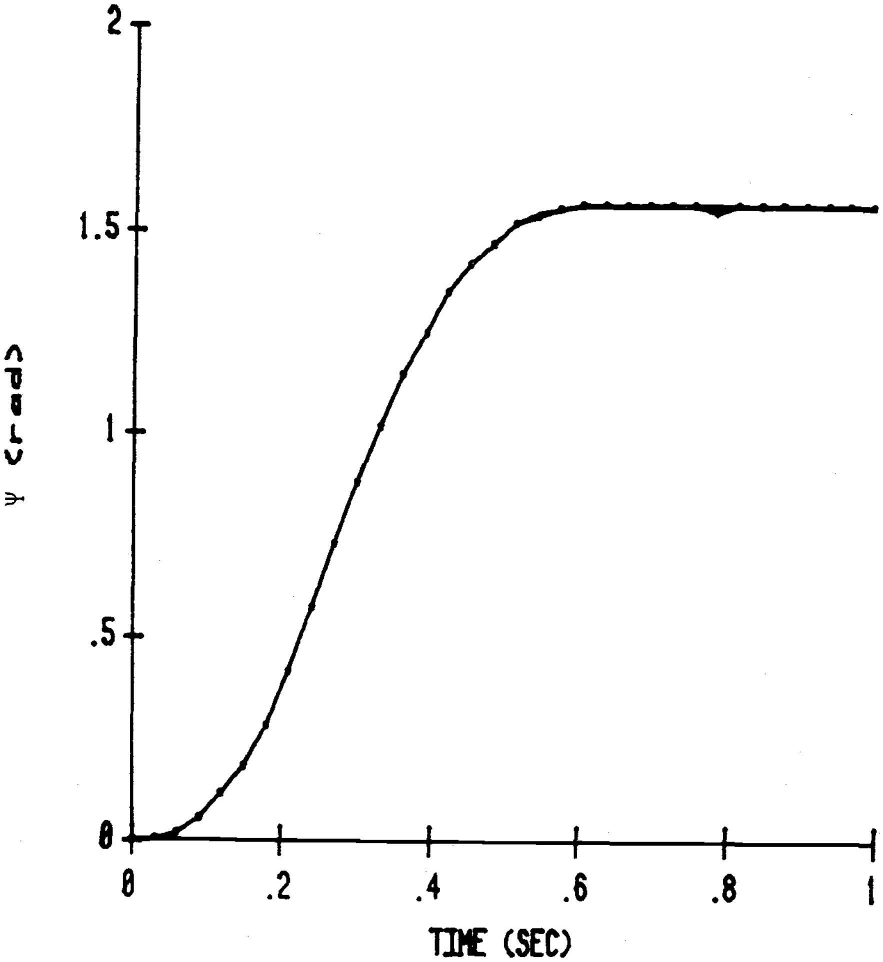

Figure 5.3.7 Arm Angle (Ψ) of Robot (case 1)

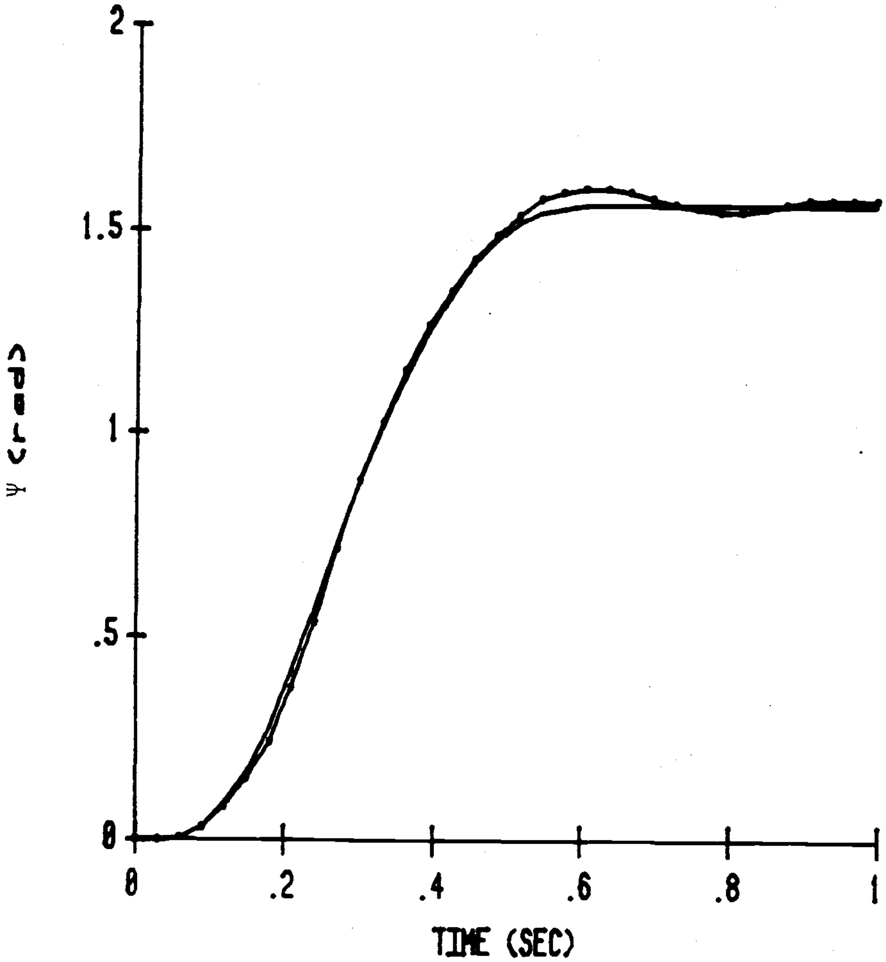

Figure 5.3.8 Arm Angle (Ψ) of Robot (case 2)

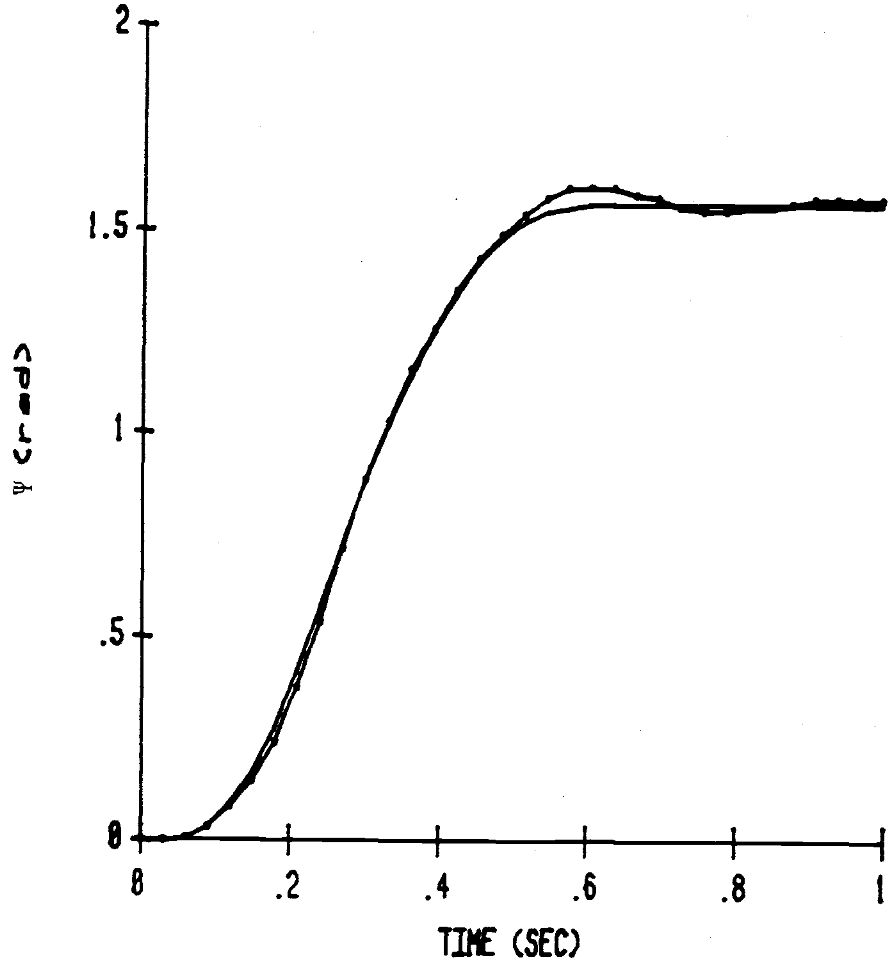

Figure 5.3.9 Arm Angle (Ψ) of Robot (case 3)

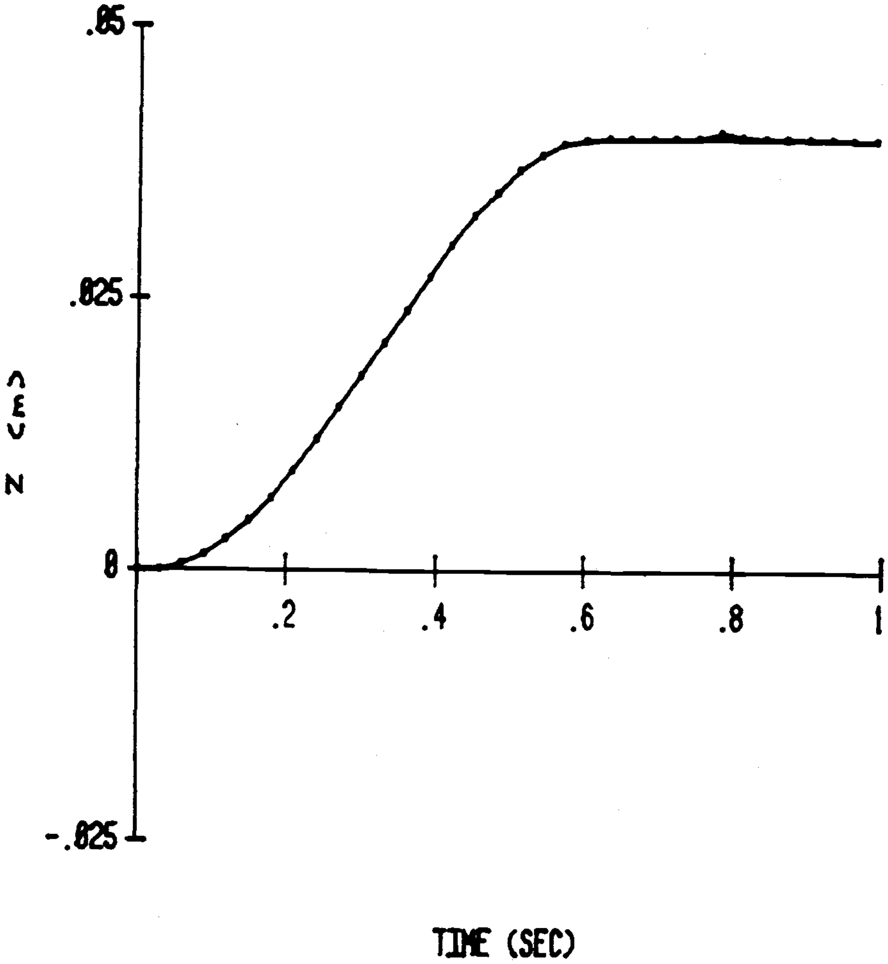

Figure 5.3.10 Arm Height (z) of Robot (case 1)

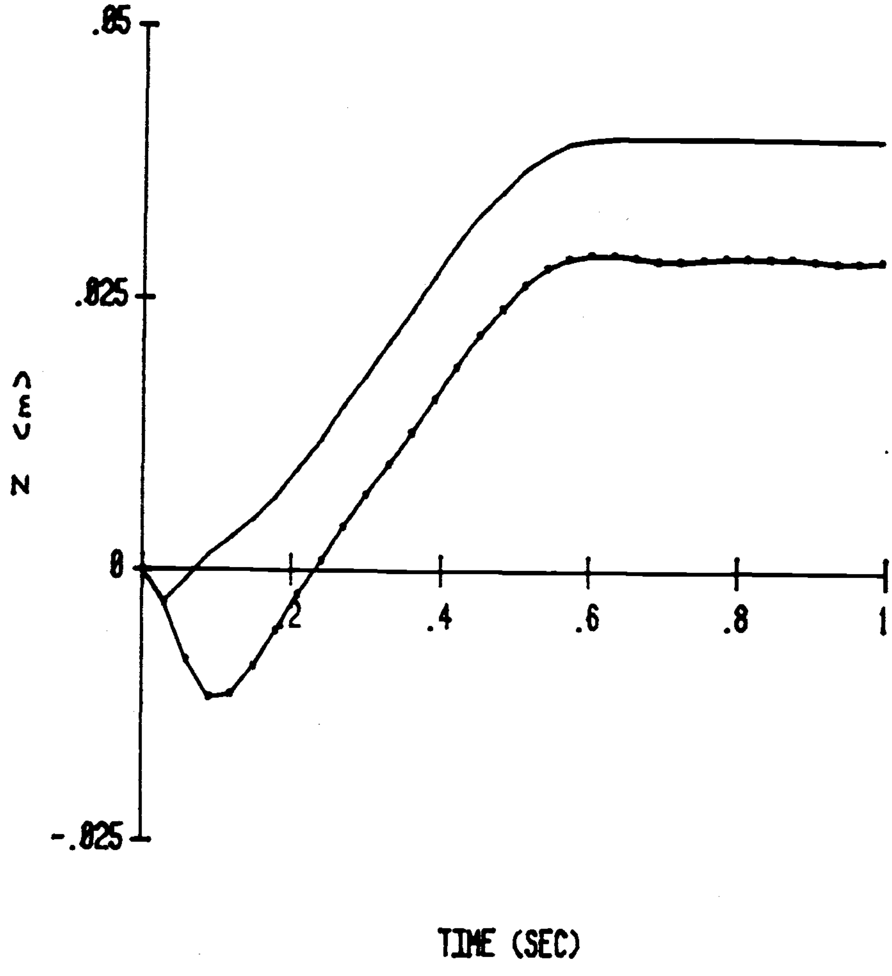

Figure 5.3.11 Arm Height (z) of Robot (case 2)

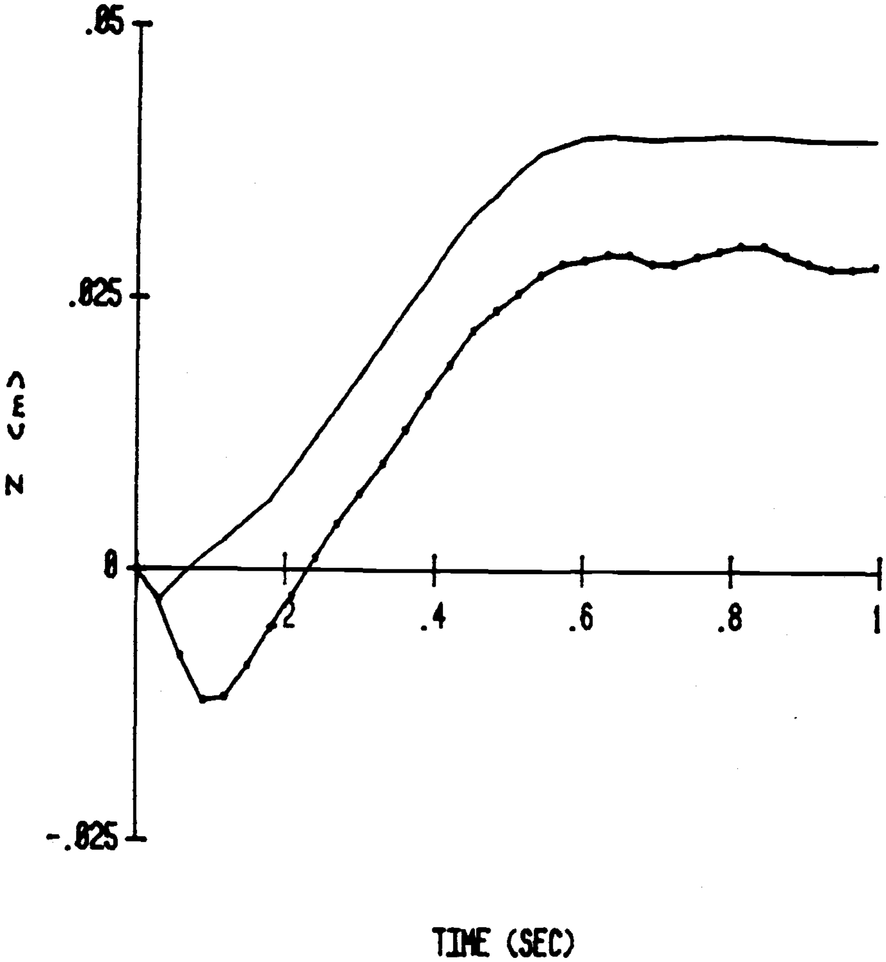

Figure 5.3.12 Arm Height (z) of Robot (case 3)

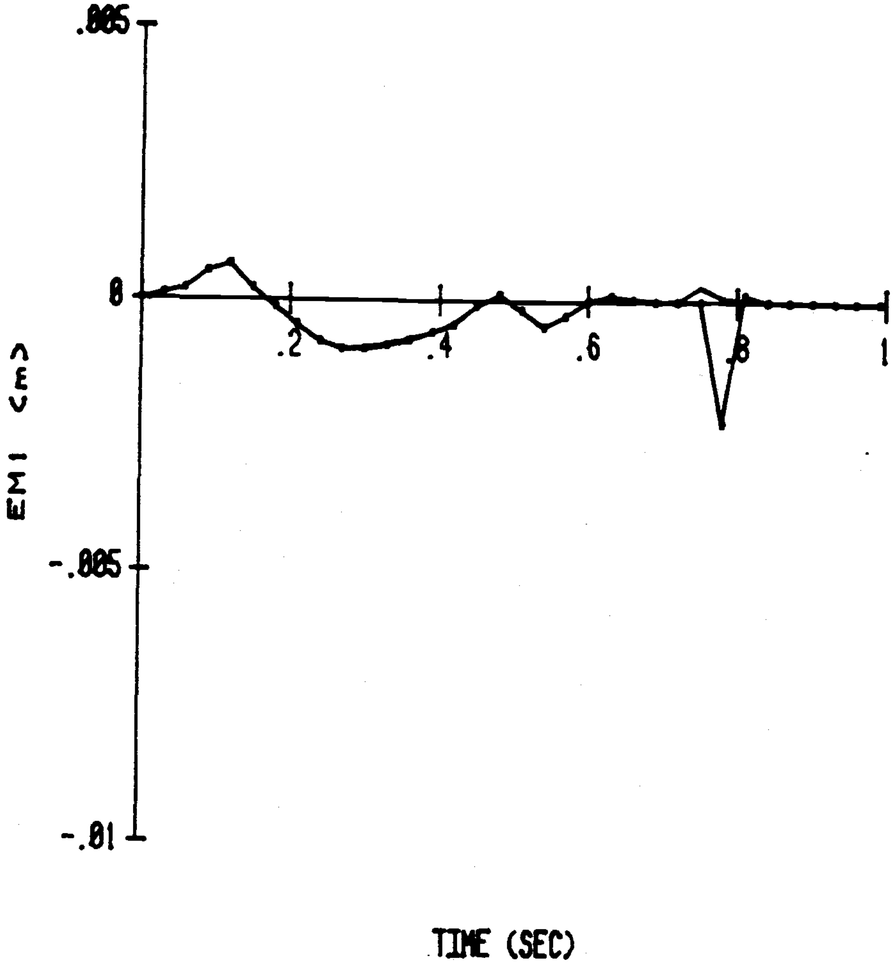

Figure 5.3.13 Relative Error (EM1) in Radius r (case 1)

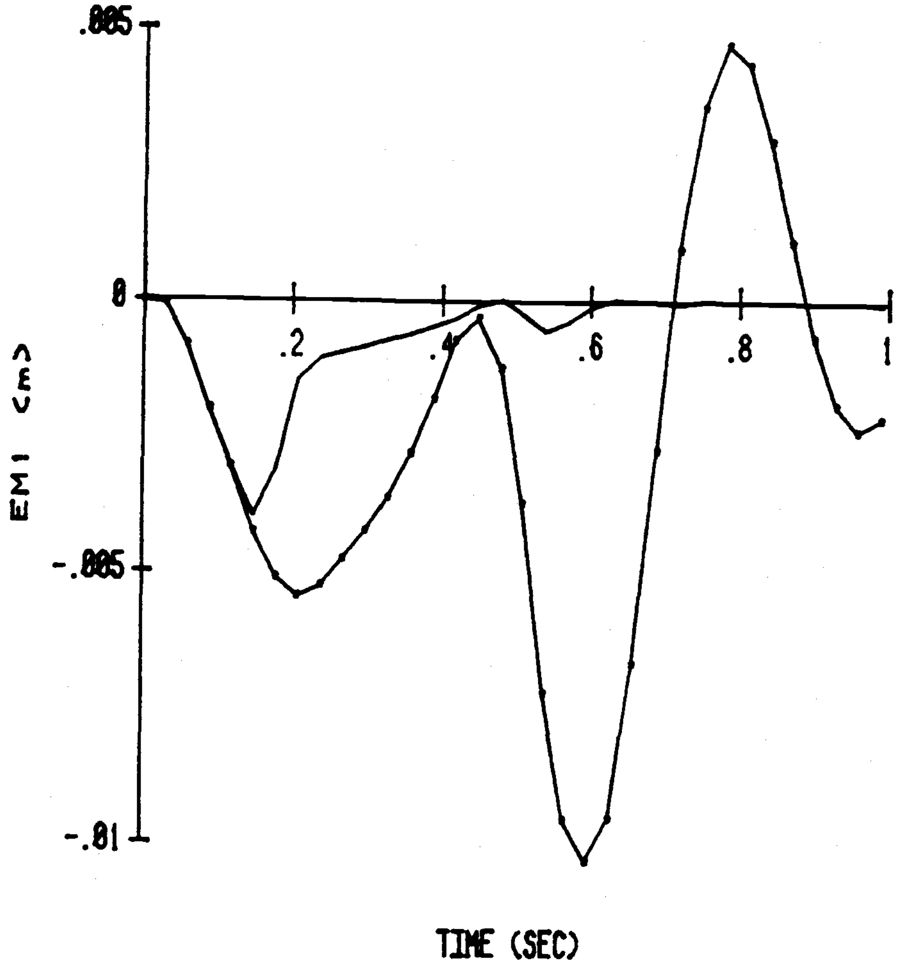

Figure 5.3.14 Relative Error (EM1) in Radius r (case 2)

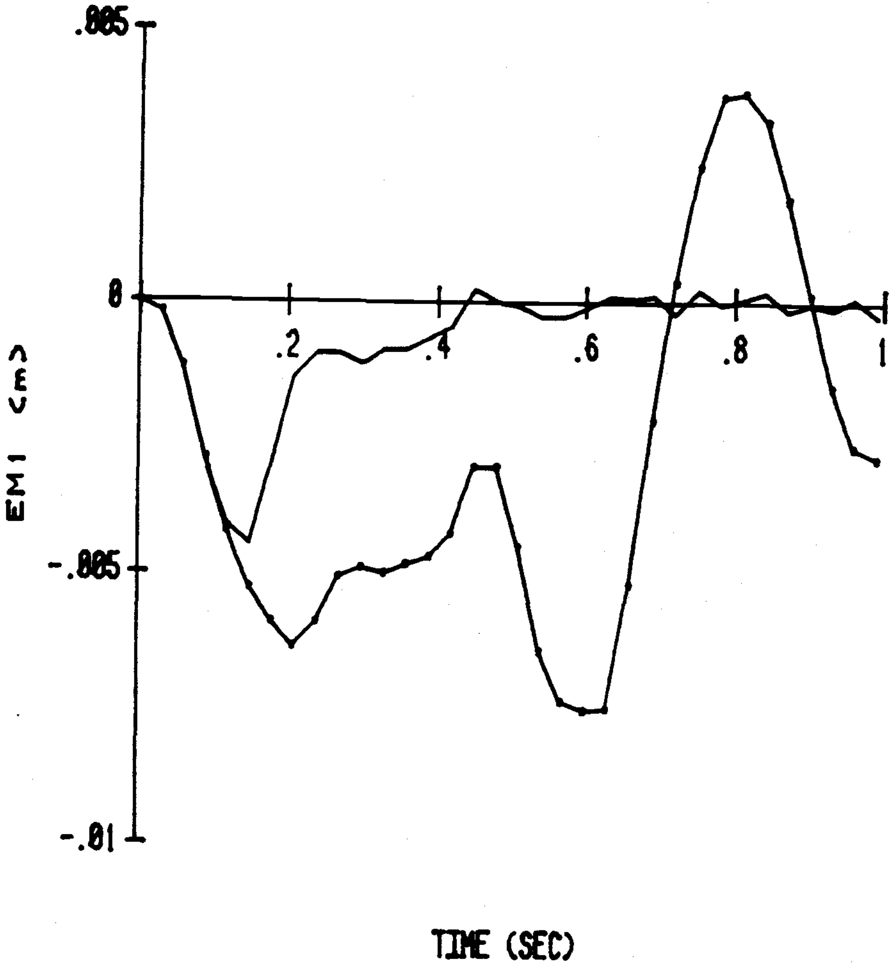

Figure 5.3.15 Relative Error (EM1) in Radius r (case 3)

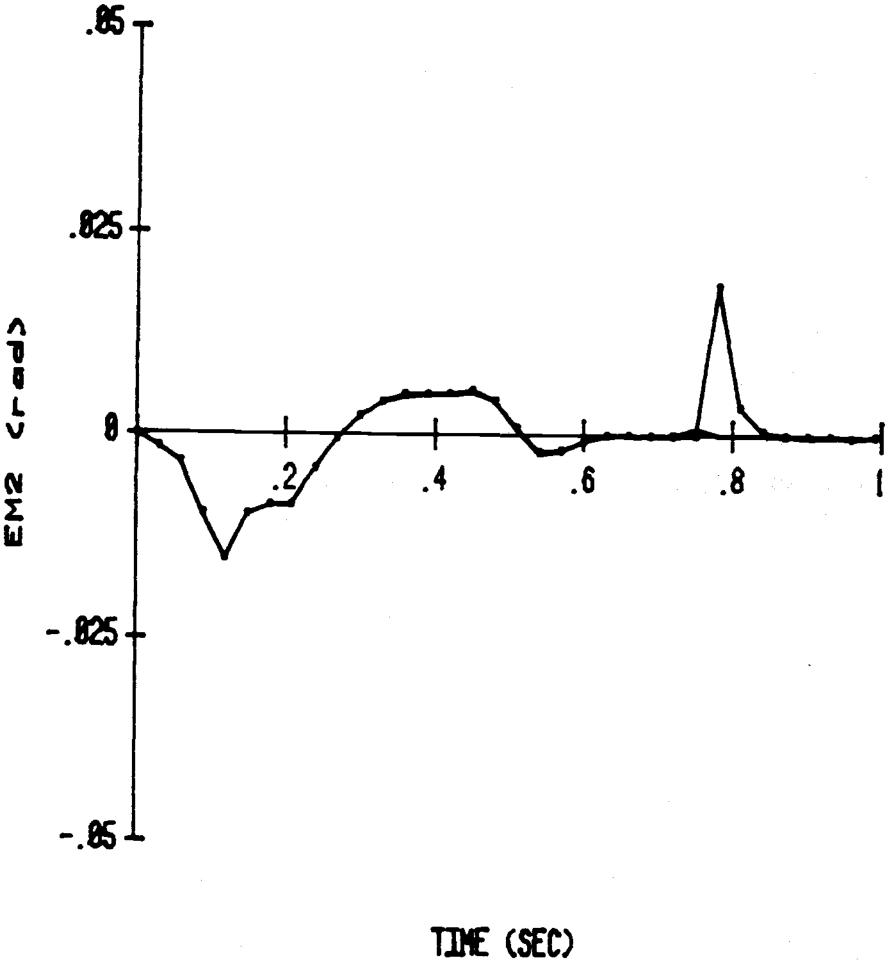

Figure 5.3.16 Relative Error (EM2) in Angle Ψ (case 1)

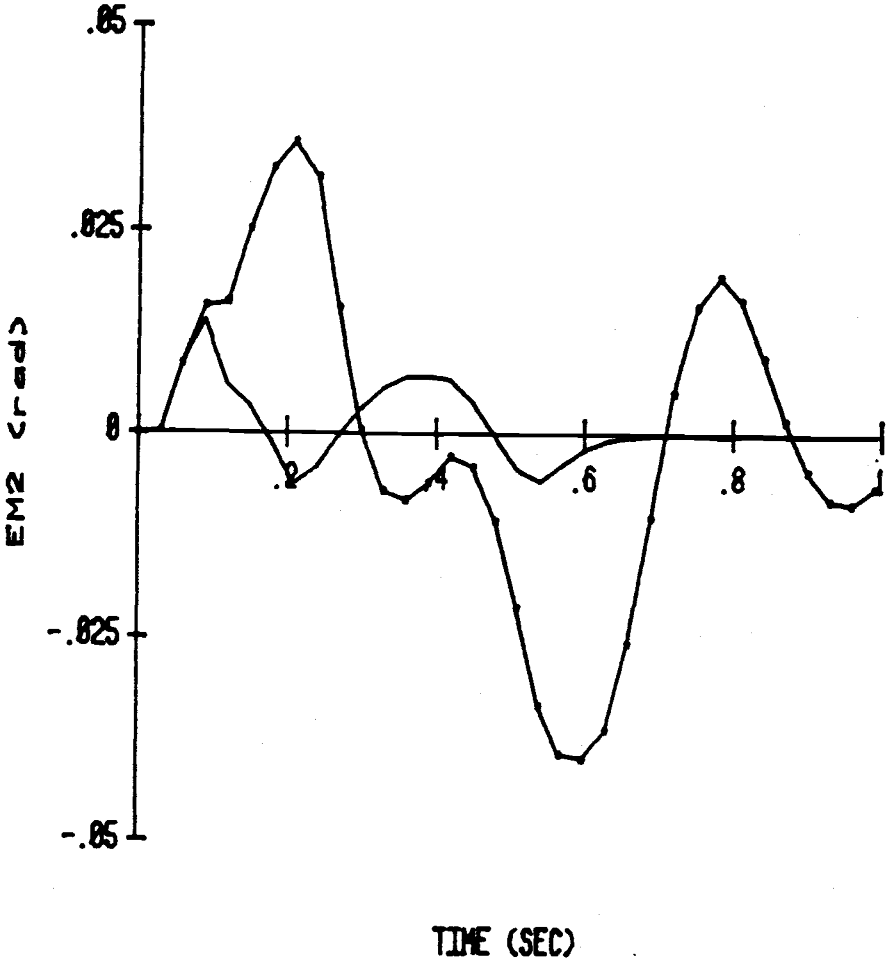

Figure 5.3.17 Relative Error (EM2) in Angle Ψ (case 2)

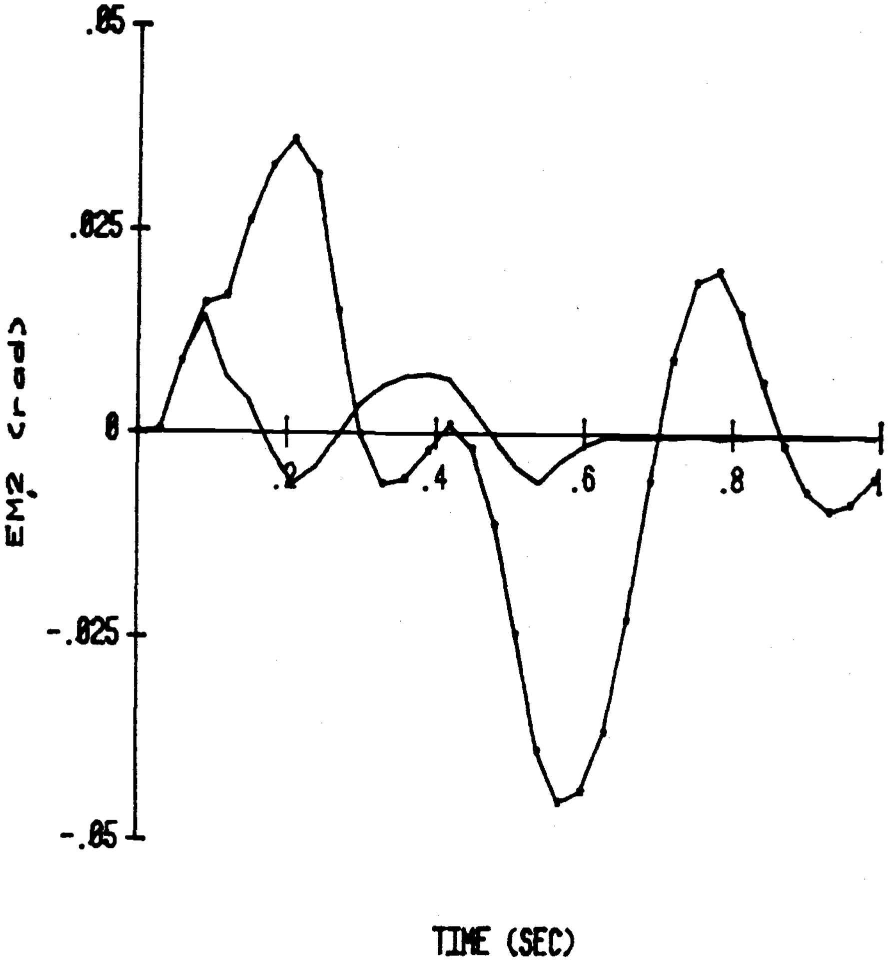

Figure 5.3.18 Relative Error (EM2) in Angle Ψ (case 3)

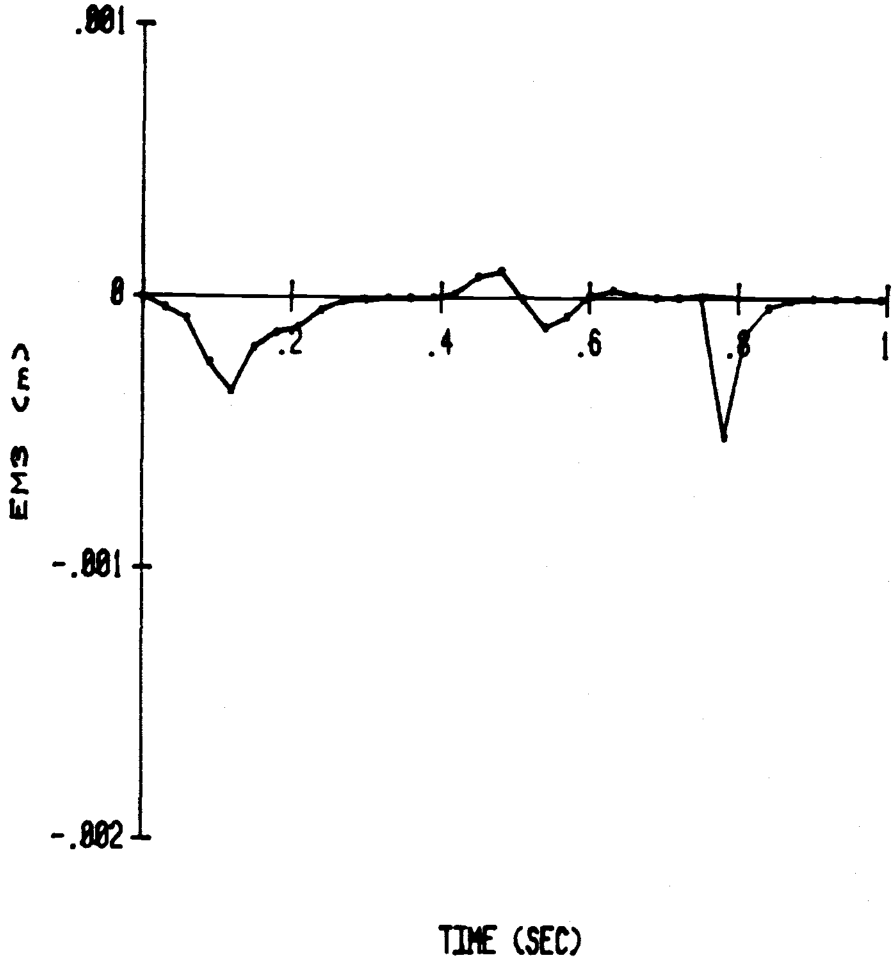

Figure 5.3.19 Relative Error (EM3) in Height z (case 1)

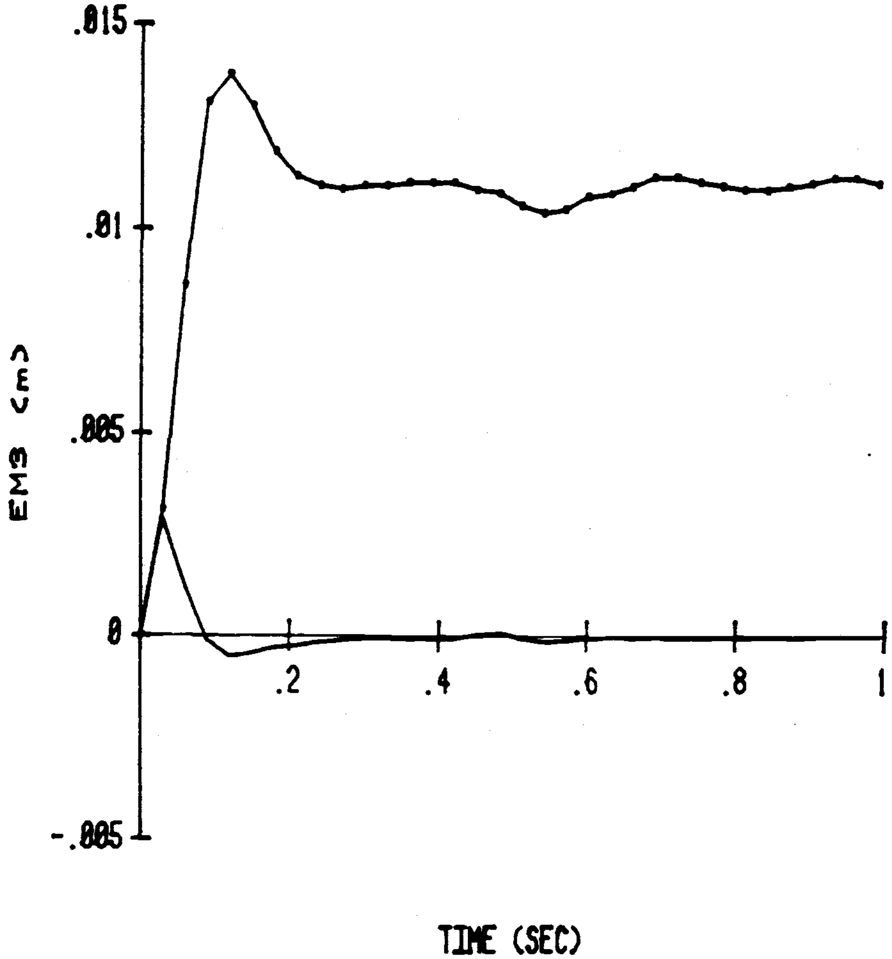

Figure 5.3.20 Relative Error (EM3) in Height z (case 2)

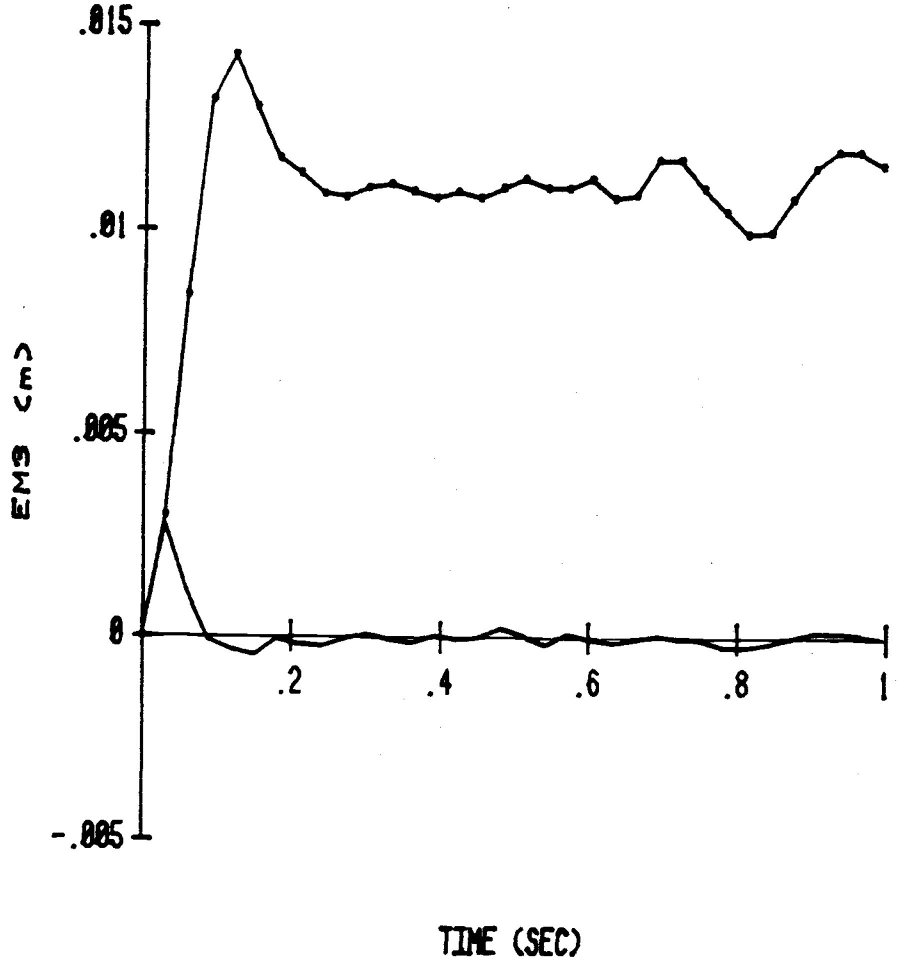

Figure 5.3.21 Relative Error (EM3) in Height z (case 3)

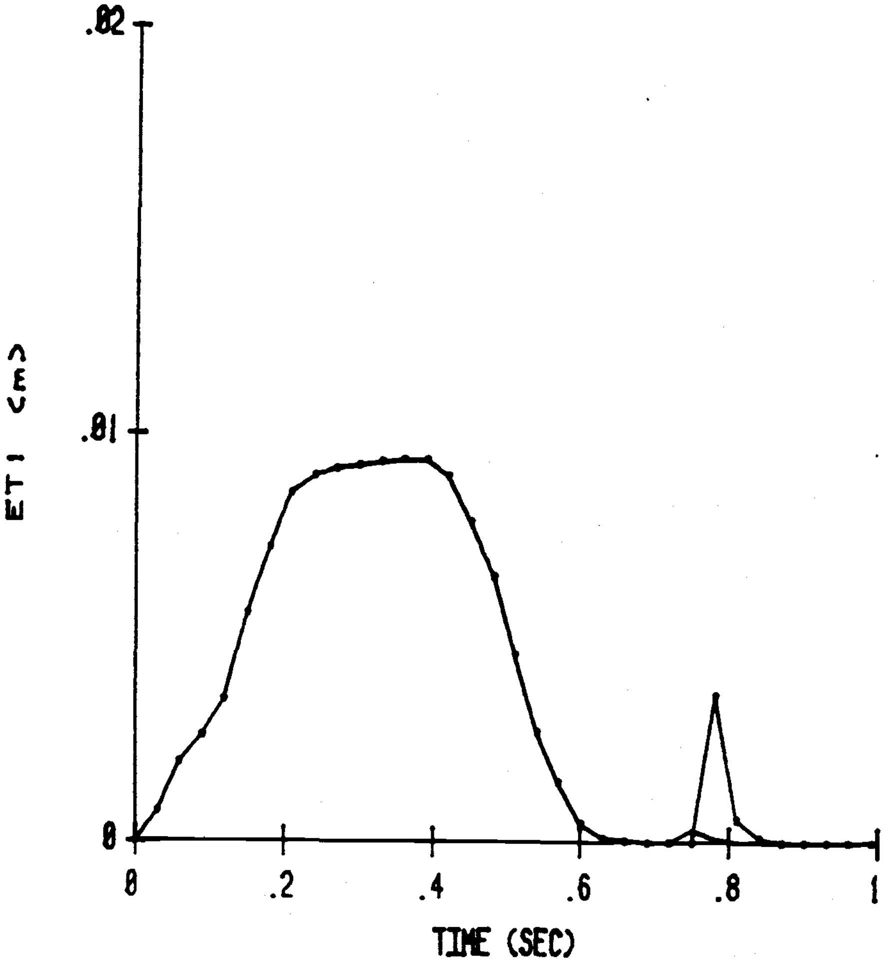

Figure 5.3.22 Absolue Error (ET1) between trajectory and manipulator (case 1)

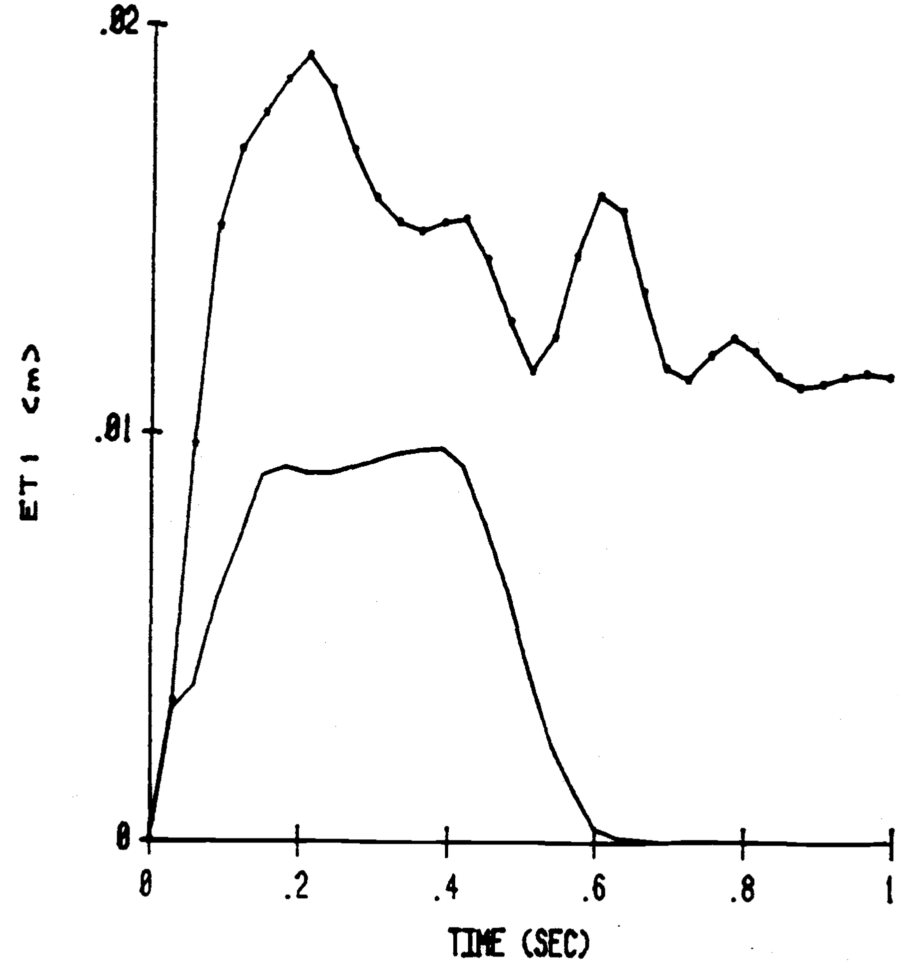

Figure 5.3.23 Absolute Error (ET1) between trajectory and manipulator (case 2)

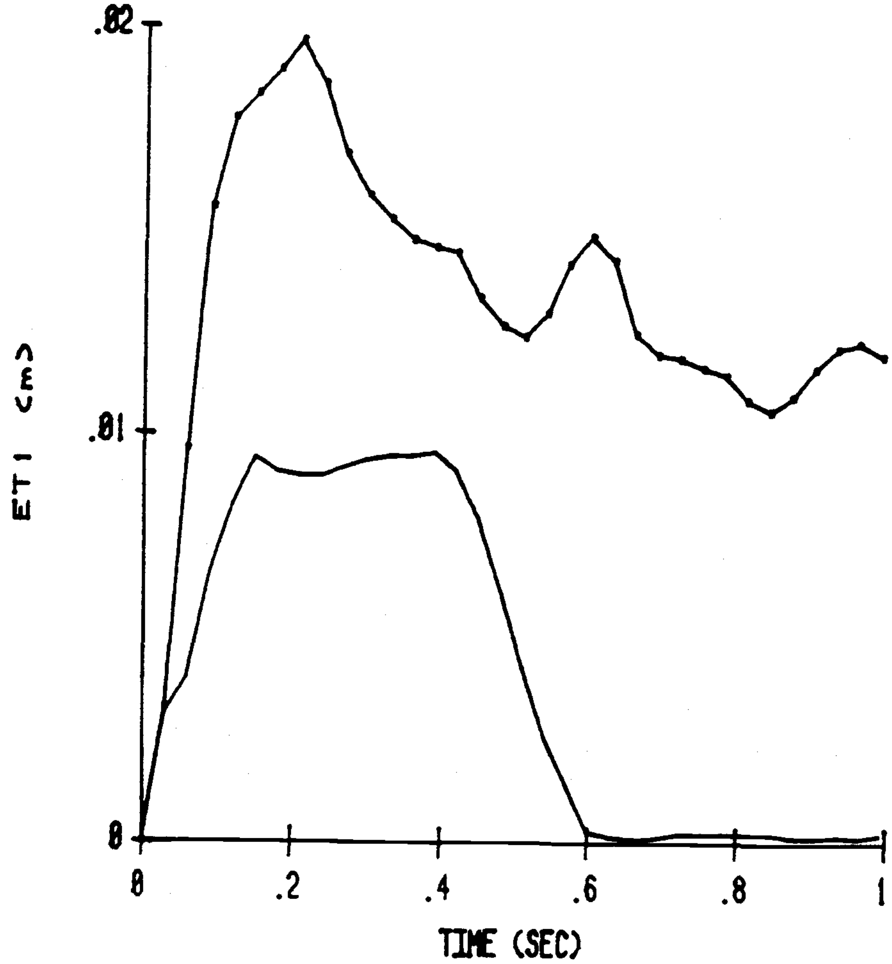

Figure 5.3.24 Absolute Error (ETl) between trajectory and manipulator (case 3)

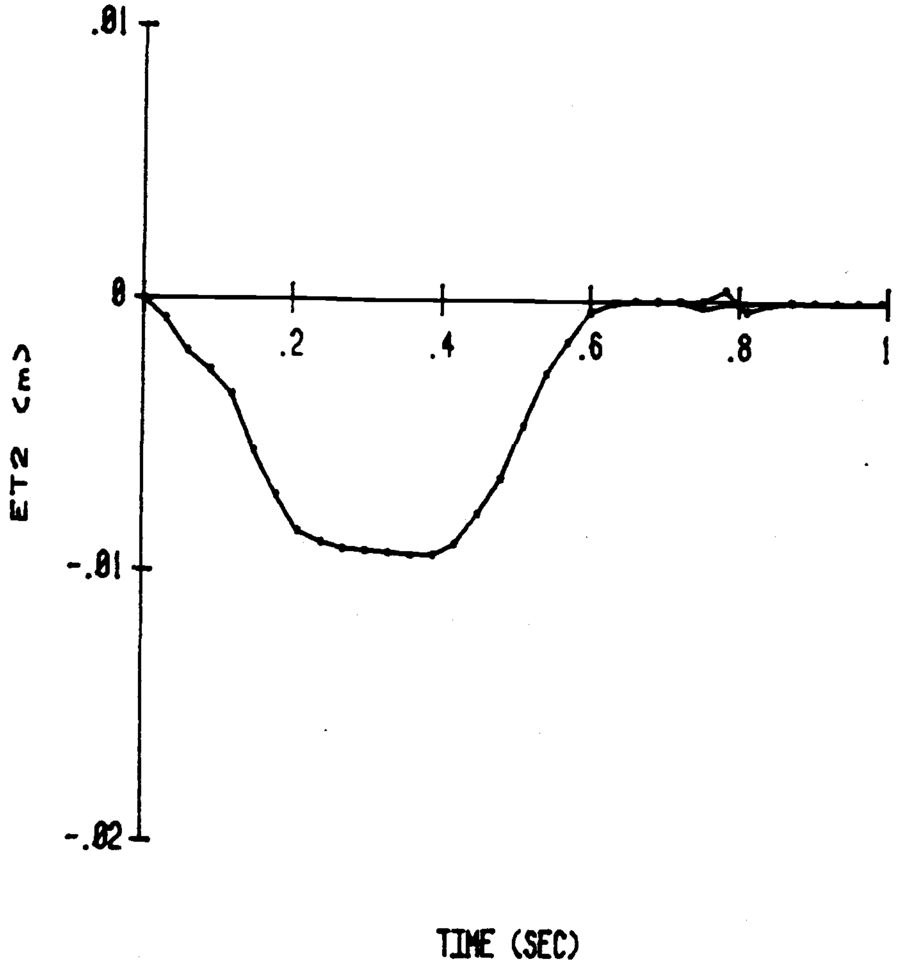

Figure 5.3.25 Absolute Error (ET2) along the Trajectory (case 1)

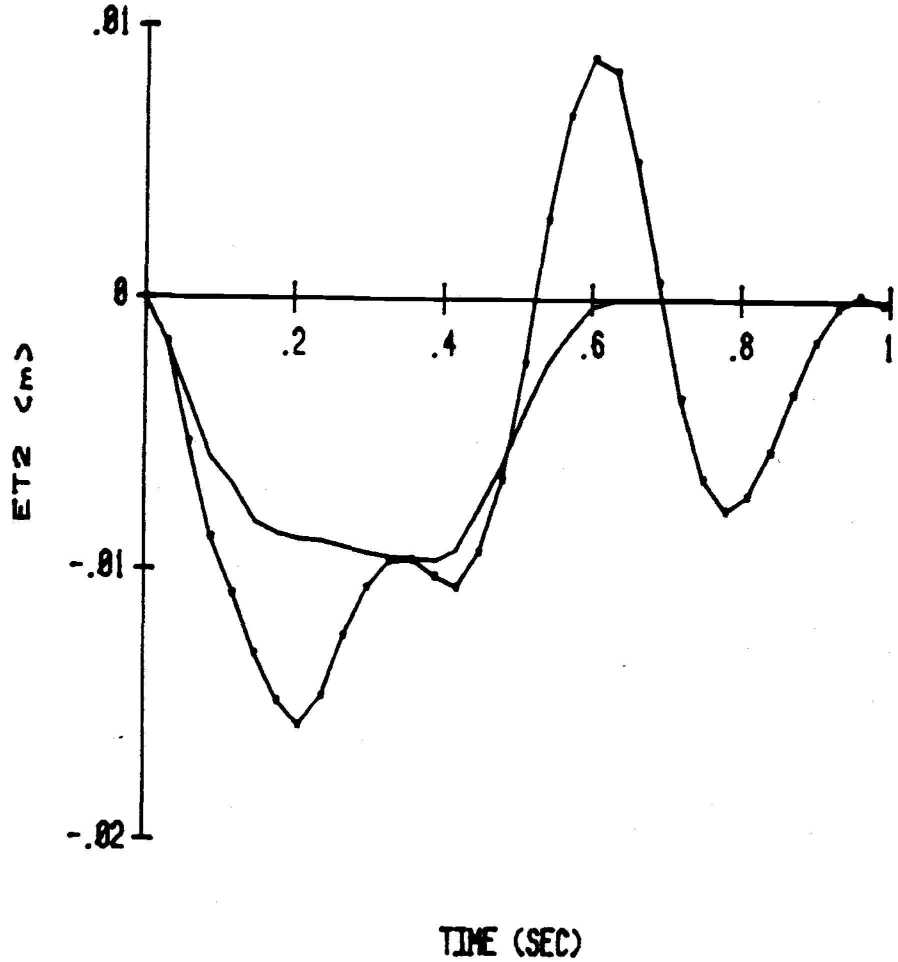

Figure 5.3.26 Absolute Error (ET2) along the Trajectory (case 2)

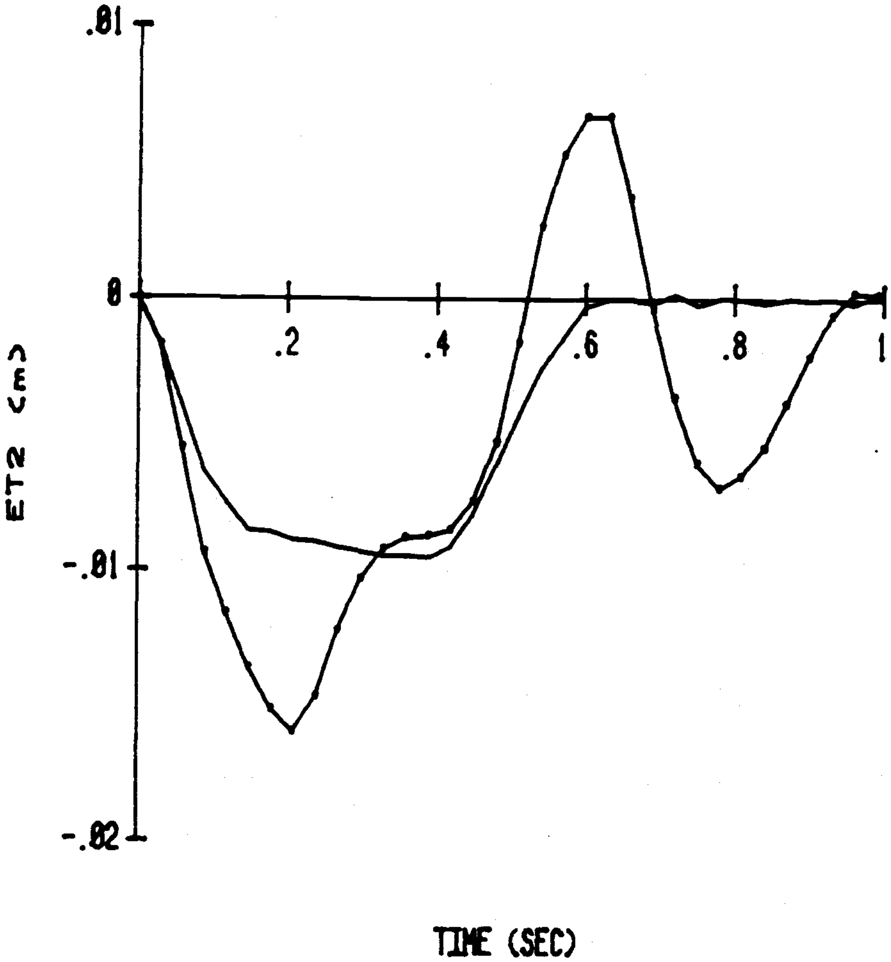

Figure 5.3.27 Absolute Error (ET2) along the Trajectory (case 3)

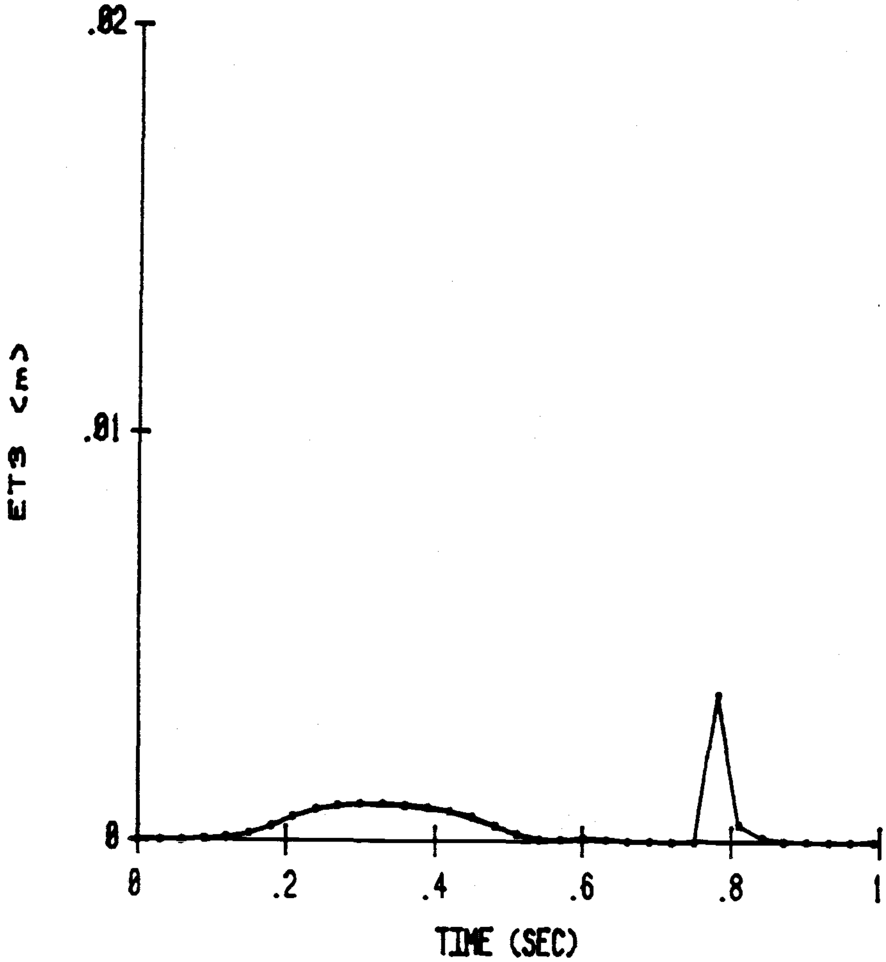

Figure 5.3.28 Absolute Error (ET3) on the perpendicular to the Trajectory (case 1)

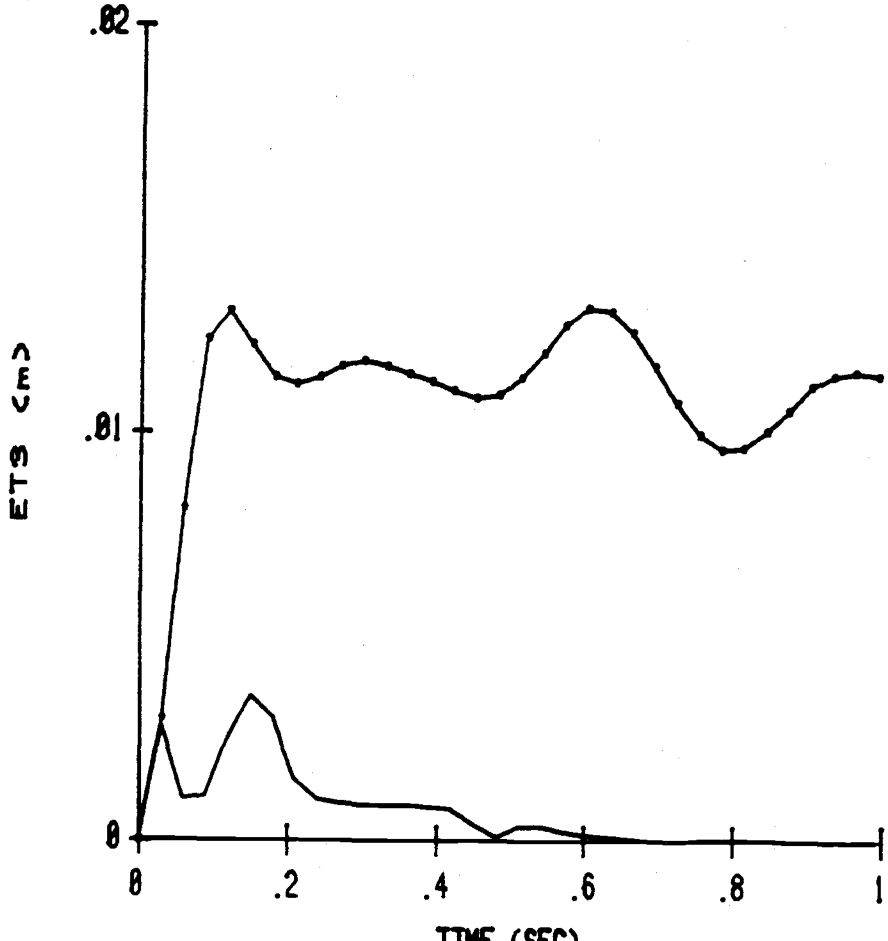

Figure 5.3.29 Absolute Error (ET3) on the perpendicular to the Trajectory (case 2)

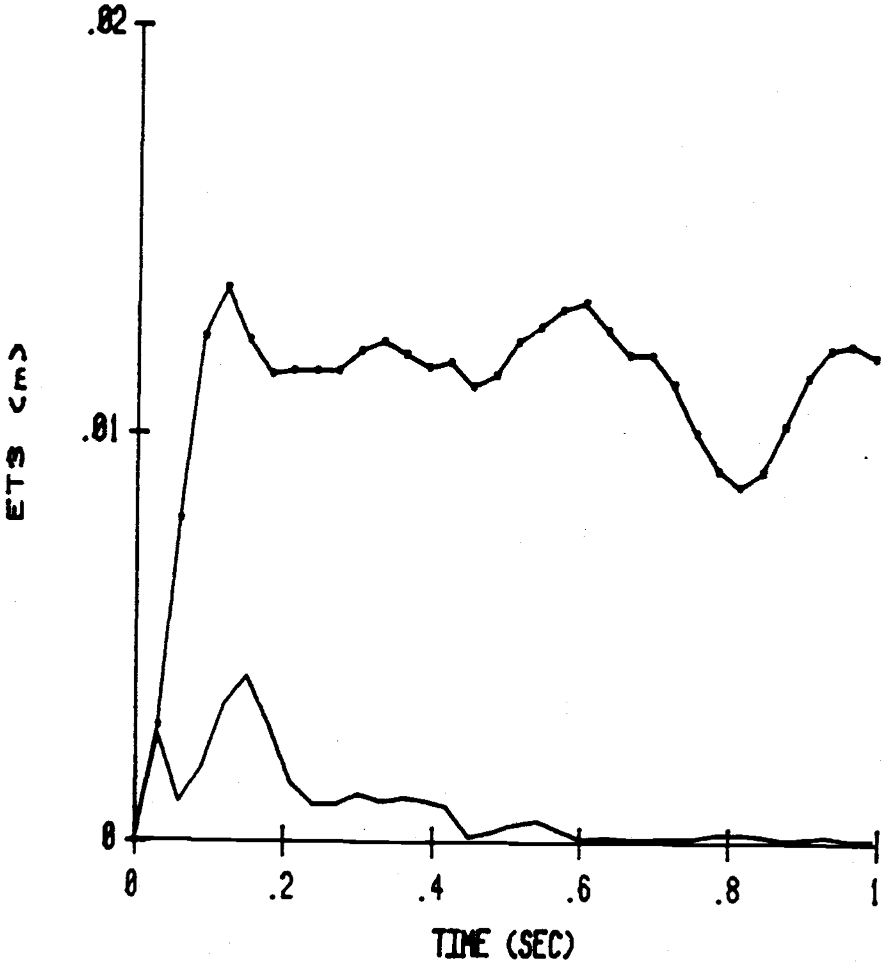

Figure 5.3.30 Absolute Error (ET3) on the perpendicular to the Trajectory (case 3)

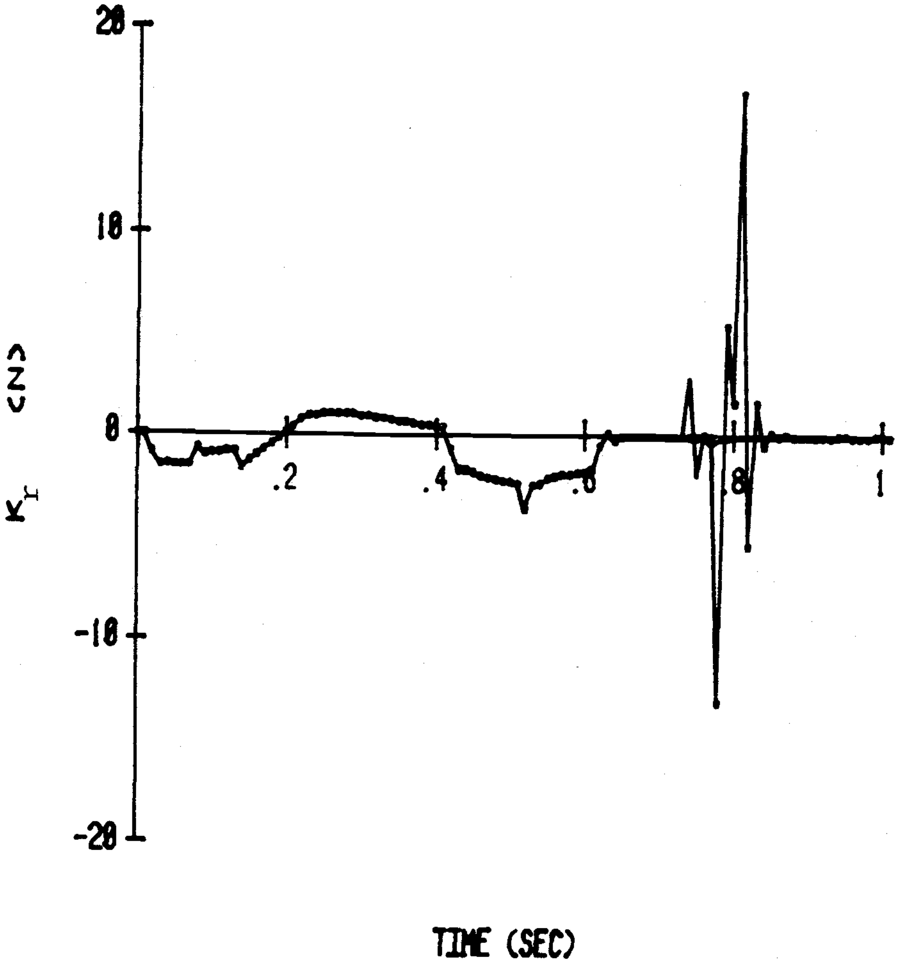

Figure 5.3.31 Inpute Force K_r. (case 1)

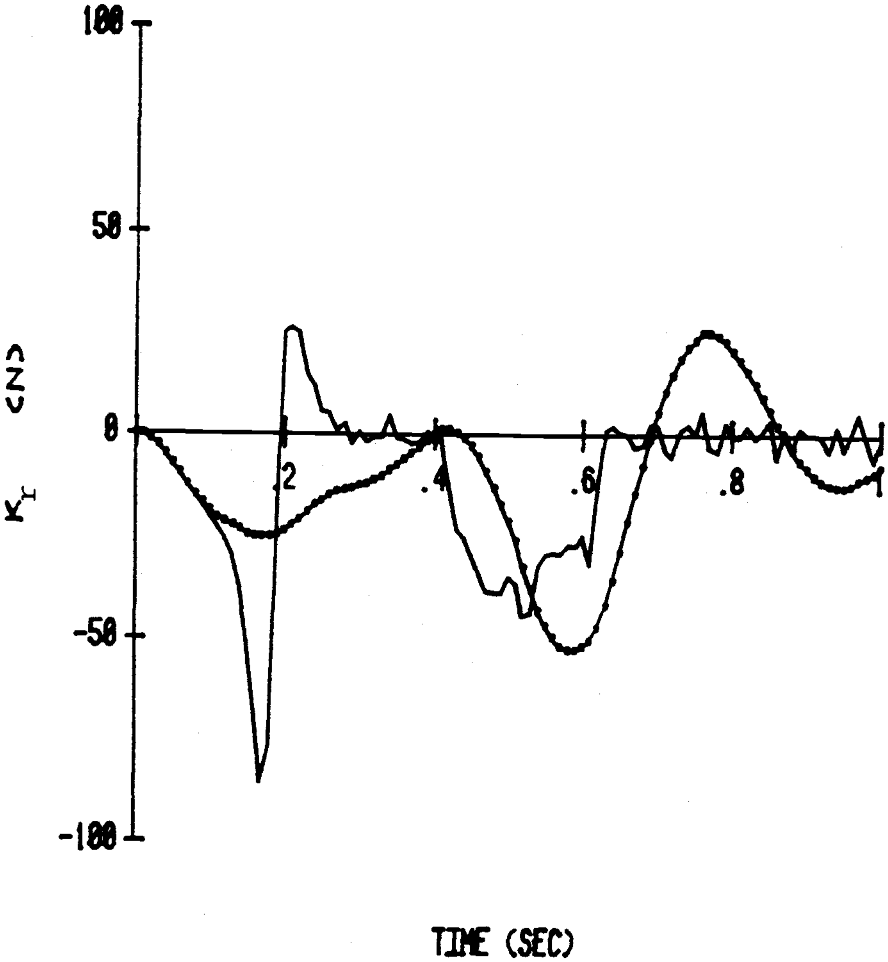

Figure 5.3.32 Input Force K_r (case 2)

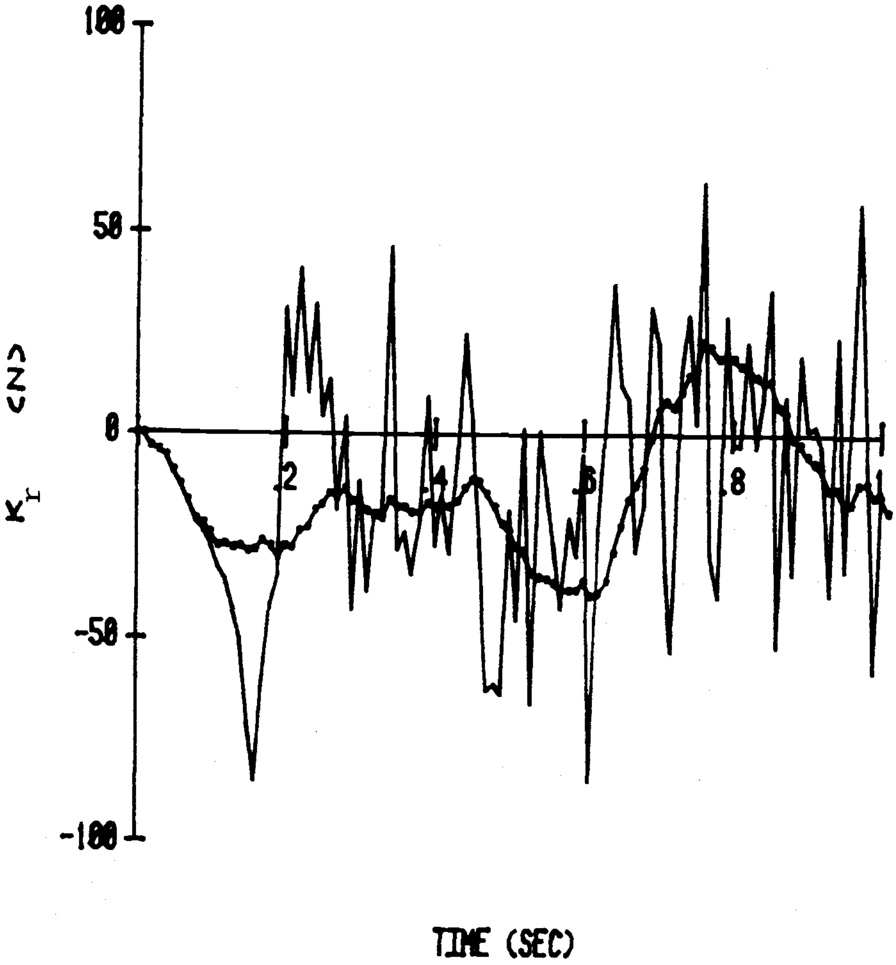

Figure 5.3.33 Input Force K_r (case 3)

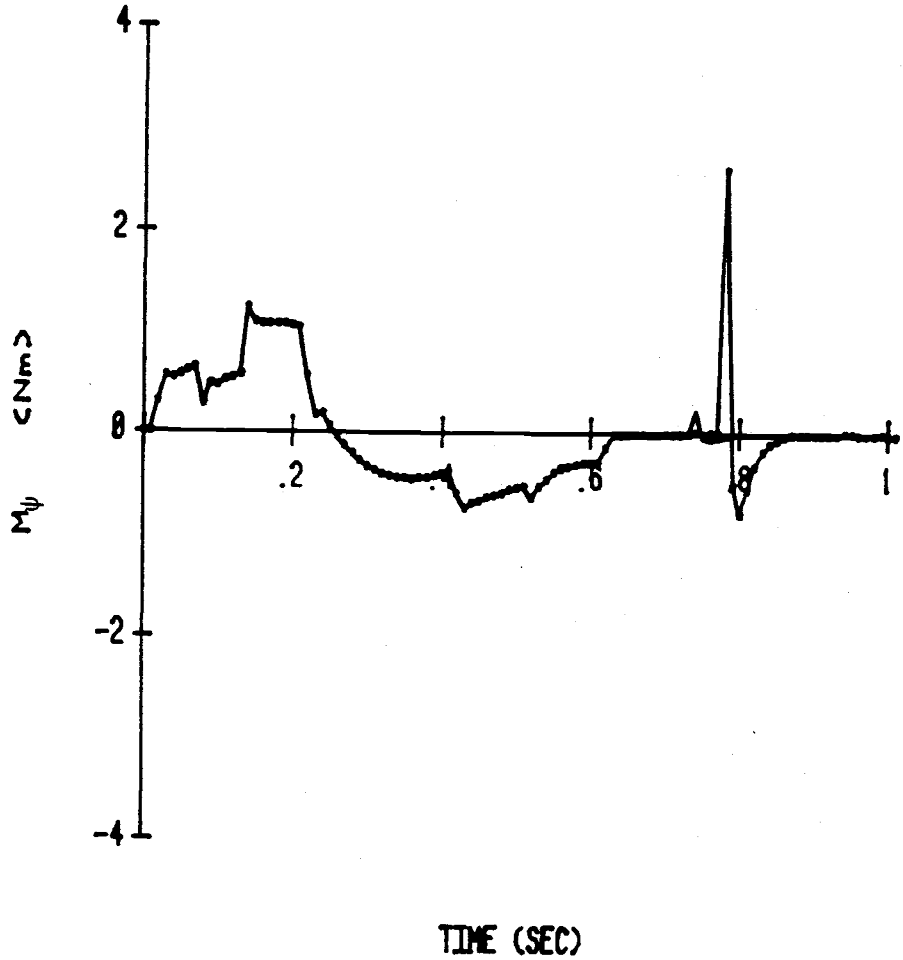

Figure 5.3.34 Input Torque M_ψ (case 1)

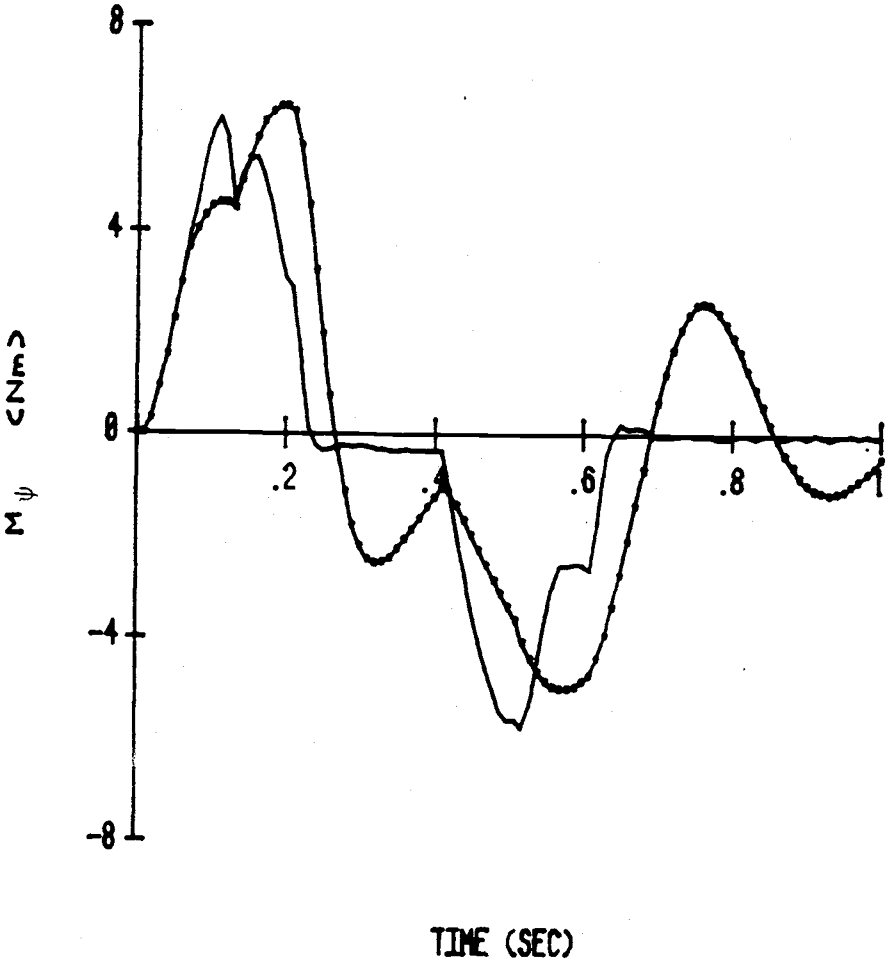

Figure 5.3.35 Input Torque M_ψ (case 2)

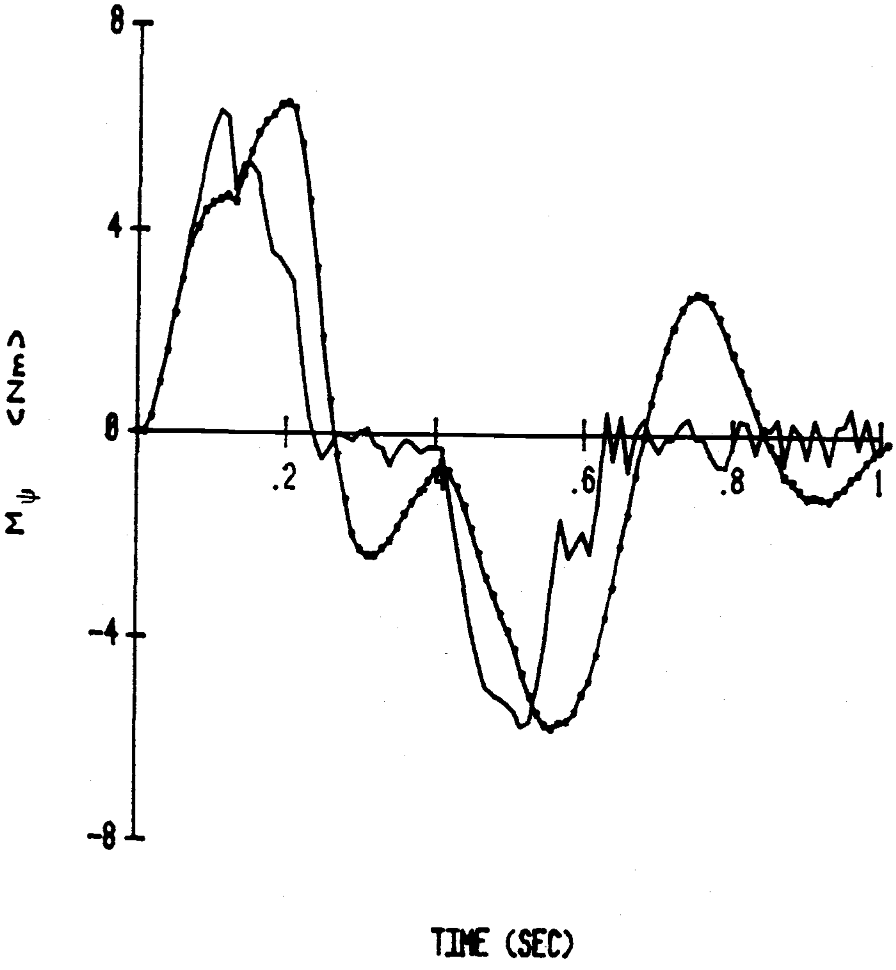

Figure 5.3.36 Input Torque M_ψ (case 3)

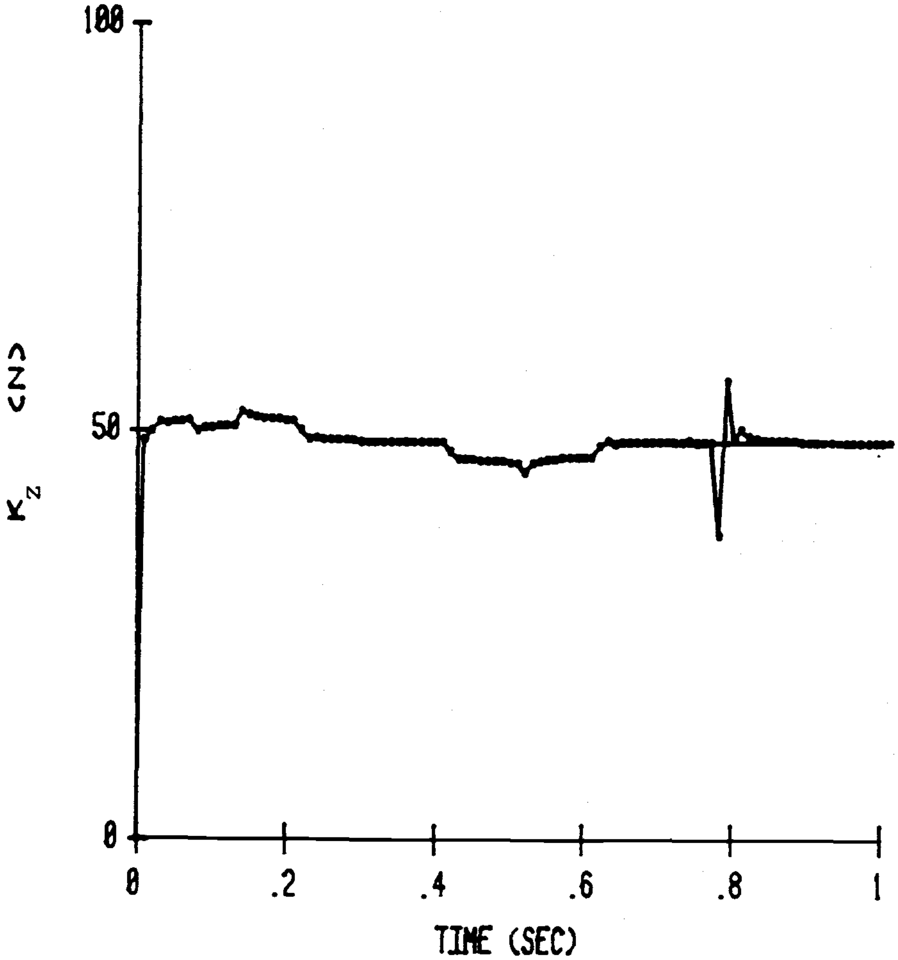

Figure 5.3.37 Input Force K_z (case 1)

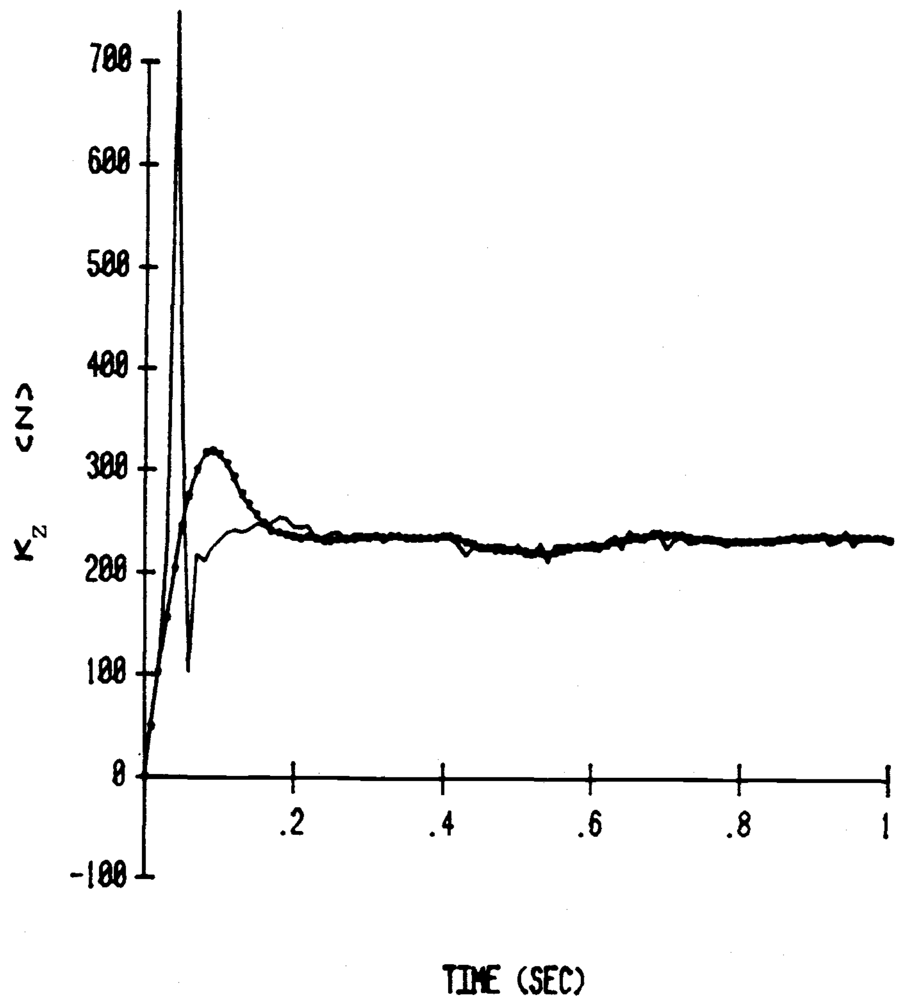

Figure 5.3.38 Input Force K_z (case 2)

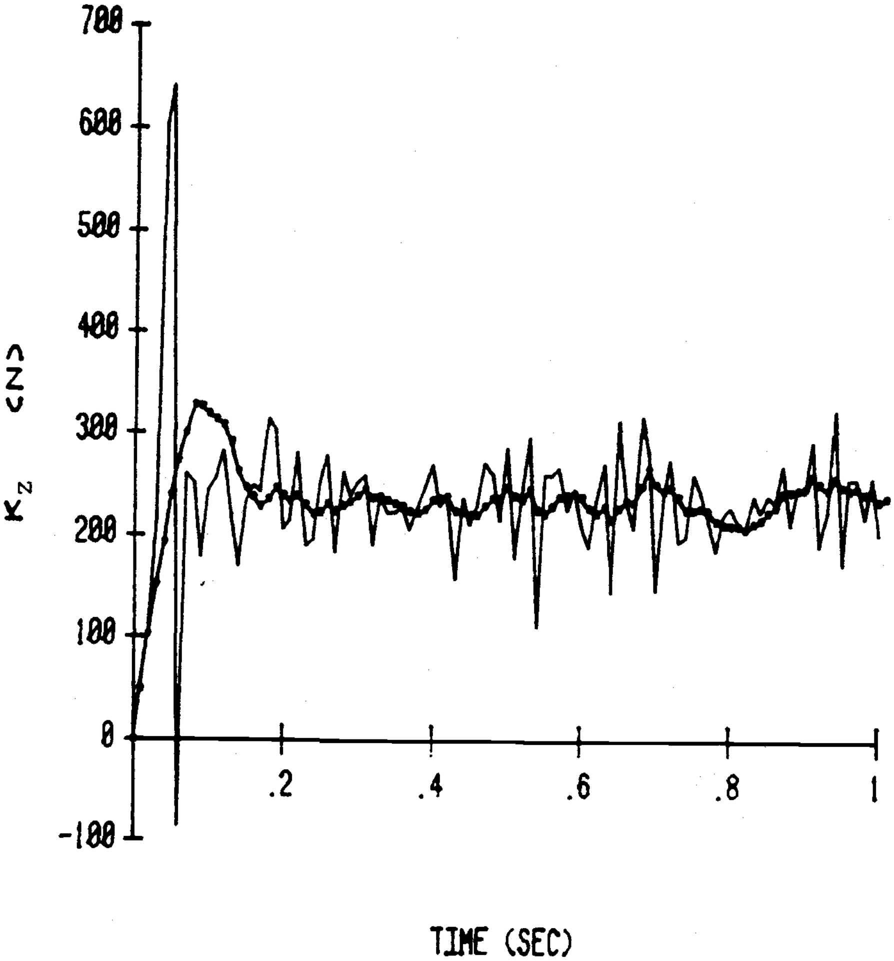

Figure 5.3.39 Input Force K_z (case 3)

VI. CONCLUSIONS

Discrete-time model reference adaptive control (MRAC) for single-input single-output (SISO) system has been reviewed and a new approach using this adaptive control concept for robotic manipulators which are multi-input multi-output (MIMO) systems, has been presented in this work.

The simulation results of case 1, show that using an approximate discrete-time model of the robot system is justified since the discrete-time linear control law gives almost perfect model following responses for the continuous-time nonlinear robot model.

The simulations of case 2 and case 3 show the effects of payload uncertainty and random process noise on the MRAC scheme. Eventhough a 20 kg payload, which is extremely heavy compared to 4 kg of estimated robot weight (excluding the base portion) was used, the robotic manipulator was well controlled by the adaptive control scheme while the conventional linear control scheme provided poor performance. These results show that the MRAC scheme can be operated over wide range of payloads. The range of payloads is usually very restrictive in most high performance industrial robots. In this work the power limitations of each actuator was not considered. However, the forces and torques which were generated in the simulations for the three cases, are quite reasonable in size except for few time steps.

It can also be seen in the results of case 2 and case 3 that the MRAC law provides acceptable control signals to obtain the high performance of control system with respect to stochastic process noise of the form introduced. It shows that MRAC systems can have robust properties with respect to stochastic process noise.

Besides the results presented here, this MRAC algorithm, which was determined for a rigid robot model was implemented for the flexible model of the robot system. It was found in this case that the MRAC system becomes unstable. The instability is presumed to be caused by unmodelled flexible dynamics in control scheme, which is well known problem in adaptive control techniques. Therefore it should be mentioned that more study is needed to apply adaptive control schemes in practical cases, since it is almost impossible to obtain a perfectly rigid model in real systems. The effect of unmodelled flexible dynamics in discrete-time MRAC of a flexible link was discussed in ref. [53].

As mentioned above, further research of adaptive techniques should be directed toward studying the robustness properties of control systems in the presence of unmodelled dynamics.

MRAC techniques applied to flexible robot systems are suggested as further research on this subject.

REFERENCES

1. Lecture Notes in Control and information Sciences, V. 24, Methods and Applications in Adaptive Control, Proc. of International Symposium, Bochum,1980, edited by H. Unbehauen, Spring-Verlag.

2. Astrom, K.J., "Why use Adaptive Techniques for Steering Large Tankers?," Int. J. Control, V. 32, No. 4, 1980, pp. 689-708.

3. Dahhou, B., Nijim, K., M'Saad, M., and Youlal, B., "Model Reference Adaptive Control of An Industrial Phosphate Drying Furnace," IFAC Adaptive Systems in Control and Signal Processing, San Francisco, USA, 1983, pp. 315-321.

4. Kallstrom, C.G., Astrom, K.J., Thorell, N.E., Eriksson, J., and Sten, L., "Adaptive Autopilots for Tankers," Automatica, V. 15, 1979, pp. 241-254.

5. Andreiev, N., "A Process Controller that Adapts to Signal and Process Conditions," Control Engng., Dec. 1977, pp. 38-40.

6. Park, P.C., "Lyapunov Redesign of Model Reference Adaptive Control Systems," IEEE Trans. Auto. Control, AC-11, 1966, pp. 362-367.

7. Monopoli, R.V., "Model Reference Adaptive Control with An Augmented Error Signal," IEEE Trans. Auto. Control, AC-19, Oct. 1974, pp. 474-484.

8. Landau, I.D., "A Survey of Model-reference Adaptive Techniques: Theory and Applications," Automatica, V. 10, July 1974, pp. 353-379.

9. Landau, I.D., Adaptive Control - The Model Reference Approach, Marcel Dekker, New York, 1979.

10. Kalman, R.E., "Design of A Self-optimizing Control System," Trans. ASME v.80, 1958, pp. 468-478.

11. Astrom, K.J., "Self-tuning Regulators - Design Priciples and Applications," In Applications of Adaptive Control, Academic Press, New York, 1980.

12. Whitaker, H.P., Yamron, J., and Kezer, A., "Design of Model-reference Adaptive Control Systems for Aircraft," Report R-164, Instrumentation Lab., MIT, Cambridge, 1958.

13. Park, P.C., 1966, op. cit.

14. Lindorff, D.P. and Carroll, R.L., "Survey of adaptive Control using Liapunov Design," Int. J. Control, 1973, V. 18, No. 5, pp. 897-914.

15. Luders, G., and Narendra, K.S., "Lyapunov Functions for Quadratic Differential Equations with Applications to Adaptive Control," IEEE Trans. Auto. Control, V. AC-17, Dec. 1972, pp. 798-801.

16. Lin, Y., and Narendra, K.S., "A New Error Model for Adaptive Systems," IEEE Trans. Auto. Control, V. AC-25, No. 3, June 1980, pp. 585-587.

17. Ohkawa, F., "A Model Reference Adaptive Control System for a Class of Discrete Linear Time-varing Systems with Time Delay," Int. J. Control, 1985, V. 42, No 5, pp. 1227-1238.

18. Popov, V.M., Hyperstability of Control Systems, Springer-Verlag, 1973.

19. Landau, I.D., and Silveira, H.M., "A Stability Theorem with Applications to Adaptive Control," IEEE Trans. Auto. Control, V. AC-24, No. 2, April 1979, pp. 305-312.

20. Anderson, B.D.O., Bitmeat, R.R., Johnson, C.R., and Kosut, R.L., "Stability Theorems for the Relaxation of the Strictly Positive Real Condition in Hyperstable Adaptive Scheme," IEEE Proc. 23rd Conf. on Decision and Control, Las Vegas, 1984, pp. 1286-1291.

21. Balestino, A., De Maria, G., and Sciavicco, L., "Hyperstable Adaptive Model Following Control of Nonlinear Plants," Systems and Control Letters, V. 1, No. 4, Jan. 1982, pp. 232-236.

22. Landau, I.D., 1979, op. cit.

23. Dubowsky, S. and DesForges, D.T., "The Adpplication of Model Reference Adaptive Control to Robotic Manipulators," ASME J. Dynamic Systems, Meas. and Cont., V. 101, Sept. 1979, pp. 193-200.

24. Balestrino, A., De Maria, G., and Sciavicco, L., "An Adaptive Model Following Control for Robotic Manipulators," ASME J. Dynamic Systems, Meas. and Cont., V. 105, Sept. 1983, pp. 143-151.

25. Nicosia, S. and Tomei, P., "Model Reference Adaptive Control Algorithms for Industrial Robots," Automatica, V. 20, No. 5, Sept. 1984, pp. 635-644.

26. Tomizuka, M. and Horowitz, R., "Model Reference Adaptive Control of Mechanical Manipulators," IFAC Adaptive Systems in Control and Signal Proc., San Francisco, 1983, pp. 27-32.

27. J. Yuh, and W.E. Holley, "Discrete-time Model Reference Adaptive Control for Robotic Manipulators," Proc. 19th Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove CA, Nov. 1985.

28. Tomizuka, M. and Horowitz, R., 1983, op. cit.

29. Landau, I.D., Lecture Notes on Adaptive Control, University of California, Berkeley Extension, 1984.

30. Popov, V.M., 1973, op. cit.

31. Freeman, H., Discrete-Time Systems, John Wiley and Sons, Inc. New York, 1965.

32. Popov, V.M., 1973, op. cit.

33. Anderson, B.D.O., "A System Theory Criterion for Positive Real Matrices," J. SIAM Control, V. 5, No. 2, 1967, pp. 171-182.

34. Hitz, l., and Anderson, B.D.O., "Discrete Positive-real Functions and Their Application to System Stability," Pro. IEE, V. 116, No. 1, Jan. 1969.

35. Landau, I.D., 1979, op. cit.

36. Popov, V.M., 1973, op. cit.

37. Anderson, B.D.O., "A Simplified Viewpoint of Hyperstability," IEEE Trans. Auto. Control, June 1968, pp. 292-294.

38. Ionescu, T., "Hyperstability of Linear Time-varying Discrete Systems," IEEE Trans. Auto. Control, Dec. 1970, pp. 645-647.

39. Landau, I.D., "Hyperstability Concepts and Their Application to Discrete Control Systems," Proc. 1972 JACC, Stanford, CA, Aug. 1972.

40. Landau, I.D., "An Extension of a Stability Theorem Applicable to Adaptive Control," IEEE Trans. Auto. Control, V. AC-25, No. 4, Aug. 1980.

41. Landau, I.D., 1984, op. cit.

42. Ionescu, I. and Monopoli, R.V., "Discrete Model Reference Adaptive Control with an augmented error signal," Automatica, V. 13, 1977, pp. 507.

43. Landau, I.D., "Unbiased Recursive Identification using Model Reference Adaptive Techniques," IEEE Trans. Aut. Control, AC-21, 1976, pp. 194-202.

44. Landau, I.D., and Lozano, R., "Unification of Discrete Time Explicit Model Reference Adaptive Control Designs," Automatica, V. 17, No. 4, 1981, pp. 593-611.

45. Lozano, R., "Adaptive Control of Non-Minimum Phase Systems with Independent Tracking and Regulation Specifications," IFAC Software for Computer Control, Madrid, Spain, 1982.

46. Dugard, L. and Landau, I.D., "Stochastic Model Reference Adaptive Controllers," presented at the 19th IEEE CDC, Albuquerque, USA, 1980, pp. 1132-1137.

47. Landau, I.D. and Lozano, R., 1981, op. cit.

48. Industrial Robot Model-700 Manual, Seiko Instruments Inc., Torrance, CA, 1985.

49. Arnold, V.I., Mathmatical Methods of Classical Mechanics, Springer-Verlag, 1978.

50. Landau, I.D., and Lozano, R., 1981, op. cit.

51. Whitney, C.A., "Generating and Testing Pseudo-Random Numbers," Byte, Oct. 1984, pp. 128-464.

52. J. Yuh, and W.E. Holley, "An Algorithm for Solving Stiff Non-linear Dynamic Equations Based on the Secant Method," to be published.

53. J. Yuh, and W.E. Holley, "Effect of Unmodelled Flexible dynamics in Discrete-time Model Reference Adaptive Control of Flexible Link," to be published.

54. Anderson, B.D.O., "A System Theory Criterion for Positive Real Matrices," J. SIAM Control, V. 5, No. 2, 1967, pp.171-182.

55. Arnold, V.I., 1978, op. cit.

APPENDICES

APPENDIX A

PROOF OF THEOREM 2.3.1

We will first prove the equivalency between propositions 2 and 4 of the theorem. From the continuous version of the positive real lemma [54], it is known that the square transfer matrix $H(s) = D + C(sI-A)^{-1}B$ is positive real if and only if, there exist a real symmetric positive definite matrix P and real matrices L and K such that

$$\begin{aligned} PA + A^TP &= -LL^T \\ B^TP + K^TL^T &= C \\ K^TK &= D + D^T \end{aligned} \tag{A.1}$$

for the continuous-time system:

$$\begin{aligned} x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{A.2}$$

Using this result,

Necessity: we consider first the case where H(z) of equation (2.2.3) is analytic at z=-1. By means of the bilinear transformation:

$$s = \frac{z - 1}{z + 1} \tag{A.3}$$

the matrix H(z) is transformed into a matrix

$$H_c(s) = D_c + C_c(sI - A_c)^{-1}B_c \tag{A.4}$$

where

$$\begin{aligned} A_c &= (A + I)^{-1}(A - I) \\ B_c &= 2(A+I)^{-1}B \\ C_c &= C \\ D_c &= D - C(A + I)^{-1}B \end{aligned} \tag{A.5}$$

From the continuous version of positive real lemma, there exist real matrices $P_c = P_c^T > 0$, L_c and W_c such that

$$\begin{aligned} &P_c(A + I)^{-1}(A - I) + (A^T - I)(A^T + I)^{-1}P_c = -L_cL_c^T \\ &2B^T(A^T + I)^{-2}P_c + K_cL_c^T = C \\ &K_c^TK_c = D + D^T - C(A + I)^{-1}B - B^T(A^T + I)^{-1}C \end{aligned} \tag{A.6}$$

with the definitions

$$\begin{aligned} P &= 2(A^T + I)^{-1}P_c(A + I)^{-1} \\ L &= L_c \\ K &= K_c + L_c^T(A + I)^{-1}B \end{aligned} \tag{A.7}$$

equation (A.6) immediately reduce to equations (2.2.11) to (2.2.13).

The general case, where $H(z)$ has a simple pole at $z = -1$, can be treated by an expansion of $H(z)$ which separates out this pole. Thus

$$H(z) = H_1(z) + H_2(z) \tag{A.8}$$

where

$$H_1(z) = \frac{z-1}{z+1} M = M - \frac{2M}{z+1} \quad , \text{ M: a constant matrix} \tag{A.9}$$

and where $H_2(z)$ has no pole at $z = -1$.

Application of definition (2.2.1) then shows that M is real and nonnegative definite symmetric and that H(z) is discrete positive real if and only if both $H_1(z)$ and $H_2(z)$ are positive real. These two matrices will each have some minimal realization (A_1, B_1, C_1, D_1) and (A_2, B_2, C_2, D_2) in terms of which a minimal realization for H(z) is given by:

$$H(z) = D_1 + D_2 + [C_1 C_2] \left[zI - \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \right]^{-1} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tag{A.10}$$

By hypothesis, $H_2(z)$ is discrete positive real, and hence, from the previous arguments, there exist matrices $P_2=P_2^T>0$, L_2 and W_2 which satisfy equations (2.2.11) to (2.2.13) for the matrices A_2, B_2, C_2 and D_2. Since the matrix M in equation (A.9) is nonnegative definite symmetric, there exists a nonsingular T such that

$$M = T^{-1} \left(\sum_{i=1}^{r} x_i^T x_i \right) (T^T) = \sum_{i=1}^{r} y_i y_i \tag{A.11}$$

where r is the rank of M and the x_i are linearly independent real vectors. Therefore,

$$
\begin{aligned}
A_1 &= -I_r \\
B_1 &= [y_1\ y_2\ \cdots\ y_r]^T \\
C_1 &= -2[y_1\ y_2\ \cdots\ y_r]^T \\
D_1 &= \sum_{i=1}^{r} y_i y_i
\end{aligned}
\qquad (A.12)
$$

for which the matrices $P_1 = 2I_r$, $L_1 = 0$, $W_1 = 0$ are readily seen to satisfy equations (2.2.11) to (2.2.13).

It is now easily checked that the matrices

$$
\begin{aligned}
P &= \begin{bmatrix} 2I_r & 0 \\ 0 & P_2 \end{bmatrix} \\
L &= \begin{bmatrix} 0 \\ L_2 \end{bmatrix} \\
W &= W_2
\end{aligned}
\qquad (A.13)
$$

satisfies equations (2.2.11) to (2.2.13) for the square transfer matrix $H(z)$ of equation (2.2.3).

Sufficiency: It suffices to show that equations (2.2.11) to (2.2.13) implies equation (2.2.4) of definition 2.2.1.

From equation (2.2.11), it is readily verified that

$$
\begin{aligned}
&(z^*I - A^T)P(zI - A) + (z^*I - A^T)PA + A^TP(zI - A) \\
&= (|z|^2 - 1)P + LL^T
\end{aligned}
\qquad (A.14)
$$

After some manipulation, the use of equations (2.2.12) and (2.2.13), there results

$$D + D^T + C(zI - A)^{-1}B + B^T(z^*I - A\ )^{-1}C$$
$$= (|z|^2 - 1)B^T(z^*I - A^T)^{-1}P(zI - A)^{-1}B$$
$$+ \{K^T + B^T(z^*I - A)^{-1}L\}\{K + L^T(zI - A)^{-1}B\} \quad (A.15)$$

The right-hand side is clearly nonnegative definite in $|z|>1$ while the left-hand side is precisely $H(z) + H^T(z^*)$. This completes the proof of the equivalency between propositions 4. and 2. of theorem 2.3.1.

Now we show the equivalency propositions 3. and 4., and propositions 3. and 5.

Since the matrix of (2.2.10) must be at least positive semidefinite, it can be factored as:

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = NN^T$$
$$= \begin{bmatrix} L \\ K^T \end{bmatrix} \begin{bmatrix} L^T & K \end{bmatrix}$$
$$= \begin{bmatrix} LL^T & LK \\ K^TL^T & K^TK \end{bmatrix} \quad (A.16)$$

where L is an nxq - dimensional matrix and K^T is an m x q - dimensional matrix, and q is arbitrary.

Replacing Q by LL^T, S by K^TL^T, and R^T by K^TK in equations (2.2.7) to (2.2.9), we obtain equations (2.2.11) to

(2.2.13). Thus, proposition 3. is equivalent to proposition 4.

We replace the term $x(k)^T Qx(k)$ in equation (2.2.14) by an equivalent term obtained from equations (2.2.1) and (2.2.7).

Thus,

$$\begin{aligned} x(k)^T Qx(k) &= -x(k)^T A^T PAx(k) + x(k)^T Px(k) \\ &= -[x(k+1)^T - u(k)^T B^T]P[x(k+1) - Bu(k)] \\ &\quad + x(k)^T Px(k) \\ &= -x(k+1)^T Px(k+1) + x(k+1)^T PBu(k) \\ &\quad + u(k)^T B^T Px(k+1) - u(k)^T B^T PBu(k) + x(k)^T Px(k) \end{aligned} \tag{A.17}$$

Adding $2u(k)^T S^T x(k) + u(k)^T Ru(k)$ to the both sides of equation (A.17), and using equations (2.2.8), (2.2.9), and (2.2.2),

$$\begin{aligned} & x(k)^T Qx(k) + 2u(k)^T S^T x(k) + u(k)^T Ru(k) \\ & = -x(k+1)^T Px(k+1) + x(k)^T Px(k) + 2y(k)^T u(k) \end{aligned} \tag{A.18}$$

Thus,

$$\sum_{k=0}^{k_1} y(k)^T u(k) = -\frac{1}{2} \sum_{k=0}^{k_1} x(k+1)\ Px(k+1) - -\frac{1}{2} \sum_{k=0}^{k_1} x(k)^T Px(k)$$

$$+ \frac{1}{2} \sum_{k=0}^{k_1} [x(k)^T Qx(k) + 2u(k) S^T x(k) + u(k)^T Ru(k)]$$

(A.19)

Using equation (2.2.14) (A.19) becomes

$$\sum_{k=0}^{k_1} y(k)^T u(k) = \frac{1}{2} x(k_1 + 1)^T Px(k_1 + 1) - \frac{1}{2} x(0)^T Px(0) + \frac{1}{2} \sum_{k=0}^{k_1} [x(k)^T Qx(k) + 2u(k)^T S^T x(k) + u(k)^T Ru(k)]$$

(A.20)

Thus, proposition 5. is equivalent to proposition 3.

APPENDIX B

DYNAMICAL MODEL OF NATURAL SYSTEMS

Let $q = [q_1 \ldots\ldots q_n]$ be local coordinates and $\dot{q}$ a tangent vector (i.e. velocity vector) in this coordinate system.

As defined in Arnold [55], the Lagrangian function for a natural system is given by the difference between kinetic energy T and potential energy V:

$$L(q,\dot{q}) = T(q,\dot{q}) - V(q) \qquad \text{(B.1)}$$

where

$$T(q,\dot{q}) = \frac{1}{2}\,\dot{q}^T M(q)\dot{q} \quad , \quad M > 0 \qquad \text{(B.2)}$$

Thus, using the Lagrangian approach,

$$Q(t) = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} \qquad \text{(B.3)}$$

where Q is the n-dimensional vector of the generalized forces, dynamical model can be derived in the form:

$$M(q)\ddot{q} = -c(q,\dot{q}) + r(q) + Q(t) \qquad \text{(B.4)}$$

where $c(q,\dot{q})$ is the n-dimensional vector:

$$c(q,\dot{q}) = C(q)\begin{bmatrix} \dot{q}_1 & \dot{q}_2 \\ \dot{q}_1 & \dot{q}_2 \\ & \vdots & \\ \dot{q}_{n-1} & \dot{q}_n \\ \dot{q}_n & \dot{q}_n \end{bmatrix} \tag{B.5}$$

r(q) is the n-dimensional vector:

$$r(q) = -\frac{\partial V(q)}{\partial q} \tag{B.6}$$

Choosing the state vector x, $x \in R^{2n}$ in the form:

$$x^T = [x_1 \; .... \; x_{2n}] \tag{B.7}$$

where $x_i = q_i$ and $x_{i+n} = q_i$, $i=1,...,n$

and the input vector u, $u \in R^n$:

$$u = Q \tag{B.8}$$

the equation (B.4) can be written in the state space model:

$$\dot{x}(t) = A(x,t)x(t) + B(x,t)u(t) + W(x,t) \tag{B.9}$$

where

$$A(x,t) = \left[\begin{array}{c|c} 0 & I_n \\ \hline A_1(x,t) & A_2(x,t) \end{array}\right] \tag{B.10}$$

$$B(x,t) = \left[\begin{array}{c} 0 \\ \hline B_1(x,t) \end{array}\right] \tag{B.11}$$

$$W(x,t) = \begin{bmatrix} 0 \\ \hline W_1(x,t) \end{bmatrix} \tag{B.12}$$

In this equation, A_1, A_2 and B_1 are the nxn matrices, W_1 is the n-dimensional vector, and I_n represents the n x n identity matrix.

As a particular case of (B.9), using the notation:

$$L^T(x_{n+i}) = \left[\begin{array}{c|c} x_{n+i}I_n & \begin{array}{c} 0^T \\ \hline \begin{array}{c|c} x_{n+2}I_{n-2} & \begin{array}{c} 0^T \\ \hline \ddots \\ \hline x_{2n} \end{array} \end{array} \end{array} \end{array}\right] \tag{B.13}$$

the matrices B_1, A_1, A_2, and W_1 can be defined by:

$$B_1 = M^{-1}(x_i) \tag{B.14}$$

$$A_1 = 0 \tag{B.15}$$

$$A_2 = -B_1(x_i)C(x_i)L(x_{n+i}) \tag{B.16}$$

$$W_1 = B\ (x_i)r(x_i) \tag{B.17}$$

APPENDIX C

COMPUTER SIMULATION PROGRAM

C-1 Simulation model equations

The following equations were implemented in the simulation model:

Plant:
$$M_1\ddot{r}_p + M_2\dot{\psi}_p = K_r + w_{s1}^1$$
$$M_3\ddot{\psi}_p + M_4\dot{\psi}_p = M_\psi + w_{s1}^2 \qquad \text{(C.1)}$$
$$M_5\ddot{z}_p + M_6 = K_z + w_{s1}^3$$

where
$$M_1 = m_R + m_L$$
$$M_2 = -\frac{1}{2}\dot{\psi}^2[2m_R(r_p - \frac{\ell_1}{2}) + 2m_L r_p]$$
$$M_3 = \frac{1}{12}[(m_R\ell_1^2 + m_3\,\ell_2^2) + 3\{m_R(a_R^2 + b_R{}^2) + m_s(a_s^2 + b_s^2)\}] + m_R(r_p - \frac{\ell_1}{2}) + m_s(D - \frac{\ell_2}{2}) + m_L r^2 + m_T D^2 + \frac{1}{2}m_c(a_c^2 + b_c^2)$$
$$M_4 = 2r\,[m_R(r_p - \frac{\ell_1}{2}) + m_L r_p]$$
$$M_5 = m_R + m_s + m_c + m_T + m_L$$
$$M_6 = M_5 g \qquad \text{(C.2)}$$

In equations (C.2), m_s=1.26kg, m_R=0.37kg, m_c=1.97kg, m_T=0.4kg, m_L = payload, a_s=0.0125m, a_R=0.0075m, a_c=0.02m,

b_s=0.019m, b_R=0.011m, b_c=0.03m, ℓ_1=0.23m, ℓ_2=0.25m, D=0.17m, and g=9.8m/sec^2.

Reference model:

$$r_p(k+1) = r_p(k) + 0.01\dot{r}_p(k)$$
$$\psi_p(k+1) = \psi_p(k) + 0.01\dot{\psi}_p(k)$$
$$z_p(k+1) = z_p(k) + 0.01\dot{z}_p(k)$$
$$\dot{r}_p(k+1) = -188.01r_p(k) - 0.98\dot{r}_p(k) + 188.01u_r(k)$$
$$\dot{\psi}_p(k+1) = -188.01\psi_p(k) - 0.98\dot{\psi}_p(k) + 188.01u_\psi(k)$$
$$\dot{z}_p(k=1) = =188.01z_p(k) - 0.98\dot{z}_p(k) + 188.01u_z(k) \qquad (C.3)$$

Linear model following control (LMFC) law:

$$K_r(k) = [\dot{r}_M(k+1) + 33.3r_M(k) + 0.17\dot{r}_M(k) - 33.3r_p(k) - 0.17\dot{r}_p(k) - ESTY_1\psi_p(k) + ESTV_1d^*]/ESTU_1$$
$$M_\psi(k) = [\dot{\psi}_M(k+1) + 33.3\psi_M(k) + 0.17\dot{\psi}_M(k) - 33.3\psi_p(k) - ESTY_2\dot{\psi}_p(k) + ESTV_2d^*]/ESTU_2$$
$$K_z(k) = [\dot{z}_M(k+1) + 33.3z_M(k) + 0.17\dot{z}_M(k) - 33.3z_p(k) - 0.17\dot{z}_p(k) + ESTV_3d^*]/ESTU_3 \qquad (C.4)$$

where
$$ESTU_1 = 100M_1$$
$$ESTU_2 = 100M3$$
$$ESTU_3 = 100M_5$$

$$ESTY_1 = \dot{\psi}_p[2m_R(r_p - \frac{\ell 1}{2}) + 2m_Lr_p]/200M_1$$

$$ESTY_2 = -2\dot{r}_p[m_R(r_p - \frac{\ell 1}{2}) + m_L r_p]/100M_3 + 1.17 \quad (C.5)$$

Adaptation diagonal gain matrix F(k):

$$\text{daig } F(k) = [f_i(k)], \; i=1 \ldots 10$$

At each time step, the elements of adaptation gain matrix are computed by

$$f_1(k+1) = [f_1(k) - \{f_1(k)K_r(k)\}^2/(\delta + sum)]/\lambda_1(k)$$
$$f_2(k+1) = [f_2(k) - \{f_2(k)M_\psi(k)\}^2/(\delta + sum)]/\lambda_1(k)$$
$$f_3(k+1) = [f_3(k) - \{f_3(k)K_z(k)\}^2/(\delta + sum)]/\lambda_1(k)$$
$$f_8(k+1) = [f_8(k) - \{f_p(k)\dot{\psi}_p(k)\}^2/(\delta + sum)]/\lambda_1(k)$$
$$f_{10}(k+1) = [f_{10}(k) - \{f_{10}(k)d^*\}^2/(\delta + sum)]/\lambda_1(k)$$
$$f_4 = f_5 = f_6 = f_7 = f_9 = 0 \quad (C.6)$$

where $\delta = 0.5$

$$sum = f_1(k)K_r(k)^2 + f_2(k)M_\psi(k)^2 + f_3(k)K_z(k) + f_8(k)\psi_p(k)^2 + f_{10}(k)d^{*2}$$

and $\lambda_1(k)$ is computed such that trace[F(k)]=trace[F(0)].

A priori adaptation error vector $e_o^f = [e_{10}^f, e_{20}^f, e_{30}^f, e_{40}^f, e_{50}^f, e_{60}^f]^T$:

$$e_{10}^f = [r_p(k+1) - r_M(k+1)] - [r_p(k) - r_M(k)] - [\dot{r}_p(k) - \dot{r}_M(k)]/100$$

$$e_{20}^f = [\psi_p(k+1) - \psi_M(k+1)] - [\psi_p(k) - \psi_M(k)] - [\dot{\psi}_p(k) - \dot{\psi}_M(k)]/100$$

$$e_{30}^{f} = [z_p(k+1) - z_M(k+1)] - [z_p(k) - z_M(k)] - [\dot{z}_p(k) - \dot{z}_M(k)]/100$$

$$e_{40}^{f} = [\dot{r}_p(k+1) - \dot{r}_M(k+1)] + 33.3[r_p(k) - r_M(k)] + 0.17[\dot{r}_p(k) - \dot{r}_M(k)]$$

$$e_{50}^{f} = [\dot{\psi}_p(k+1) - \dot{\psi}_M(k+1)] + 33.3[\psi_p(k) - \psi_M(k)] + 0.17[\dot{\psi}_p(k) - \dot{\psi}_M(k)]$$

$$e_{60}^{f} = [\dot{z}_p(k+1) - \dot{z}_M(k+1)] + 33.3[z_p(k) - z_M(k)] + 0.17[\dot{z}_p(k) - \dot{z}_M(k)] \quad (C.7)$$

A posteriori adaptation error vector:

$$e^{f} = e_{o}^{f}/[1 + sum] \quad (C.8)$$

Parameter adaptation law:

$$ESTU_1(k+1) = ESTU_1(k) + f_1(k)K_r(k)e_4^f(k+1)$$

$$ESTU_2(k+1) = ESTU_2(k) + f_2(k)M_{\psi}(k)e_5^f(k+1)$$

$$ESTU_3(k+1) = ESTU_3(k) + f_3(k)K_z(k)e_6^f(k+1)$$

$$ESTY_1(k+1) = ESTY_1(k) + f_8(k)\psi_p(k)e_4^f(k+1)$$

$$ESTY_2(k+1) = ESTY_2(k) + f_8(k)\psi_p(k)e_5^f(k+1)$$

$$ESTV_1(k+1) = ESTV_1(k) + f_{10}(k)de_4^{*f}(k+1)$$

$$ESTV_2(k+1) = ESTV_2(k) + f_{10}(k)de_5^{*f}(k+1)$$

$$ESTV_3(k+1) = ESTV_3(k) + f_{10}(k)de_6^{*f}(k+1) \quad (C.9)$$

C-2 FORTRAN program

```
      PROGRAM ROBOT
      DOUBLE PRECISION DSEED
      DIMENSION E(10),X(10),Y(10),X00(10),Y00(10),X0(10),Y0(10),XE(10)
      DIMENSION ESTA(10,10),ESTB(10,10),ESTY(10,10),ESTU(10,10)
      DIMENSION CDIR(10),DY(10),ESTV(10),F(10),G(10)
      DIMENSION REGY(10),REGX(10),U(10),UM(10),U0(10),W(10)
      COMMON N,EE,COT,COA,COM,COK,M
      COMMON CR(10,10),SC(10)
      DSEED=100000.D0
      NR=3
      N=6
      M=3
      T=0.
      EE=0.00001
      ITMAX=50
   43 WRITE(3,700)
      READ(3,701) NSTOCH
      IF(NSTOCH.EQ.0) GO TO 810
         WRITE(3,811)
         READ(3,812) SR
  810 WRITE(3,35)
      READ(3,36) PLOAD
      WRITE(3,351)
      READ(3,361) DT
      WRITE(3,352)
      READ(3,362) Z
      DO 31 I=1,M
         WRITE(3,27) I
         READ(3,28) G(I)
   31 CONTINUE
      DO 311 I=1,N
         WRITE(3,27) I
         READ(3,28) F(I)
  311 CONTINUE
      WRITE(3,277)
      READ(3,288) FV
      WRITE(3,278)
      READ(3,289) SC(1),SC(2),SC(3),SC(4),SC(5),SC(6)
      WRITE(3,34)
      READ(3,23) TMAX
      WRITE(3,41)
      READ(3,42) CHANGE
      IF(CHANGE.EQ.1.) GO TO 43

      DATA OF SEIKO ROBOT(MASS,RADIUS,LENGTH)

      COLM=1.97
      ARMM=0.37
      SLEM=1.26
      ACTM=0.4
      CIR=2./100.
```

```
      AIR=0.75/100.
      SIR=1.25/100.
      COR=3./100.
      AOR=1.1/100.
      SOR=1.9/100.
      SLEN=25./100.
      ALEN=23./100.
      ACD=17./100.

      CALCULATION OF CONSTANT TERMS AND TOTAL MASS
      INCLUDING MOMENT OF INERTIA.

      COA=PLOAD+ARMM
      COT=ARMM*ALEN
      CYL1=(SLEM*SLEN**2+3.*SLEM*(SIR**2+SOR**2))/12.
     ++SLEM*(ACD-SLEN/2)**2
      CYL2=(ARMM*ALEN**2+3.*ARMM*(AIR**2+AOR**2))/12.
     ++ARMM*(ALEN**2)/4.
      CYL3=COLM*(CIR**2+COR**2)/2.
      CACT=ACTM*ACD**2
      COK=CYL1+CYL2+CYL3+CACT
      COM=COLM+SLEM+ACTM

      DEFINE THE INITIAL AND FINAL POINTS
      OF ROBOT MANIPULATOR.

      DO 24 I=1,N
         X0(I)=0.
         Y0(I)=0.
         XE(I)=0.
   24 CONTINUE
      X0(1)=0.1
      Y0(1)=0.1
      XE(2)=0.15
      XE(3)=0.04

      TRANSFORMATION OF RECTANGULAR COORD. TO CYLINDERICAL COORD.

      CALL TRANS(X,X0)
      CALL TRANS(Y,Y0)

      INITIALIZE THE UNKNOWN PARAMETERS OF ROBOT
      CONTROL SYSTEM AND DEFINE THE REGULATOR
      DYNAMICS.

      FCOA=ARMM+1.
      A1=(2.*FCOA*X(1)-COT)*X(5)/(2.*FCOA)
      H=COK+FCOA*X(1)**2-COT*X(1)
      A2=(COT-2.*FCOA*X(1))*X(4)/H
      B1=1./FCOA
      B2=1./H
      B3=1./(FCOA+COM)
      DO 110 I=1,N
         U(I)=0.
         ESTV(I)=0.
         W(I)=0.
         DO 110 J=1,N
            CR(I,J)=0.
            ESTA(I,J)=0.
            ESTB(I,J)=0.
            ESTY(I,J)=0.
            ESTU(I,J)=0.
  110 CONTINUE
```

```
      NN=N/2
      DO 111 I=1,NN
         CR(I,I)=-1.
         II=I+3
         CR(I,II)=-DT
         CR(II,I)=1./(3.*DT)
         CR(II,II)=1./6.
         ESTA(I,I)=-1.
         ESTA(I,II)=-DT
  111 CONTINUE
      ESTA(4,4)=-1.
      ESTA(4,5)=-A1*DT
      ESTA(5,5)=-1.-A2*DT
      ESTA(6,6)=-1.
      ESTB(4,1)=DT*B1
      ESTB(5,2)=DT*B2
      ESTB(6,3)=DT*B3
      ESTV(6)=DT*9.8
      KN=N/2+1
      DO 112 I=KN,N
      DO 112 J=1,N
         ESTY(I,J)=CR(I,J)-ESTA(I,J)
  112 CONTINUE
      DO 113 J=1,NN
         I=J+NN
         ESTU(I,J)=ESTB(I,J)
  113 CONTINUE

      PRINT THE FOLLOWING INPUT VALUES:
         PAYLOAD
         SAMPLING TIME INTERVAL
         EIGENVALUE OF REF. MODEL
         SELECTION OF ADAPTATION MECHANISM
         INITIAL POINTS OF REF. MODEL AND ROBOT
         GAINS OF ADAPTATION MECHANISM
         COMPUTE THE TRACE OF INITIAL ADAP. GAIN MATRIX, DTR

      IF(NSTOCH.EQ.0) GO TO 705
         WRITE(3,702)
         GO TO 706
  705    WRITE(3,703)
  706 WRITE(3,37) PLOAD
      WRITE(3,371) DT
      WRITE(3,372) Z
      DO 26 I=1,N
         WRITE(3,25) I,X(I),Y(I)
   26 CONTINUE
      DTR=0.
      DO 45 I=1,M
         WRITE(3,29) I,G(I)
         DTR=DTR+G(I)**2
   45 CONTINUE
      DO 455 I=1,N
         WRITE(3,29) I,F(I)
         DTR=DTR+F(I)**2
  455 CONTINUE
      WRITE(3,299) FV
      IF(DTR.EQ.0.) DTR=1
      DTR=SQRT(DTR)
      WRITE(3,32)

      THE FOLLOWING ITERATION IS IMPLEMENTED TO
      CONTROL THE ROBOT MANIPULATOR.
```

```
 40 DO 30 KK=1,N
       E(KK)=X(KK)-Y(KK)
 30 CONTINUE
    CALL INPT(UM,T,X0,XE,CDIR)
    CALL ERROR(CDIR,EPS,ELONG,ETRAS,UM,Y)
    WRITE(3,33) T,Y(1),Y(2),Y(3),E(1),E(2),E(3),EPS,ELONG,ETRAS
    IF(T.GE.TMAX) GO TO 50
    T=T+DT
    DO 114 I=1,N
       X00(I)=X(I)
114 CONTINUE
    CALL SOLVM(X,UM,T,Z,DT)
    CALL REGL(REGX,X,X00)
    DO 115 I=1,N
       Y00(I)=Y(I)
       U0(I)=U(I)
115 CONTINUE
    IF(NSTOCH.EQ.0) GO TO 707
       CALL RANDOM(DSEED,NR,W,SR)
707 CALL CTR(U,ESTU,ESTY,ESTV,Y00,X,X00)
    PDT=DT/5.
    PT=T-DT
900 PT=PT+PDT
    CALL SOLVP(Y,U,W,PT,K,ITMAX,PDT)
    IF(K.EQ.ITMAX) GO TO 777
       IF(PT.LT.T) GO TO 900
          CALL REGL(REGY,Y,Y00)
          CALL ADAPT(ESTY,ESTU,ESTV,REGX,REGY,U0,Y00,F,G,FV,DTR)
    GO TO 40
777 WRITE(3,477)
 50 STOP


 34 FORMAT(' FINAL TIME= ')
 23 FORMAT(F10.3)
 25 FORMAT(/,' INITIAL VALUES OF X(I),Y(I), I=',I2,';',2F10.4)
 27 FORMAT(' G(I) OR F(I) ;  I=',I2,'  :')
 28 FORMAT(F12.7)
 29 FORMAT(/,' G(I) OR F(I) ,  I=',I2,' ;',F12.7)
277 FORMAT(' FV ;')
288 FORMAT(F12.7)
299 FORMAT(/,' FV ;',F12.7)
278 FORMAT(' SCALING FACTOR OF FUNCTIONS: SC(N)=',/)
289 FORMAT(6F10.4)
 32 FORMAT(////,3X,'T',7X,'Y(1)',6X,'Y(2)',6X,'Y(3)',6X,'E(1)'
   *,6X,'E(2)',6X,'E(3)',6X,'EPS',6X,'ELONG',6X,'ETRAS')
 33 FORMAT(1X,F5.2,2X,9G10.3)
 35 FORMAT(' PAYLOAD=')
351 FORMAT(' TIME INTERVAL DT=')
352 FORMAT(' EIGENVALUE OF REF.MODEL ON D.T., Z=')
 36 FORMAT(F5.2)
361 FORMAT(F8.4)
362 FORMAT(F10.4)
371 FORMAT(/,' TIME INTERVAL =',F8.4,/)
372 FORMAT(/,' EIGENVALUE OF REF. MODEL ON D.T., Z=',F10.4)
 37 FORMAT(/,' PAYLOAD=',F5.2,/)
 41 FORMAT(' CHANGE INPUT ?  YES=1.,NO=0. ')
 42 FORMAT(F5.2)
477 FORMAT(/,' NO CONVERGENCE')
700 FORMAT(' DO YOU WANT RANDOM NOISE? Y=1,N=0')
701 FORMAT(I2)
```

```
702 FORMAT(/,' THIS SYSTEM HAS A PROCESS RANDOM NOISE')
703 FORMAT(/,' THIS SYSTEM HAS NO PROCESS RANDOM NOISE')
811 FORMAT(' SCALING FACTOR OF STOCH. NOISE VECTOR=')
812 FORMAT(F12.5)
    END



    SUBROUTINE INPT(X,T,X0,XE,CDIR)
    DIMENSION X(10),X0(10),XE(10),CDIR(10),XT(10)
    COMMON N,EE,COT,COA,COM,COK,M

    GENERATE TRAJECTORY INPUT USING TRPERZOIDAL VELOCITY LAW IN
    CYLINDERICAL COORDINATE.

    A0=2.30827316
    V0=0.46165463
    D=SQRT((XE(1)-X0(1))**2+(XE(2)-X0(2))**2+(XE(3)-X0(3))**2)
    DO 10 I=1,M
       CDIR(I)=(XE(I)-X0(I))/D
 10 CONTINUE
    T1=V0/A0
    T2=D/V0
    T3=T1+T2
    IF(T.LE.T3) S=D-(A0/2.)*(D/V0+V0/A0-T)**2
    IF(T.LE.T2) S=V0*T-(V0**2)/(2.*A0)
    IF(T.LE.T1) S=(A0/2.)*T**2
    DO 20 I=1,M
       XT(I)=X0(I)+CDIR(I)*S
 20 CONTINUE
    X(1)=SQRT(XT(1)**2+XT(2)**2)
    IF(XT(1).EQ.0.) XT(1)=0.1**10
    X(2)=ATAN(XT(2)/XT(1))
    X(3)=XT(3)
    RETURN
    END



    SUBROUTINE ADAPT(ESTY,ESTU,ESTV,REGX,REGY,U,Y,F,G,FV,DTR)
    DIMENSION ERRA(10),ERRB(10),ESTY(10,10),ESTU(10,10)
    DIMENSION REGY(10),REGX(10),U(10),Y(10)
    DIMENSION F(10),G(10),TF(10),TG(10)
    DIMENSION ESTV(10)
    COMMON N,EE,COT,COA,COM,COK,M
    COMMON CR(10,10),SC(10)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
 YUH,J. AND HOLLEY,W.E. MECH.ENGR., OREGON STATE UNIV 8/1/85
  ADJUST THE UNKNOWN PARAMETERS OF ROBOT CONTROL SYSTEM
  USING PARAMETER ADAPTATION ALGORITHM
  INPUT.....
     U= CONTROL SIGNAL FOR ROBOT SYSTEM
     Y= OUTPUT OF ROBOT SYSTEM
  OUTPUT.....
     ESTY= ADJUSTABLE CONTROLLER CORRESPONDING TO Y
     ESTU= ADJUSTABLE CONTROLLER CORRESPONDING TO U
     ESTV= ADJUSTABLE CONTROLLER CORRESPONDING TO NOISE
  PARAMETER.....
     F= ADAPTATION GAIN CORRESPONDING TO Y
```

```
      G= ADAPTATION GAIN CORRESPONDING TO U
      FV= ADAPTATION GAIN CORRESPONDING TO NOISE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      D=1
      SUM=0.
      TRACE=0.
      DELTA=0.5
      DO 10 I=1,M
         SUM=SUM+G(I)*U(I)**2
   10 CONTINUE
      DO 20 I=1,N
         SUM=SUM+F(I)*Y(I)**2
   20 CONTINUE
      DO 30 I=1,N
         ERRA(I)=REGY(I)-REGX(I)
         ERRB(I)=ERRA(I)/(1.+SUM+FV*D**2)
   30 CONTINUE
      ESTU(4,1)=ESTU(4,1)+G(1)*U(1)*ERRB(4)
      ESTU(5,2)=ESTU(5,2)+G(2)*U(2)*ERRB(5)
      ESTU(6,3)=ESTU(6,3)+G(3)*U(3)*ERRB(6)
      ESTY(4,5)=ESTY(4,5)+F(5)*Y(5)*ERRB(4)
      ESTY(5,5)=ESTY(5,5)+F(5)*Y(5)*ERRB(5)
      ESTV(4)=ESTV(4)+FV*D*ERRB(4)
      ESTV(5)=ESTV(5)+FV*D*ERRB(5)
      ESTV(6)=ESTV(6)+FV*D*ERRB(6)
      ALPHA=DELTA+SUM
      DO 40 I=1,M
         TG(I)=G(I)
         G(I)=G(I)-(U(I)**2)*(G(I)**2)/ALPHA
         TRACE=TRACE+G(I)**2
   40 CONTINUE
      DO 50 I=1,N
         TF(I)=F(I)
         F(I)=F(I)-(Y(I)**2)*(F(I)**2)/ALPHA
         TRACE=TRACE+F(I)**2
   50 CONTINUE
      IF(TRACE.EQ.0.) GO TO 61
         TRACE=SQRT(TRACE)
         DO 70 I=1,M
            G(I)=G(I)*DTR/TRACE
   70    CONTINUE
         DO 80 I=1,N
            F(I)=F(I)*DTR/TRACE
   80    CONTINUE
         GO TO 60
   61 DO 81 I=1,M
         G(I)=TG(I)
   81 CONTINUE
      DO 82 I=1,N
         F(I)=TF(I)
   82 CONTINUE
   60 RETURN
      END


      SUBROUTINE SOLVM(X,UM,T,Z,DT)
      DIMENSION X(10),UM(10),P(10)
      COMMON N,EE,COT,COA,COM,COK,M
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
 YUH,J. AND HOLLEY,W.E. MECH.ENGR., OREGON STATE UNIV 3/1/86
```

```
C     GENERATE THE OUTPUT OF DISCRETE-TIME REFERENCE MODEL
C     INPUT.....
C        UM= REFERENCE INPUT COMPUTED BY DESIRED TRAJECTORY
C        Z= EIGENVALUE OF REFERENCE MODEL
C     OUTPUT.....
C        X= OUTPUT OF REFERENCE MODEL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         A1=2.*(1.-Z)/DT
         A0=(Z**2+DT*A1-1)/(DT**2)
         DO 10 I=1,M
            II=I+M
            P(I)=X(I)+DT*X(II)
            P(II)=-DT*A0*X(I)+(1.-DT*A1)*X(II)+DT*A0*UM(I)
      10 CONTINUE
         DO 20 I=1,N
            X(I)=P(I)
      20 CONTINUE
         RETURN
         END




         SUBROUTINE SOLVP(X,UM,W,T,K,ITMAX,DT)
         DIMENSION X(10),DELX(10),XX(10)
         DIMENSION VEL(10),VV(10),TA(10),PT(10)
         DIMENSION P(10),D(10,10),X0(10),Q(10),AJ(10,10)
         DIMENSION PP(10,10),DELP(10),UM(10)
         DIMENSION Z(10),R(10,10),QQ(10,10),DELPP(10)
         DIMENSION W(10)
         COMMON N,EE,COT,COA,COM,COK,M
         COMMON CR(10,10),SC(10)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     YUH,J. AND HOLLEY,W.E. MECH.ENGR., OREGON STATE UNIV 9/1/84
C     INPUT....
C        UM= CONTROL SIGNAL TO ROBOT SYSTEM
C        W= RANDOM PROCESS NOISE
C     OUTPUT.....
C        X= OUTPUT OF ROBOT SYSTEM
C
C     SUBROUTINES REQD.- FUNCA,ORTHO,INVERS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
         TOL=0.00001
         K=0
         IF(T.GT.DT) GO TO 600
         DO 10 I=1,N
            VEL(I)=EE
            X0(I)=X(I)
            X(I)=X(I)+VEL(I)*DT
      10 CONTINUE
            CALL FUNCA(P,X,VEL,UM,W,T)
         DO 20 I=1,N
            XX(I)=X(I)
            VV(I)=VEL(I)
      20 CONTINUE
         DO 40 I=1,N
               XX(I)=X(I)+EE
               VV(I)=(XX(I)-X0(I))/DT
               CALL FUNCA(PT,XX,VV,UM,W,T)
            DO 30 J=1,N
               PP(J,I)=PT(J)
      30    CONTINUE
```

```
      XX(I)=X(I)
      VV(I)=VEL(I)
   40 CONTINUE
      DO 50 I=1,N
         DO 50 L=1,N
            AJ(I,L)=(PP(I,L)-P(I))/EE
   50 CONTINUE
      CALL INVERS(AJ,N)
  800 DO 60 L=1,N
         DELX(L)=0.
         DO 60 ME=1,N
            DELX(L)=DELX(L)-AJ(L,ME)*P(ME)
   60 CONTINUE
  102 DO 70 I=1,N
         X(I)=X(I)+DELX(I)
         VEL(I)=(X(I)-X0(I))/DT
   70 CONTINUE
      CALL FUNCA(Q,X,VEL,UM,W,T)
      IF(K.EQ.0) GO TO 700
      DO 80 I=1,N
         IF(ABS(DELX(I)).GT.TOL) GO TO 700
   80 CONTINUE
      GO TO 900
  600 DO 90 I=1,N
         X0(I)=X(I)
         X(I)=X(I)+VEL(I)*DT
   90 CONTINUE
         CALL FUNCA(P,X,VEL,UM,W,T)
      GO TO 800
  700 IF(K.EQ.ITMAX) GO TO 900
      DO 100 I=1,N
         DELP(I)=Q(I)-P(I)
  100 CONTINUE
      DO 103 I=1,N
         IF(DELP(I).NE.0.) GO TO 101
  103 CONTINUE
      DO 104 I=1,N
         DELX(I)=-Q(I)
  104 CONTINUE
      GO TO 102
  101 K=K+1
      MM=K
      DO 112 I=1,N
         DELPP(I)=DELP(I)
  112 CONTINUE
      CALL ORTHO(DELP,N,N,MM,QQ,R)
      DO 222 I=1,N
         Z(I)=QQ(I,MM)
  222 CONTINUE
      TT=R(MM,MM)
      E0=0.1**20
      E1=0.1**15
      E2=0.1**10
      E3=0.1**5
      ALPHA=1.
      IF(ABS(TT).LE.E3) ALPHA=10.**5
      IF(ABS(TT).LE.E2) ALPHA=10.**10
      IF(ABS(TT).LE.E1) ALPHA=10.**15
      IF(ABS(TT).LE.E0) GO TO 999
      TT=ALPHA*TT
      DO 120 I=1,N
         TA(I)=0.
         DO 121 J=1,N
```

```
          TA(I)=TA(I)+AJ(I,J)*DELPP(J)
121    CONTINUE
       TA(I)=TA(I)-DELX(I)
120 CONTINUE
    DO 140 I=1,N
       DO 130 J=1,N
          D(I,J)=ALPHA*TA(I)*Z(J)/TT
          AJ(I,J)=AJ(I,J)-D(I,J)
130    CONTINUE
140 CONTINUE
999 DO 141 I=1,N
       P(I)=Q(I)
141 CONTINUE
    GO TO 800
900 RETURN
    END
```

```
    SUBROUTINE INVERS(A,N)
    DIMENSION INDEX(10,2),A(10,10),B(10,10),C(10,10)
    DO 10 I=1,N
       DO 10 J=1,N
          B(I,J)=A(I,J)
 10 CONTINUE
    DO 20 I=1,N
       INDEX(I,1)=0
 20 CONTINUE
    II=0
 30 AMAX=-1.
    DO 40 I=1,N
       IF(INDEX(I,1)) 40,50,40
 50       DO 60 J=1,N
             IF(INDEX(J,1)) 60,70,60
 70             TEMP=ABS(A(I,J))
             IF(TEMP-AMAX) 60,60,80
 80             IROW=I
                ICOL=J
                AMAX=TEMP
 60       CONTINUE
 40 CONTINUE
    IF(AMAX) 190,220,90
 90    INDEX(ICOL,1)=IROW
    IF(IROW-ICOL) 110,100,110
110 DO 120 J=1,N
       TEMP=A(IROW,J)
       A(IROW,J)=A(ICOL,J)
       A(ICOL,J)=TEMP
120 CONTINUE
    II=II+1
    INDEX(II,2)=ICOL
100 PIVOT=A(ICOL,ICOL)
    A(ICOL,ICOL)=1.
    PIVOT=1./PIVOT
    DO 130 J=1,N
       A(ICOL,J)=A(ICOL,J)*PIVOT
130 CONTINUE
    DO 160 I=1,N
       IF(I-ICOL) 140,160,140
140       TEMP=A(I,ICOL)
          A(I,ICOL)=0.
```

```
          DO 150 J=1,N
             A(I,J)=A(I,J)-A(ICOL,J)*TEMP
150       CONTINUE
160 CONTINUE
    GO TO 30
170 ICOL=INDEX(II,2)
    IROW=INDEX(ICOL,1)
    DO 180 I=1,N
       TEMP=A(I,IROW)
       A(I,IROW)=A(I,ICOL)
       A(I,ICOL)=TEMP
180 CONTINUE
    II=II-1
190 IF(II) 170,200,170
200 DO 210 I=1,N
       DO 210 J=1,N
          C(I,J)=0.
          DO 210 K=1,N
             C(I,J)=C(I,J)+B(I,K)*A(K,J)
210 CONTINUE
220 RETURN
    END
```

```
    SUBROUTINE ORTHO(B,N,ND,M,Q,R)
    DIMENSION B(10),Q(10,10),R(10,10)
    IF(M.GT.N) GO TO 100
       IF(M.NE.1) GO TO 200
          DO 5 I=1,N
             DO 4 J=1,N
                Q(I,J)=0.
 4           CONTINUE
             Q(I,I)=1.
 5        CONTINUE
200    DO 11 I=1,N
          SUM=0.
          DO 10 J=1,N
             SUM=SUM+Q(J,I)*B(J)
 10       CONTINUE
          R(I,M)=SUM
 11    CONTINUE
       IF(M.GE.N) GO TO 700
          R0=0.
          DO 20 I=M,N
             R0=R0+R(I,M)*R(I,M)
 20      CONTINUE
         R0=SQRT(R0)
         IF(R(M,M).LT.0.) R0=-R0
            B(M)=R(M,M)+R0
            C=SQRT(R0*B(M))
            IF(C.LE.0.) GO TO 700
               R(M,M)=-R0
               B(M)=B(M)/C
               IF(M.GE.N) GO TO 500
                    MPN=M+1
                    DO 21 I=MPN,N
                       B(I)=R(I,M)/C
                       R(I,M)=0.
 21                 CONTINUE
500            DO 32 I=1,N
```

```
                  SUM=0.
                  DO 30 K=M,N
                     SUM=SUM+Q(I,K)*B(K)
 30               CONTINUE
                  DO 31 J=M,N
                     Q(I,J)=Q(I,J)-SUM*B(J)
 31               CONTINUE
 32            CONTINUE
   GO TO 700
100 M=N
   DO 40 I=1,N
      DO 40 J=1,N
         IF(J.LE.1) GO TO 40
            JJ=J-1
            R(I,JJ)=R(I,J)
 40 CONTINUE
   DO 43 I=1,N
      SUM=0.
      DO 42 J=1,N
         SUM=SUM+Q(J,I)*B(J)
 42   CONTINUE
      R(I,N)=SUM
 43 CONTINUE
   NPP=N-1
   DO 52 K=1,NPP
      KLK=K+1
      R0=SQRT(R(K,K)**2+R(KLK,K)**2)
      IF(R(K,K).LT.0.) R0=-R0
      BK=R(K,K)+R0
      BKP=R(KLK,K)
      C=R0*BK
      IF(C.EQ.0.) GO TO 52
         R(K,K)=-R0
         R(KLK,K)=0.
         DO 50 J=KLK,N
            R0=BK*R(K,J)+BKP*R(KLK,J)
            R(K,J)=R(K,J)-R0*BK/C
            R(KLK,J)=R(KLK,J)-R0*BKP/C
 50      CONTINUE
         DO 51 I=1,N
            R0=Q(I,K)*BK+Q(I,KLK)*BKP
            Q(I,K)=Q(I,K)-R0*BK/C
            Q(I,KLK)=Q(I,KLK)-R0*BKP/C
 51      CONTINUE
 52 CONTINUE
700 RETURN
   END
```

```
   SUBROUTINE FUNCA(FPLT,Y,V,U,W,T)
   DIMENSION FPLT(10),V(10),Y(10),UM(10)
   DIMENSION B(10,10),P0(10),U(10)
   DIMENSION W(10)
   COMMON N,EE,COT,COA,COM,COK,M
   COMMON CR(10,10),SC(10)
   H=1./(COK-COT*Y(1)+COA*Y(1)**2)
   P0(1)=Y(4)-V(1)
   PA1=(Y(1)-COT/(2.*COA))*Y(5)
   P0(4)=PA1*Y(5)-V(4)+W(1)
   P0(2)=Y(5)-V(2)
```

```
      PA2=H*(COT-2.*COA*Y(1))*Y(4)
      P0(5)=PA2*Y(5)-V(5)+W(2)
      P0(3)=Y(6)-V(3)
      P0(6)=-V(6)-9.8+W(3)
      CALL BMATX(B,Y,T)
      DO 60 I=1,N
         A=0.
         DO 50 K=1,M
            A=A+B(I,K)*U(K)
 50      CONTINUE
         FPLT(I)=P0(I)+A
 60   CONTINUE
      FPLT(4)=FPLT(4)/B(4,1)
      FPLT(5)=FPLT(5)/B(5,2)
      FPLT(6)=FPLT(6)/B(6,3)
      DO 100 I=1,N
         FPLT(I)=FPLT(I)/SC(I)
100   CONTINUE
      RETURN
      END


      SUBROUTINE TRANS(X,X0)
      DIMENSION X(10),X0(10)
      X(1)=SQRT(X0(1)**2+X0(2)**2)
      X(4)=SQRT(X0(4)**2+X0(5)**2)
      IF(X0(1).EQ.0.) X0(1)=0.1**10
      IF(X0(4).EQ.0.) X0(4)=0.1**10
         X(2)=ATAN(X0(2)/X0(1))
         X(5)=ATAN(X0(5)/X0(4))
      X(5)=X0(5)
      X(6)=X0(6)
      RETURN
      END


      SUBROUTINE BMATX(B,Y,T)
      DIMENSION B(10,10),Y(10)
      COMMON N,EE,COT,COA,COM,COK,M
      H=1./(COK-COT*Y(1)+COA*Y(1)**2)
      DO 10 I=1,N
         DO 10 J=1,N
            B(I,J)=0.
 10   CONTINUE
      B(4,1)=1./COA
      B(5,2)=H
      B(6,3)=1./(COM+COA)
      RETURN
      END


      SUBROUTINE CTR(U,ESTU,ESTY,ESTV,Y00,X,X00)
      DIMENSION U(10),X00(10),Y00(10),ESTY(10,10),ESTU(10,10)
      DIMENSION UM(10),UU(10)
```

```
      DIMENSION X(10),ESTV(10)
      COMMON N,EE,COT,COA,COM,COK,M
      COMMON CR(10,10),SC(10)

      GENERATE THE CONTROL SIGNAL, U FOR ROBOT SYSTEM
      BASED ON INDEPENDENT TRACKING AND REGULATION ALGORITHM

      D=1
      KN=1+N/2
      DO 10 I=KN,N
      SUM=0.
         DO 20 J=1,N
            SUM=SUM+(-ESTY(I,J)*Y00(J)+CR(I,J)*X00(J))
20       CONTINUE
         UU(I)=SUM+X(I)+ESTV(I)*D
10    CONTINUE
      DO 30 I=1,M
         II=I+M
         U(I)=UU(II)/ESTU(II,I)
30    CONTINUE
      RETURN
      END



      SUBROUTINE REGL(REG,XX,XX00)
      DIMENSION REG(10),XX(10),XX00(10)
      COMMON N,EE,COT,COA,COM,COK,M
      COMMON CR(10,10),SC(10)
      DO 10 I=1,N
         SUM=0.
         DO 20 J=1,N
            SUM=SUM+CR(I,J)*XX00(J)
20       CONTINUE
         REG(I)=XX(I)+SUM
10    CONTINUE
      RETURN
      END



      SUBROUTINE ERROR(CDIR,EPS,ELONG,ETRAS,UM,X)
      DIMENSION CDIR(10),DX(10),UM(10),X(10)
      DIMENSION UMIT(10),XIT(10)
      COMMON N,EE,COT,COA,COM,COK,M

      COMPUTE THE ERROR BETWEEN END POINT OF ROBOT ARM
      AND THE DESIRED TRAJECTORY

      CALL IVTR(UMIT,UM)
      CALL IVTR(XIT,X)
      SUM=0.
      DO 10 I=1,M
         DX(I)=XIT(I)-UMIT(I)
         SUM=SUM+DX(I)**2
10    CONTINUE
      EPS=SQRT(SUM)
      ELONG=0.
      DO 20 I=1,M
```

```
      ELONG=ELONG+DX(I)*CDIR(I)
 20 CONTINUE
    DIFF=EPS**2-ELONG**2
    IF(DIFF.LE.0.) GO TO 777
      ETRAS=SQRT(DIFF)
      GO TO 778
777 ETRAS=0.
778 RETURN
    END
```

```
    SUBROUTINE IVTR(XIT,X)
    DIMENSION XIT(10),X(10)
    XIT(1)=X(1)*COS(X(2))
    XIT(2)=X(1)*SIN(X(2))
    XIT(3)=X(3)
    RETURN
    END
```

```
    SUBROUTINE RANDOM(DSEED,NR,R,SR)
    DOUBLE PRECISION DSEED,DC,DM,DD
    DIMENSION R(10)

    GENERATE THE RANDOM PROCESS NOISE VECTOR

    DC=16807.D0
    DM=2147483647.D0
    DD=2144783648.D0
    DO 10 I=1,NR
      SUM=0.
      DO 20 J=1,3
        DSEED=DMOD(DC*DSEED,DM)
        SUM=SUM+DSEED/DD
 20   CONTINUE
      R(I)=(SUM-1.5)*2./SR
 10 CONTINUE
    RETURN
    END
```