A TWO-PARAMETER COMPENSATOR FOR

SET-POINT CONTROL AND NOISE REDUCTION

by

Jyan-Bang Chen

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Masters of Science

in the

Electrical Engineering

Program

Robert H. Foulkes Jr. 3-13-89

Adviser Date

Sally M. Hotchkiss March 15, 1989

Dean of the Graduate School Date

YOUNGSTOWN STATE UNIVERSITY

March, 1989

ABSTRACT

A TWO-PARAMETER COMPENSATOR FOR SET-POINT CONTROL AND NOISE REDUCTION

Jyan-Bang Chen

Masters of Science in Engineering

Youngstown State University, 1989

This thesis discusses a Two-Parameter method in the design of control systems. The system considered contains a linear dynamic analog plant, a digital controller, and suitable interface hardware. The implementable transfer function is developed and implemented on a digital computer. The Two-Parameter configuration is introduced to realize the implementable transfer function, and two compensator transfer functions are obtained by solving linear algebraic equations.

Finally, several examples illustrate the application of this method and explain how chosen design parameters affect the reduction of noise.

ACKNOWLEDGMENTS

A number of individuals contributed to the successful completion of this work, and I would like to dedicate a word to those who somehow contributed to my educational experience.

I thank Dr. Robert H. Foulkes for having given me the opportunity to work on this project and for his constant support, patience and the numberless guidance hours. I thank Dr. Matthew Siman and Professor Raymond Kramer for being on my committee.

I thank my colleague, Mr. Nai-Chain Guo, for his help and companionship during working on this project. Also, I thank Mr. Michael Repetski and Mr. Richard L. Laughlin for providing the necessary components and equipment.

My sincerest appreciation and deep gratitude goes to my entire family for their constant encouragement, support and understanding throughout my learning experiences.

TABLE OF CONTENTS

LIST OF SYMBOLS

SYMBOLS	DEFINITION
$G(z)$	Open loop transfer function
$G_o(z)$	Closed loop transfer function
$N(z)$	Numerator polynomial of $G(z)$
$D(z)$	Denominator polynomial of $G(z)$
$N_o(z)$	Numerator polynomial of $G_o(z)$
$D_o(z)$	Denominator polynomial of $G_o(z)$
$C_1(z)$	Compensator of $G(z)$
$C_2(z)$	Compensator of $G(z)$
$L(z)$	Numerator polynomial of $C_1(z)$
$M(z)$	Numerator polynomial of $C_2(z)$
$A(z)$	Denominator polynomial of $C_1(z)$ and $C_2(z)$
$\overline{D}_p(z)$	Polynomial of cancelled poles
$y(k)$	Discrete time output signal
$u(k)$	Discrete time input signal
ζ	The damping ratio
W_n	The natural frequency

LIST OF FIGURES

LIST OF TABLES

Chapter I
Introduction

1.1 Background and Objective

The design method in this thesis is based on an article by C.T. Chen that appeared in the IEEE Control Systems Magazine [1]. Chen discusses some aspects of the linear algebraic method of control system design as applied to analog systems. These methods can also be directly applied to digital control system design. The objective of the design is concerned with the design configuration shown as Figure 1.1.1. G(s) is a fixed plant with input or control signal u(t) and C(z) is a digital controller. The output signal y(t) is to follow the reference signal r(t) while satisfying the specifications on such performance measures as settling time, overshoot, and steady state error

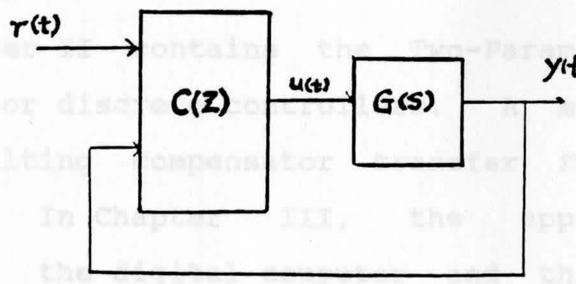

Figure 1.1.1 Control system block diagram

In the design of control systems, there are two approaches, inward and outward [1]. In the outward approach, the first step is to choose a configuration and a compensator with unspecified parameters. Then the second step is to search for some parameter values such that the resulting overall system will meet all the specifications. Both the root-locus method and the frequency-domain method are used in this approach.

In the inward approach, the first step is to find an overall system to meet all the specifications. Then the compensator configuration is chosen and the required compensator transfer function is computed algebraically. In this thesis, a two-degree-of-freedom compensator (or two-parameter configuration) is used in approach.

The method to be discussed in this thesis will be called the linear algebraic method, because once an overall system is chosen, the design can be completed by solving a set of linear algebraic equations.

1.2 Overview

Chapter II contains the Two-Parameter method design procedure for discrete controllers. A minimal realization of the resulting compensator transfer function is also introduced. In Chapter III, the appropriate hardware interfacing the digital computer and the analog plant is introduced. A brief description of a computer program for real-time implementation of the digital compensator is also given. In Chapter IV, a variety of open-loop system examples are explored. Several designs for each example are presented and discussed in order to illustrate how the design for a set-point control affects the noise-handling capability of the closed-loop system.

Finally, Chapter V contains a conclusion along with recommendations for future work.

Chapter II

THE LINEAR ALGEBRAIC METHOD

2.1 INTRODUCTION

It is the purpose of this chapter to extend the design procedure in [1] to discrete-time systems. The only difference is that all poles must be located inside the unit circle of the z-plane to ensure stability rather than inside the open left-half s-plane as in the continuous-time case. Section 2.2 contains the conditions under which a design will be successful. Section 2.3 contains the design procedure. In section 2.4, the minimal realization of the digital compensator is presented and discussed in detail.

2.2 IMPLEMENTABLE TRANSFER FUNCTION

Before proceeding, it is important to realize what conditions must be satisfied for a control system design to be successful. To be successful, the design must be capable of being built with real-world hardware, and the behavior of the resulting hardware must meet the specifications. The first condition is achieved by a suitable choice of the closed-loop transfer function $G_o(z)=N_o(z)/D_o(z)$. The second condition is achieved by ensuring that certain constraints are satisfied by the choice of $G_o(z)$. Chen calls such transfer functions "implementable" [1].

Consider a plant with a proper transfer function $G(z)=N(z)/D(z)$, then $G_o(z)=N_o(z)/D_o(z)$ is implementable if and only if [1]:

(1) $D_o(z)$ is Hurwitz.

(2) The degree of $D_o(z)$ minus the degree of $N_o(z)$ is greater than or equal to the degree of D(z) minus of N(z).

(3) All zeros of N(z) on or outside the unit circle are retained in $N_o(z)$ (retainment of nonminimum-phase zeros).

2.3 DESIGN PROCEDURE: A Two-Parameter CONFIGURATION

Figure (2.3.1) shows a Two-Parameter configuration for continuous-time systems considered by Chen in [1]. It consists of a plant $G_1(s)$ and a compensator with transfer functions $C_1(s)$ and $C_2(s)$ (the " two-parameters").

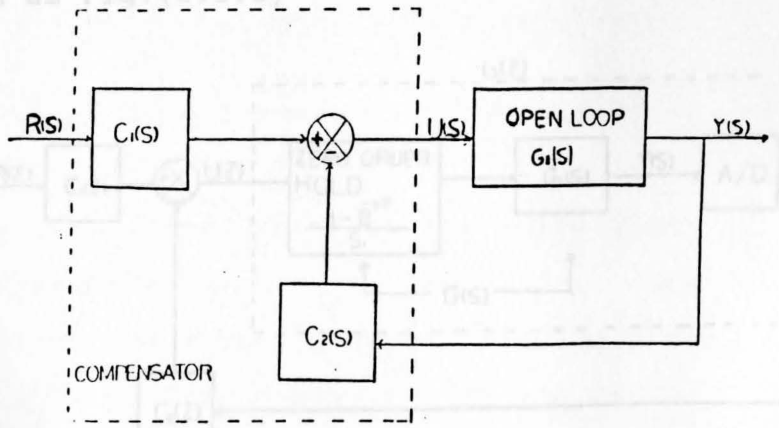

Fig. 2.3.1. The continuous-time system for a Two-Parameter configuration

Fig. 2.3.2 shows a similar configuration for sampled-data control. Here, the plant is interfaced to the digital controller by means of an A/D converter and a D/A converter.

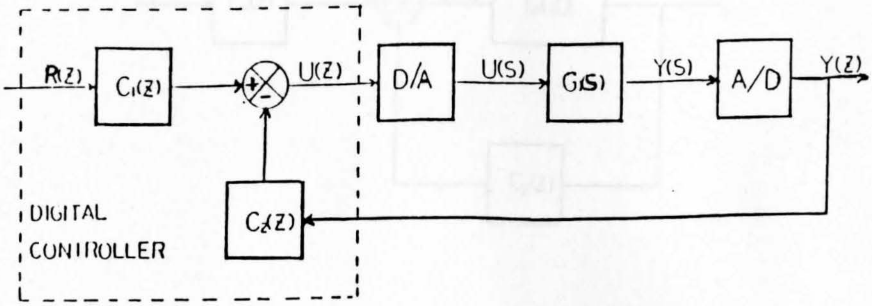

Fig. 2.3.2 A Two-Parameter Digital Controller

For the design procedure, a discrete-time model of the D/A, the plant, and the A/D is needed. The D/A converter samples and holds the controller output. Mathematically, the process of sampling signal and holding by means of a Zero-Order-Hold can be considered equivalent to impulse sampling followed by a transfer function $(1-e^{-TS})/s$, as shown as Fig.(2.3.3)

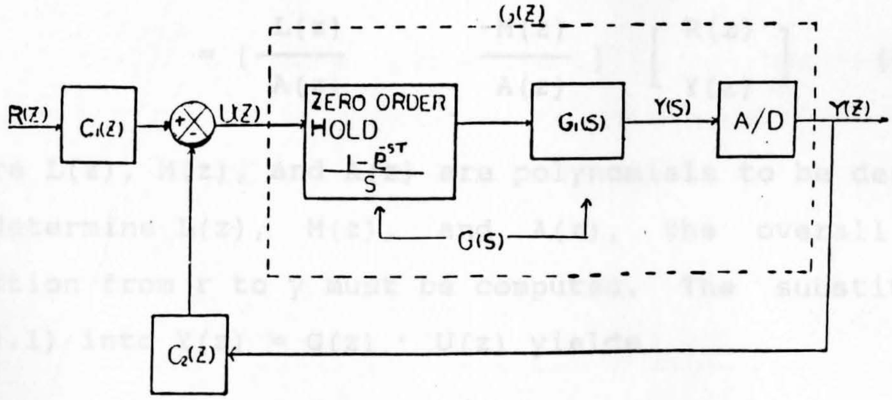

Fig. 2.3.3. Block diagram rearranged to show digital relations.

Finally, the overall discrete-time system is shown in Fig.(2.3.4)

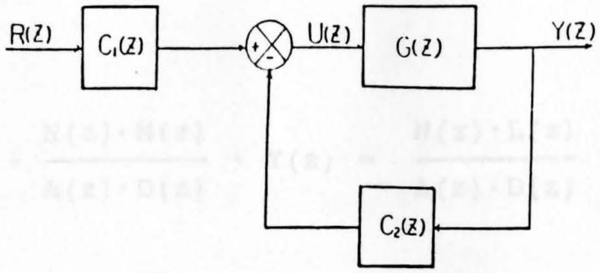

Fig. 2.3.4. Discrete-time control system for a Two-Parameter configuration.

G(z) is the Zero-Order-Hold equivalent transfer function of $G_1(s)$ together with the D/A and A/D converters, and $C_1(z)$ and $C_2(z)$ are the two compensator transfer functions. In the design procedure, $C_1(z) = L(z)/A(z)$ and $C_2(z)=M(z)/A(z)$, where L(z), M(z), and A(z) are polynomials in z. The compensator output is given by

$$U(z) = C_1(z)\cdot R(z) - C_2(z)\cdot Y(z)$$

$$= \left[\frac{L(z)}{A(z)} \quad \frac{-M(z)}{A(z)} \right] \begin{bmatrix} R(z) \\ Y(z) \end{bmatrix} \qquad (2.3.1)$$

where L(z), M(z), and A(z) are polynomials to be designed. To determine L(z), M(z), and A(z), the overall transfer function from r to y must be computed. The substitution of (2.3.1) into Y(z) = G(z) · U(z) yields

$$Y(z) = \frac{N(z)}{D(z)} \cdot \left[\frac{L(z)}{A(z)} \cdot R(z) - \frac{M(z)}{A(z)} \cdot Y(z) \right].$$

Hence,

$$Y(z) + \frac{N(z)\cdot M(z)}{A(z)\cdot D(z)} \cdot Y(z) = \frac{N(z)\cdot L(z)}{A(z)\cdot D(z)} \cdot R(z),$$

and

$$\left[1+\frac{N(z)\cdot M(z)}{A(z)\cdot D(z)}\right] \cdot Y(z) = \frac{N(z)\cdot D(z)}{A(z)\cdot D(z)} \cdot R(z).$$

Thus the transfer function from r to y is

$$\frac{Y(z)}{R(z)} = \frac{\dfrac{D(z)\cdot L(z)}{A(z)\cdot D(z)}}{1+\dfrac{N(z)\cdot M(z)}{A(z)\cdot D(z)}} = \frac{N(z)\cdot L(z)}{A(z)\cdot D(z)+M(z)\cdot N(z)} \quad (2.3.4)$$

Adapting Chen's procedure [1] to the present case yields the following procedure to determine L(z), M(z), and A(z). It is assumed that G(z) is proper.

Step 1: Compute

$$\frac{G_o(z)}{N(z)} = \frac{N_o(z)}{D_o(z)\cdot N(z)} = \frac{N_p(z)}{D_p(z)} \quad (2.3.5)$$

where $N_p(z)$ and $D_p(z)$ are coprime. If $N_o(z)$ and $D_o(z)$ are coprime, then common factors may exist only between $N_o(z)$ and $N(z)$. Cancel all the common factors between them. If $N_o(z) = N(z)$, then $D_p(z) = D_o(z)$ and $N_p(z)=1$.

Step 2: If deg $D_p(z) = P < 2n-1$, introduce an arbitrary polynomial $\bar{D}_p(z)$ of degree 2n-1-p. Note that if deg $D_o(z)$ = deg $D(z) = n$, then deg $\bar{D}_p(z) = n-1$. Because this polynomial will be cancelled in the design, its roots should be chosen inside an acceptable pole-zero cancellation region [1]. For the purpose of designing the transient response from the reference to the output, the cancelled poles must be stable, i.e., inside the unit circle. However, not all the stable locations give good results when the system is subjected to noise. The selection of bandwidth from noise inputs to system outputs must be done carefully. The effect of the choice of the control bandwidth and the cancelled poles on the system response will be demonstrated in the examples presented later.

Step 3: Rewrite Eq. (2.3.5) as:

$$Go(z) = \frac{N(z)\ D(z)}{Dp(z)} = \frac{N(z)\ [Np(z)\ \bar{D}p(z)]}{Dp(z)\ \bar{D}p(z)} \quad (2.3.6)$$

A comparison of (2.3.4) and (2.3.6) gives

$$L(z) = Np(z)\cdot\bar{D}p(z) \quad (2.3.7)$$

$$A(z)\cdot D(z)+M(z)\cdot N(z) = Dp(z)\ \bar{D}p(z) = F(z) \quad (2.3.8)$$

where deg $F(z)$ = deg $Dp(z)$ + deg $\bar{D}p(z)$

$= (2n-1-p) + p = 2n-1$

For $n = \deg D(z) > \deg N(z)$ and $\deg A(z) > \deg M(z)$

$$\deg F(z) = \deg D(z) + \deg A(z)$$

Substituting $2n-1 = n + \deg A(z)$

yields $\deg A(z) = n-1$

The polynomial equation in (2.3.8) is transferred into a set of linear algebraic equations in order to solve it.

Let A(z), M(z) and F(z) be expressed as

$$\left.\begin{aligned} A(z) &= A_0 + A_1 z + A_2 z^2 + \cdots + A_{n-1} z^{n-1} \\ M(z) &= M_0 + M_1 z + M_2 z^2 + \cdots + M_{n-1} z^{n-1} \\ F(z) &= F_0 + F_1 z + F_2 z^2 + \cdots + F_{2n-1} z^{2n-1} \end{aligned}\right] \quad (2.3.9)$$

then A(z) and M(z) can be solved from the following linear algebraic equations.

A comparison of (2.3.9) and (2.3.8) gives

$$(A_0+A_1z+A_2z^2+ \cdots +A_{n-1}z^{n-1}) \quad (D_0z+D_1z+D_2z^2+ \cdots +D_nz^n)$$
$$+ (M_0+M_1z+M_2z^2+ \cdots +M_{n-1}z^{n-1}) \quad (N_0+N_1z+N_2z^2+ \cdots +N_nz^n)$$
$$= F_0 + F_1z + F_2z^2 + \cdots + F_{n-1}z^{n-1}$$

then,

$$(A_0D_0+M_0N_0) + (A_0D_1+M_0N_1+A_1D_0+M_1N_0)z + \cdots$$
$$+ (A_{n-1}D_n+M_{n-1}N_n)z^{2n-1} = F_0+F_1z+F_2z_2+ \cdots +F_{2n-1}Z^{2n-1}$$

This polynomial equation holds if and only if:

$$
\begin{aligned}
A_0D_0 + M_0N_0 &= F_0 \\
A_0D_1+M_0N_1+A_1D_0+M_1N_0 &= F_1 \\
. \quad . \quad . &= . \\
. \quad . \quad . &= . \\
. \quad . \quad . &= . \\
A_{n-1}D_n + N_{n-1}N_n &= F_{2n-1}
\end{aligned}
$$

There are 2n equations in 2n unknowns. These equations can be arranged in a matrix form as follows:

$$
\begin{bmatrix}
D_0 & N_0 & 0 & 0 & & 0 & 0 \\
D_1 & N_1 & D_0 & N_0 & & \vdots & \vdots \\
\vdots & \vdots & D_1 & N_1 & & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\
D_n & N_n & D_{n-1} & N_{n-1} & \cdots & 0 & 0 \\
0 & 0 & D_n & N_n & & D_0 & N_0 \\
0 & 0 & \vdots & \vdots & & D_1 & N_1 \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & 0 & & D_n & M_n
\end{bmatrix}
\cdot
\begin{bmatrix}
A_0 \\ M_0 \\ A_1 \\ M_1 \\ \vdots \\ \vdots \\ A_{n-1} \\ M_{n-1}
\end{bmatrix}
=
\begin{bmatrix}
F_0 \\ F_1 \\ F_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ F_{2n-1}
\end{bmatrix}
$$

The solution of the above equation (2.3.10) and the $L(z)$ in (2.3.7) will implement the overall system $G_o(z)$. Compensator $[C_1(z) \quad C_2(z)]$ will be in the form of

$$
C(z) = [C_1(z) \quad C_2(z)] = \left[\frac{L(z)}{A(z)} \quad \frac{-M(z)}{A(z)} \right]
$$

$$= \left[\frac{N_p(z)\bar{D}_p(z)}{A_0+\cdot\cdot+A_{n-1}z^{n-1}} - \frac{M_0+M_1z+.....+M_{n-1}z^{n-1}}{A_0+A_1z+.....+A_{n-1}z^{n-1}} \right]$$

2.4 MINIMAL REALIZATION OF VECTOR TRANSFER FUNCTIONS

To carry out the implementation of the compensator on a digital computer, the realization problem will be discussed in this section. It is simpler and more systematic to simulate transfer function by developing the state variable equations.

To exemplify the process, consider a 4th-order proper transfer function G(z):

$$G(z) = \frac{b_4z^4+b_3z^3+b_2z^2+b_1z+b_0}{a_4z^4+a_3z^3+a_2z^2+a_1z+a_0} = \frac{N(z)}{D(z)} = \frac{Y(z)}{U(z)} \qquad (2.4.1)$$

where a_i and b_i are real constants and $a_4 \neq 0$. If $b_4 \neq 0$, the transfer function is proper; if $b_4=0$, it is strictly proper [2].

First, reduce G(z) to strictly proper form. From Eq.(2.4.1) $G(\infty) = b_4/a_4 = d$, and

$$G_z(z) = G(z) - G(\infty) = \frac{b_4z^4+b_3z^3+b_2z^2+b_1z+b_0}{a_4z^4+a_3z^3+a_2z^2+a_1z+a_0} - d$$

$$= \frac{\beta_3 z^3+\beta_2 z^2+\beta_1 z+\beta_0}{a_4 z^4+a_3 z^3+a_2 z^2+a_1 z+a_0}$$

Second, the leading coefficient of the denominator is normalized to 1.

$$G_z(z)= \frac{b'_3 z^3+b'_2 z^2+b'_1 z+b'_0}{z^4+a'_3 z^3+a'_2 z^2+a'_1 z+a'_0} = \frac{N'(z)}{D'(z)}$$

G(z) in Eq.(2.4.1) can be rewritten as:

$$G(z)= \frac{Y(z)}{U(z)} = G(\infty) + \frac{N'(z)}{D'(z)} = d + G_z(z)$$

or $Y(z)=G(z)U(z)=dU(z)+G_z(z)U(z)=Y_d(z)+Y_z(z)$ (2.4.2)

$$Y_z(z)=G_z(z)U(z)= \frac{b'_3 z^3+b'_2 z^2+b'_1 z+b'_0}{z^4+a'_3 z^3+a'_2 z^2+a'_1 z+a'_0} U(z)$$

$$= \frac{b'_3 z^{-1}+b'_2 z^{-2}+b'_1 z^{-3}+b'_0 z^{-4}}{1+a'_3 z^{-1}+a'_2 z^{-2}+a'_1 z^{-3}+a'_0 z^{-4}} U(z)$$

$$Y_z(z)\cdot(1+a'_3 z^{-1}+a'_2 z^{-2}+a'_1 z^{-3}+a'_0 z^{-4})$$
$$= U_z(z)\cdot(b'_3 z^{-1}+b'_2 z^{-2}+b'_1 z^{-3}+b'_0 z^{-4})$$

This may be modified to

$$Y_Z(z) = z^{-1}\Big[(b'_3\ U(z)-a'_3\ Y(z)) + Z^{-1}\ \{\ (b'_2\ U(z)-a'_2\ Y(z)) + Z^{-1}\ [(b'_1\ U(z)-a'_1\ Y(z)) + Z^{-1}\ (b'_0\ U(z)-a'_0\ Y(z))]\ \}\Big]$$

Now, define the state variables as follows:

$$\begin{aligned} X_4 &= z^{-1}\ [b'_3\ U(z)-a'_3\ Y(z)+ X_3] \\ X_3 &= z^{-1}\ [(b'_2\ U(z)-a'_2\ Y(z))+ X_2] \\ X_2 &= z^{-1}\ [(b'_1 U(z)-a'_1 Y(z))+ X_1] \\ X_1 &= z^{-1}\ (b'_0\ U(z)-a'_0\ Y(z)) \end{aligned} \qquad (2.4.3)$$

Then Eq.(2.4.3) can be written in the form:

$$Y(z)=X_4 \qquad (2.4.4)$$

By substituting Eq.(2.4.4) into Eq.(2.4.3) and multiplying both sides of the equations by z gives:

$$\left.\begin{aligned} X_4(k+1) &= b'_3U(k)-a'_3X_4(k)+X_3(k) \\ X_3(k+1) &= b'_2U(k)-a'_2X_4(k)+ X_2(k) \\ X_2(k+1) &= b'_1U(k)-a'_1X_4(k)+ X_1(k) \\ X_1(k+1) &= b'_0U(k)-a'_0X_4(k) \end{aligned}\right] \qquad (2.4.5)$$

which, combined with Eq.(2.4.2) and Eq.(2.4.5), can be arranged in matrix form as:

$$X(k+1)=\begin{bmatrix} 0 & 0 & 0 & -a'_0 \\ 1 & 0 & 0 & -a'_1 \\ 0 & 1 & 0 & -a'_2 \\ 0 & 0 & 1 & -a'_3 \end{bmatrix} X(k) + \begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \end{bmatrix} U(k) \qquad (2.4.6)$$

$$Y(k)= [\ 0 \quad 0 \quad 0 \quad 1\ ] \cdot X(k) + d \cdot U(k)$$

The state space model given by Eq.(2.4.6) is called an observable canonical form. Figure 2.4.1 shows the block diagram of the system given by Eq.(2.4.6)

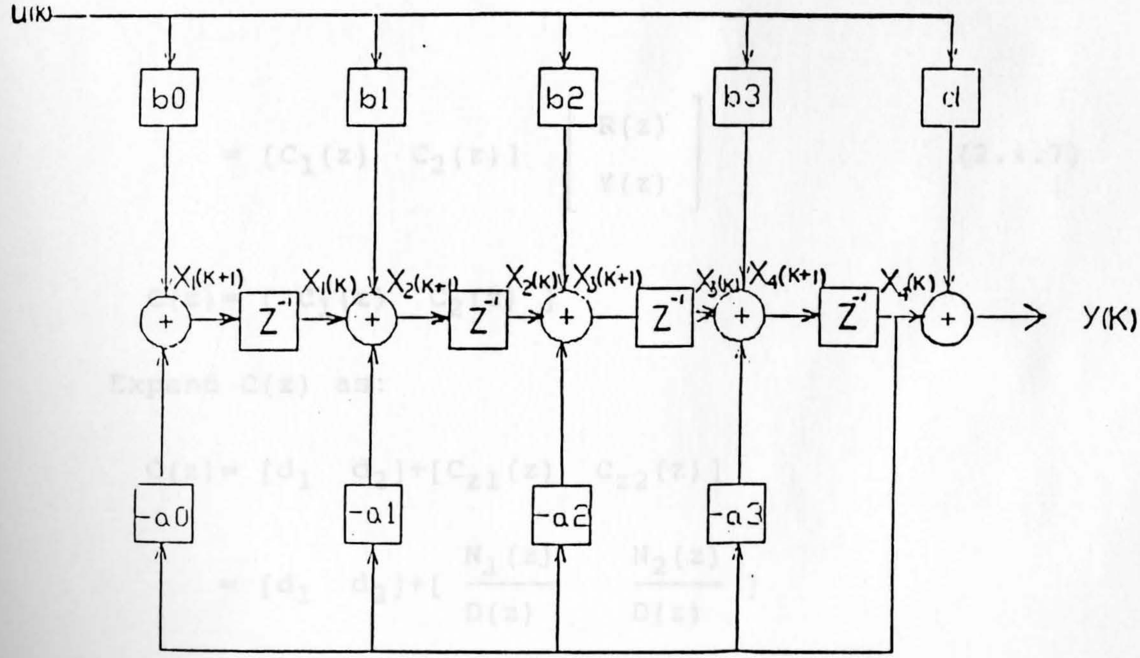

Figure 2.4.1. Block diagram representation of the system given by Eq.2.4.6

Consider a two-input one-output system shown as Fig.2.4.2

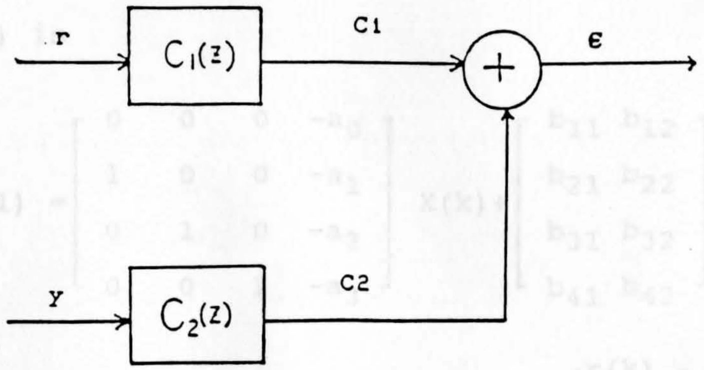

Fig.2.4.2 Two-input one-output system

The output and inputs can be expressed as:

$$E(z)= C_1(z)\cdot R(z) + C_2(z)\cdot Y(z)$$

$$= [C_1(z) \quad C_2(z)] \begin{bmatrix} R(z) \\ Y(z) \end{bmatrix} \qquad (2.4.7)$$

$$C(z)= [\ C_1(z) \quad C_2(z)\ ]$$

Expand C(z) as:

$$C(z)= [d_1 \quad d_2]+[C_{z1}(z) \quad C_{z2}(z)]$$

$$= [d_1 \quad d_2]+[\ \frac{N_1(z)}{D(z)} \quad \frac{N_2(z)}{D(z)}\ ]$$

Assume, $N_1(z)= b_{41}Z^3+b_{31}Z^2+b_{21}Z+b_{11}$

$N_2(z)= b_{42}Z^3+b_{32}Z^2+b_{22}Z+b_{12}$

$D(z) = Z^4+a_3Z^3+a_2Z^2+a_1Z+a_0$

then, using the former procedure, a minimal realization of Eq.(2.4.7) is

$$X(k+1) = \begin{bmatrix} 0 & 0 & 0 & -a_0 \\ 1 & 0 & 0 & -a_1 \\ 0 & 1 & 0 & -a_2 \\ 0 & 0 & 1 & -a_3 \end{bmatrix} X(k) + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \begin{bmatrix} r(k) \\ y(k) \end{bmatrix}$$

$$C(k) = [0 \quad 0 \quad 0 \quad 1]\, X(k) + [d_1 \quad d_2] \begin{bmatrix} r(k) \\ y(k) \end{bmatrix} \qquad (2.4.8)$$

This is a combination of the two observable form realizations of $C_1(z)$ and $C_2(z)$. Also the block diagram of Eq.(2.4.8) shown as Fig.(2.4.3)

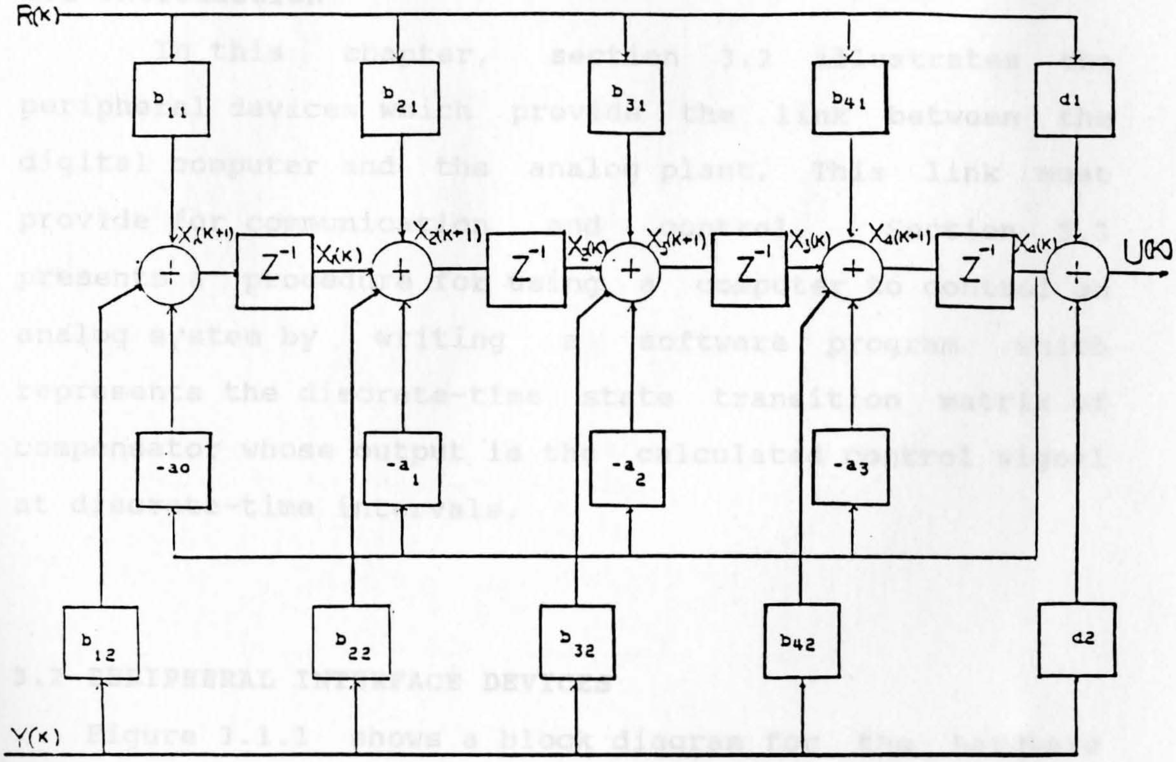

Fig. 2.4.3. Block diagram representation of the system given by Eq.(2.4.8)

Chapter III

REAL TIME IMPLEMENTATION

3.1 Introduction

In this chapter, section 3.2 illustrates the peripheral devices which provide the link between the digital computer and the analog plant. This link must provide for communication and control. Section 3.3 presents a procedure for using a computer to control an analog system by writing a software program which represents the discrete-time state transition matrix of compensator whose output is the calculated control signal at discrete-time intervals.

3.2 PERIPHERAL INTERFACE DEVICES

Figure 3.1.1 shows a block diagram for the hardware implementation. The following paragraphs give a brief description of the various parts of the interfacing hardware.

The 8255A programmable peripheral interface shown in Figure 3.2.1 and Figure 3.2.2 is used to interface the TRS 80 MODEL 4 computer [7] to the real world. The 8255A contains a control register, a status register, and three 8-bit I/O ports: A , B and C. An 8-bit data bus transfers data between the external data bus and the control register, status register, or one of the I/O ports. The 8255A is selected by a "LOW" signal at its chip select

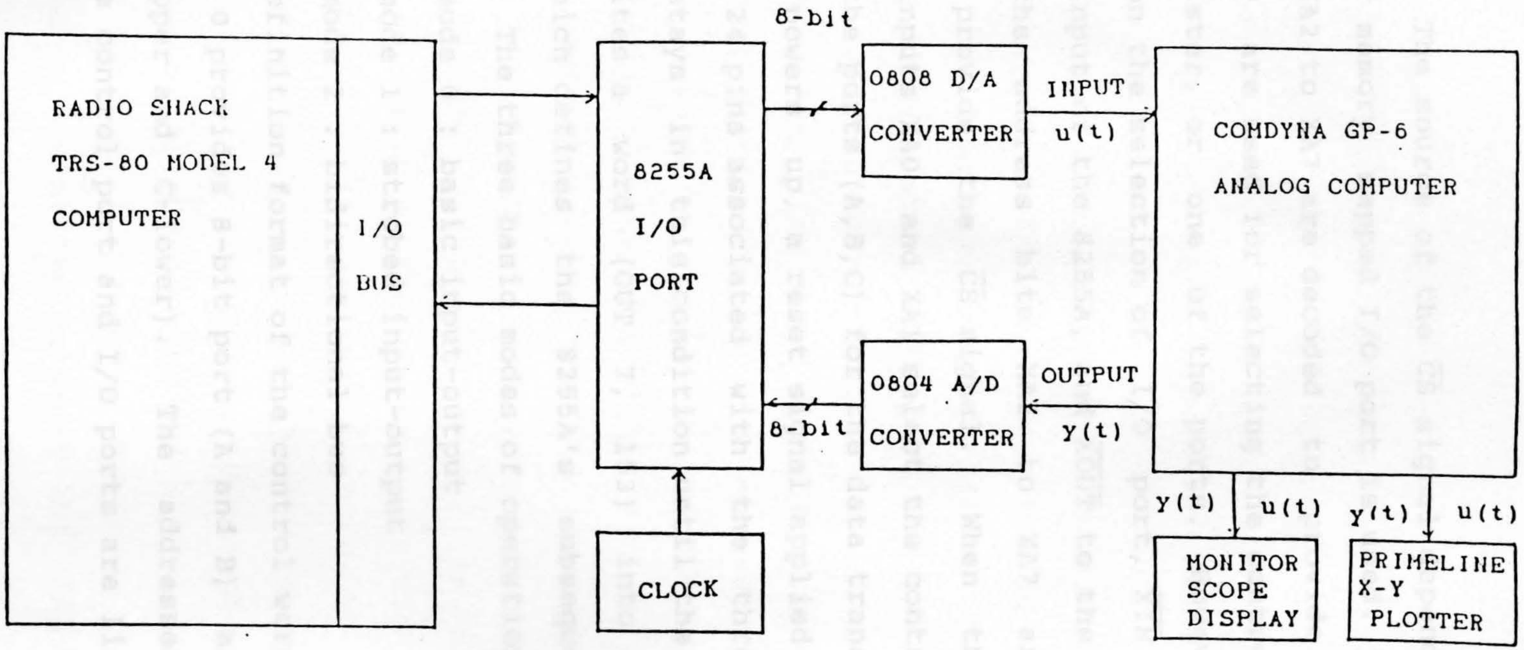

Fig. 3.1.1 The organization diagram for hardware interface

input, $\overline{CS}$. The source of the $\overline{CS}$ signal depends on whether isolated or memory mapped I/O port is used. For isolated I/O, bits XA2 to XA7 are decoded to provide $\overline{CS}$, and bits XA1 and XA0 are used for selecting the control register, status register, or one of the ports. However, in order to condition the selection of I/O port, $\overline{XIN}$ is connected to the $\overline{RD}$ input of the 8255A, and $\overline{XOUT}$ to the $\overline{WR}$ input of 8255A and the address bits XA2 to XA7 are decoded by 74LS138 to provide the $\overline{CS}$ signal. When the 8255A is selected, inputs XA0 and XA1 select the control register or one of the ports (A,B,C) for the data transfer [9]. As the system powers up, a reset signal applied to the 8255A floats all 24 pins associated with the three I/O ports. The 8255A stays in this condition until the application program writes a word (OUT 7, 153) into the control register which defines the 8255A's subsequent mode of operation. The three basic modes of operation are [9]:

1. mode 0 : basic input-output
2. mode 1 : strobed input-output
3. mode 2 : bidirectional bus

The mode definition format of the control word is shown in [9]. Mode 0 provides 8-bit port (A and B) and two 4-bit ports (C-upper and C-lower). The addresses chosen to perform the control port and I/O ports are listed below:

Address	Port	Function
04	A	input port
05	B	output port
06	C	input port (clock in)
07	control	control ports

The digital-to-analog converter (DAC0808) is a device that will convert a binary word applied to its inputs via 8255A port B to a proportional voltage at the output. The interfacing diagram shown in Figure 3.2.3. The reference voltage is a voltage source used to supply power to the current generators. In many designs the output current per bit weight is proportional to the reference voltage. The current summing is achieved by using a current follower circuit to provide a virtual-common summing point for the current generators and to convert the current signal to a proportional voltage [8].

An A/D 0804 converter shown as Figure 3.3.4. transforms an analog voltage Vx of a certain range (0 to 5 volts) into an 8-bit binary output at the D7 to D0 pins. A pulse can be applied at the 0804ADC converter's $\overline{CS}$ pin in order to start the conversion. Since the output of the A/D converter is triggered, the LOW signal start at $\overline{WR}$ pin sending the message to the 8255A port's B. The conversion is initiated by a starting pulse by the timing and control command A/D conversion time depends on the number of bits desired. Typical figures are 150 microseconds for 8 bits [9]. The final analog-to-digital voltage can never approximate the input voltage with an accuracy greater than one-half the value of the least significant bit. Therefore, the overall converter accuracy is equal to plus or minus one-half the voltage values of the least significant bit.

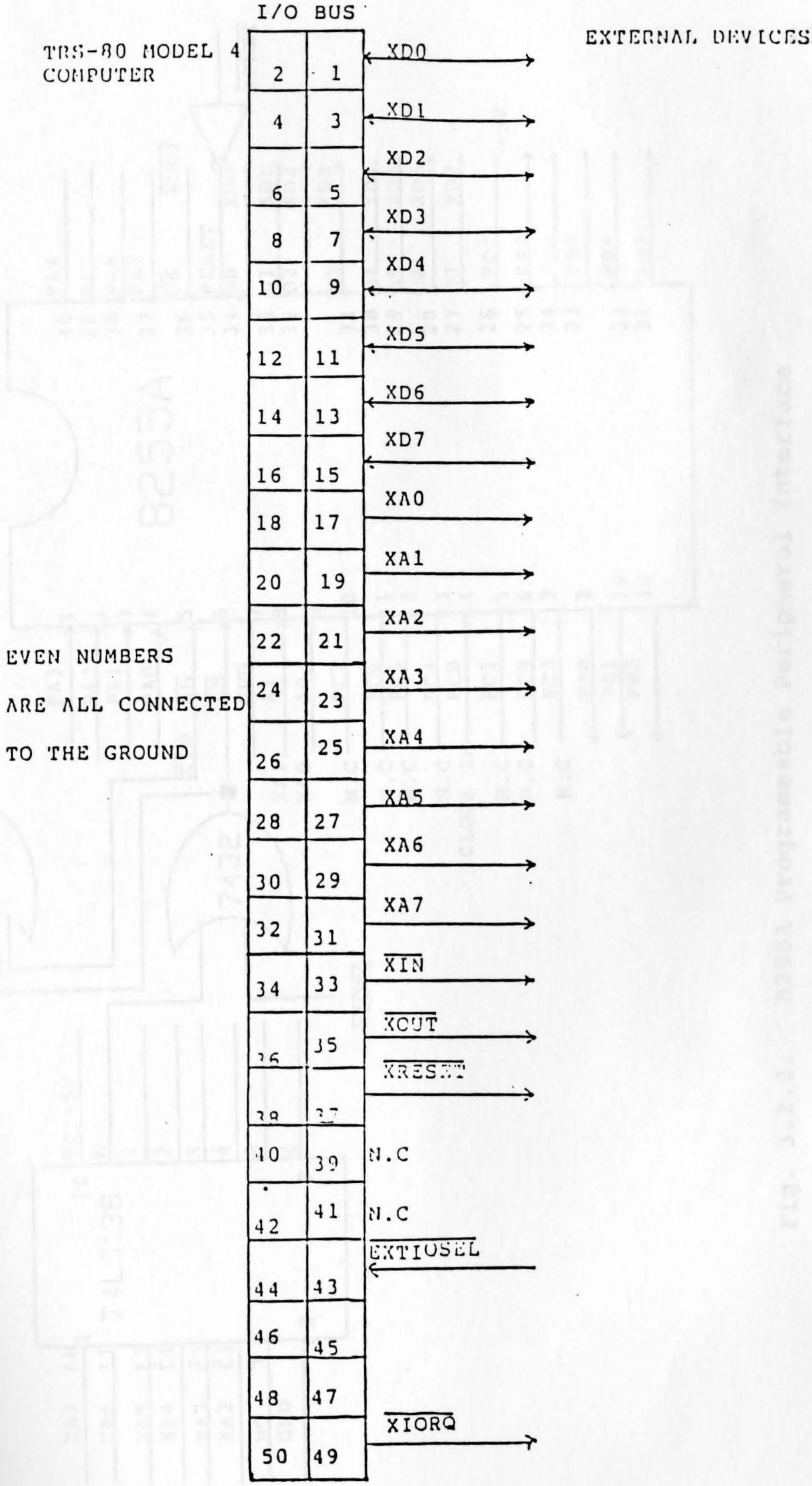

Fig. 3.2.1. I/O bus connection

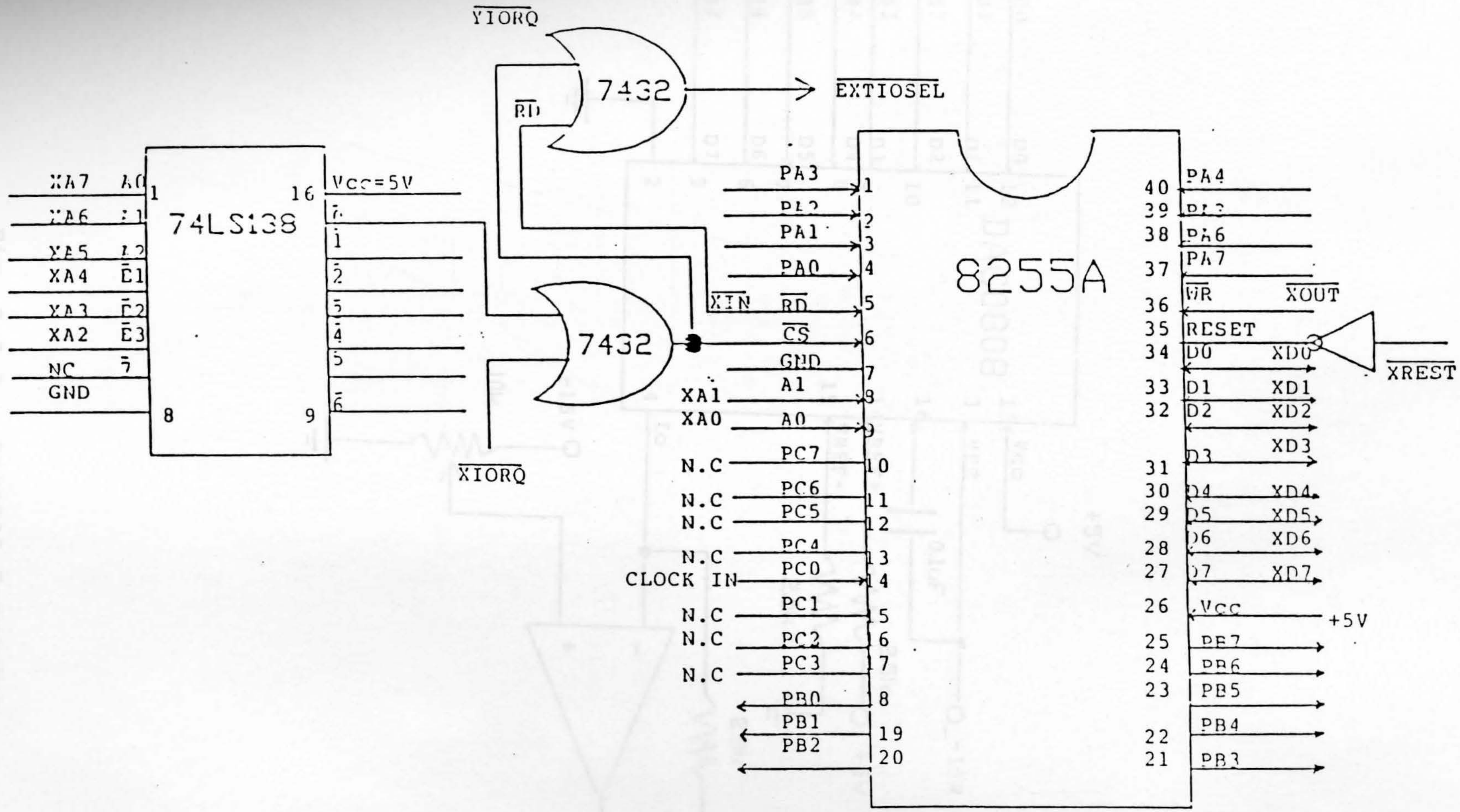

Fig. 3.2.2. 8255A Programmable Peripheral Interface

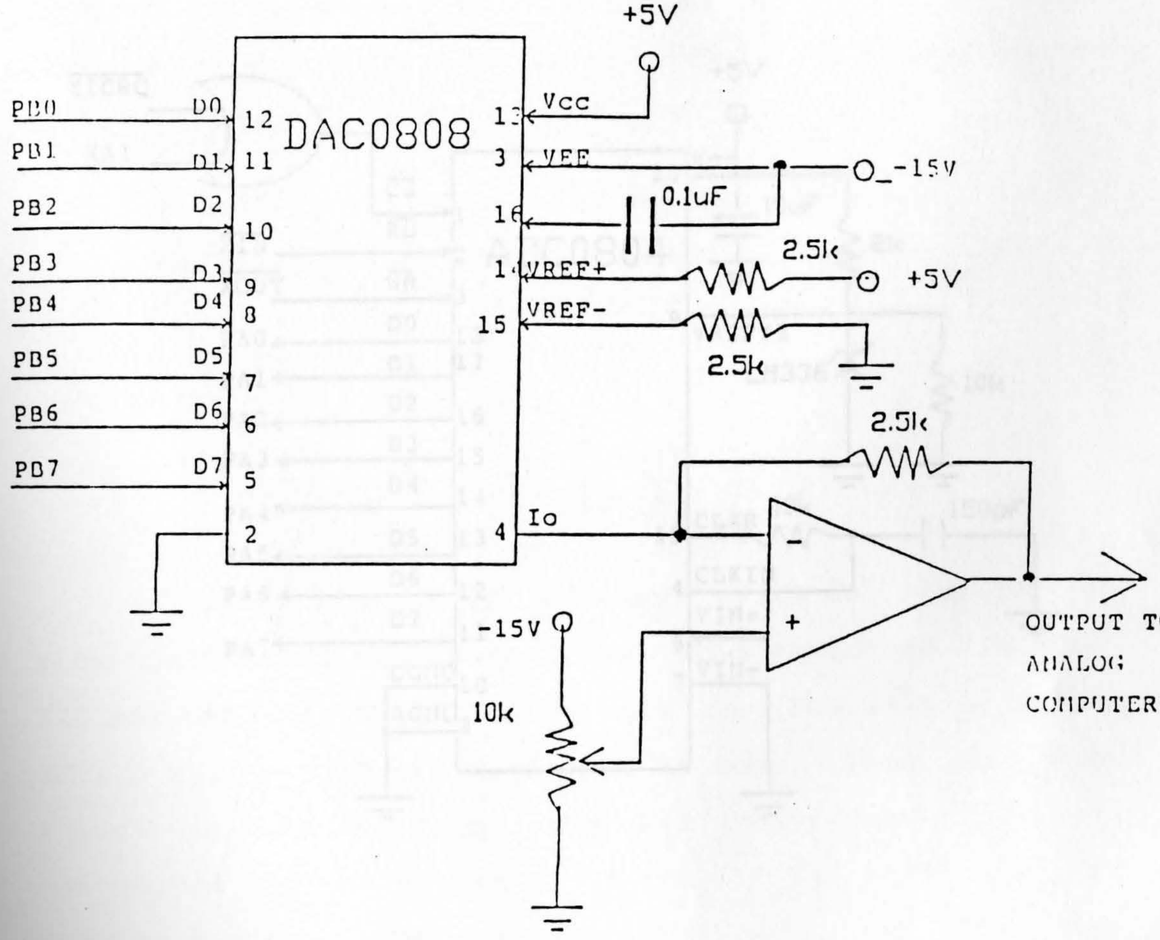

Fig. 3.2.3. DAC0808 Converter

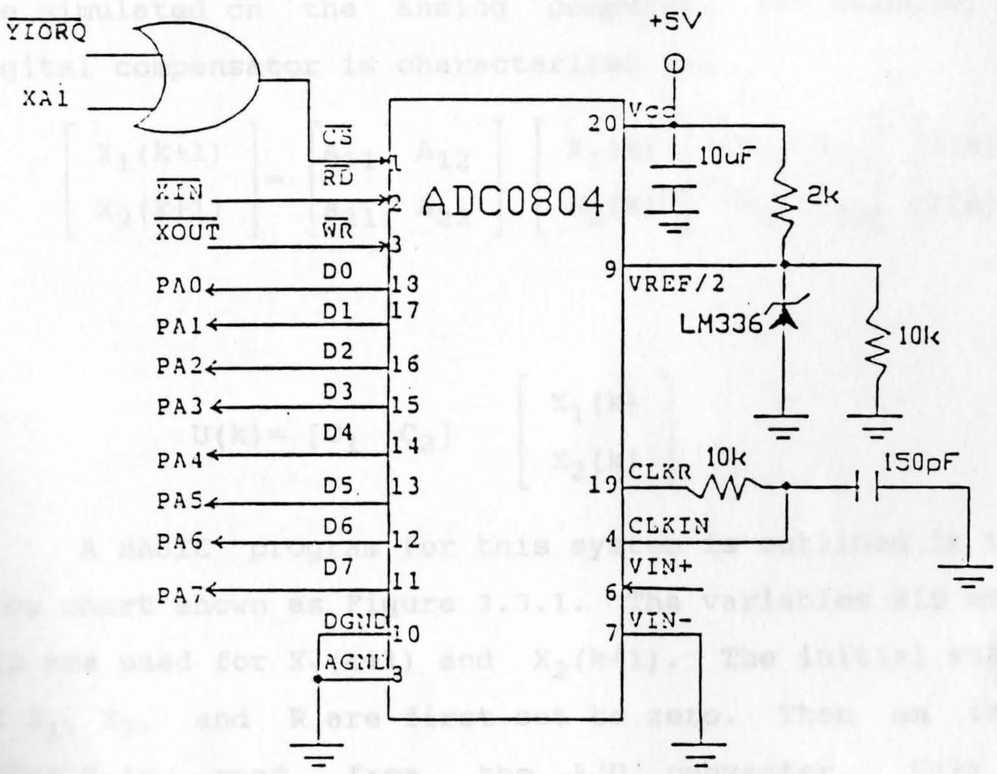

Fig. 3.2.4. ADC0804 Converter

3.3 COMPUTER SOFTWARE

Programming a digital compensator with A/D and D/A capability as a discrete time system is straightforward. Consider the simulation of a digital control system such as shown in Figure 2.3.2. The analog parts of the system are simulated on the analog computer. For example, the digital compensator is characterized as:

$$\begin{bmatrix} X_1(k+1) \\ X_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} R(k) \\ Y(k) \end{bmatrix}$$

$$U(k) = [C_1 \quad C_2] \begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix}$$

A BASIC program for this system is outlined in the flow chart shown as Figure 3.3.1. The variables X1D and X2D are used for $X_1(k+1)$ and $X_2(k+1)$. The initial values of X_1, X_2, and R are first set to zero. Then an input value Y is read from the A/D converter. U(k) is calculated and the values of $X_1(k)$, $X_2(k)$ and Y(k) are updated for the next calculation cycle. U(k) is then output to the D/A converter. The program waits for a new input sample value, computes the next U(k) data, and so forth. Also, the program supports the function to check whether there is sufficient time between samples to perform the calculations or not.

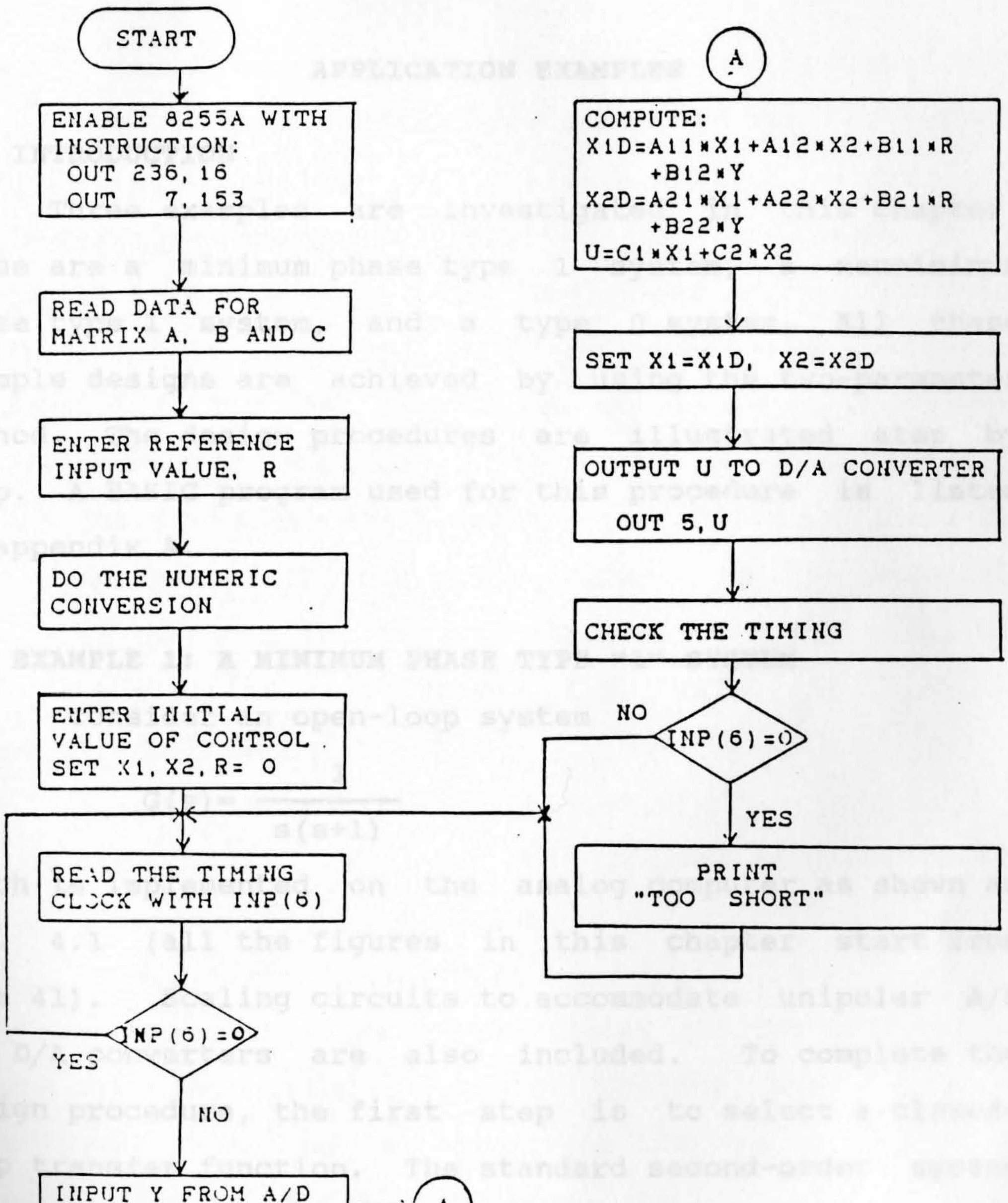

Fig. 3.3.1. Flow chart for BASIC program

CHAPTER IV

APPLICATION EXAMPLES

4.1 INTRODUCTION

Three examples are investigated in this chapter. These are a minimum phase type 1 system, a nonminimum phase type 1 system, and a type 0 system. All these example designs are achieved by using the two-parameter method. The design procedures are illustrated step by step. A BASIC program used for this procedure is listed in appendix A.

4.2 EXAMPLE 1: A MINIMUM PHASE TYPE "1" SYSTEM

Consider an open-loop system

$$G(s)=\frac{1}{s(s+1)}$$

which is implemented on the analog computer as shown as Fig. 4.1 (all the figures in this chapter start from page 41). Scaling circuits to accommodate unipolar A/D and D/A converters are also included. To complete the design procedure, the first step is to select a closed-loop transfer function. The standard second-order system described by the transfer function

$$T(s)=\frac{W_n^2}{s^2+2\zeta W_n s+W_n^2} \qquad (4.2.1)$$

is selected for the closed-loop transfer function. ζ is the damping ratio and W_n is the natural frequency. A large portion of classical design method is, in fact, based on the assumption that the closed-loop system can be made to behave in a way that closely matches that of the system function of Eq. 4.2.1 for a certain ζ and W_n. Using the developments of Chapter II, the system of Eq. 4.2.1 can be transferred into the z-plane to determine equivalent pole locations. To produce a good transient response, not too much damping and not too much overshoot, the value $\zeta=\sqrt{2}/2$ is chosen. It also helps the closed-loop system not to make the speed too much different from the open-loop system. In the other words, if the open-loop system is really slow, one can't expect to speed it up too much in the closed-loop system. Thus, value of $W_n=1$ is chosen. From the Eq. 2.3.8,

$$A(z)D(z)+M(z)N(z)=Dp(z)Dp(z)=F(z).$$

If the computation time in the program is close to one sampling period, then one choses

$$\deg A(z)= 1+ \deg M(z)$$

This means that there is a time delay in the compensator of one sampling period. Thus, all program steps can be done in one sampling period. Since the degree of D(z) is 2 and the degree of N(z) is equal to 1, the degree of A(z) will be 2. Also, the degree of $D_p(z)$ is 3 and the degree of $\bar{D}_p(z)$ is 1, which implies one more control pole and one cancelled pole have to be introduced. In the discrete case, it is possible to put the cancelled pole and even the additional control pole at the origin (z=0), which corresponds to s approaching to the minus infinity in the analog case.

The z transform of the open-loop plant is preceded by the Zero-Order-Hold equivalent with 0.2 second sampling period.

For $G(s)=\dfrac{1}{s(s+1)}$

$$G(z)=\frac{0.01873(z+0.935525)}{(z-1)(z-0.8187308)}=\frac{\bar{b}_1 Z+\bar{b}_0}{Z^2+\bar{a}_1 Z+\bar{a}_0}$$

The closed-loop control poles are $s^2+\sqrt{2}s+1$ and $s=-\infty$.

The corresponding z-transform is

$$G_o(z)=\frac{k(z+0.9355255)}{z(z^2-1.718945z+0.07536)}$$

For a DC gain of $G_o(1)=1$, the value of k is 0.0179085.

Thus, $$G_o(z)=\frac{0.017985(z+0.935525)}{z(z^2-1.718945z+0.7536)}$$

$$=\frac{P_1 z+P_0}{z^3+C_2 z^2+C_1 z+C_0}$$

The design procedure as follows:

Step 1: Choose the desired $G_o(z)$.

Step 2: Compute the following:

$$\frac{G_o(z)}{N(z)}=\frac{N_o(z)}{D_o(z)N(z)}=\frac{N_p(z)}{D_p(z)}=\frac{0.173605}{z^3+C_2 z^2+C_1 z+C_0}$$

Step 3: Introduce a polynomial $D_p(z)=z+d_0$ which will be cancelled in the design procedure. In case 1, do=0.

Step 4: $L(z)= N_p(z)\ \bar{D}_p(z) = L_1 z+ L_2$

$$A(z)\cdot D(z)+M(z)\cdot N(z)=D_p(z)\cdot \bar{D}_p(z)$$
$$=z^4+\beta_4 z^3+\beta_3 z_2+\beta_2 z+\beta_1$$

Assume $A(z)= z^2+A_1 z+A_0$, $M(z)=M_1 z+M_0$

$$(z^2+A_1 z+A_0)(z^2+\bar{a}_1 z+\bar{a}_0)+(M_1 z+M_0)(\bar{b}_1 z+\bar{b}_0)$$
$$= z^4+\beta_4 z^3+\beta_3 z^2+\beta_2 z+\beta_1$$

by matching the coefficients:

$$\begin{bmatrix} \bar{a}_0 & \bar{b}_0 & 0 & 0 \\ \bar{a}_1 & \bar{b}_1 & \bar{a}_0 & \bar{b}_0 \\ 1 & 0 & \bar{a}_1 & \bar{b}_1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} A_0 \\ M_0 \\ A_1 \\ M_1 \end{bmatrix} = \begin{bmatrix} \beta 1 \\ \beta 2 \\ \beta 3-\bar{a}_0 \\ \beta 4-\bar{a}_1 \end{bmatrix}$$

where $\bar{a}_0=0.8187308, \bar{a}_1=-1.8187308, \bar{b}_0=0.0175231$
$\bar{b}_1=0.01873,\ \beta 1=0, \beta 2=0, \beta 3=0.7536705, \beta 4=-1.718945$

The solution is

$A_0=0.05252052$ $M_0=-2.453919$

$A_1=0.09978592$ $M_1=3.411789$

Then,

$$C(z)=[C_1(z) \quad C_2(z)]=[\frac{L(z)}{A(z)} \quad \frac{-M(z)}{A(z)}]$$

$$C(z)=[\ \frac{L_1 z+L_0}{z^2+A_1 z+A_0} \quad - \frac{M_1 z+M_0}{z^2+A_1 z+A_0}\ ]$$

The state-variable form for the above compensator equation is:

$$X(k+1) = \begin{bmatrix} 0 & -A_0 \\ 1 & -A_1 \end{bmatrix} \begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix} + \begin{bmatrix} L_0 & -M_0 \\ L_1 & -M_1 \end{bmatrix} \begin{bmatrix} R(k) \\ Y(k) \end{bmatrix}$$

$$U(k) = [0 \quad 1] \begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix} = X_2(k)$$

The above equation is implemented on a Radio Shack model 4 computer. The BASIC program is shown in Appendix B. Comparing the results of the CC [14] simulation (Fig. 4.4) with the result of control system (Fig. 4.6), the difference in the input signals u(t) is the noise from the A/D converter in the actual control system. Fig. 4.22 and Fig. 4.23 show the noise-to-input signal and the noise-to-output frequency response. To get a better response, Case 2 and Case 3 illustrate pole assignments that reduce the noise. All the results are shown as Table 1. From the Fig. 4.6 to Fig. 4.21, one can easily see how the pole assignment affects the noise reduction.

TABLE 1
THE RESULTS OF 3 CASES IN EXAMPLE 1

	CASE 1	CASE 2	CASE 3
CONTROL POLES	$1/s(s^2+\sqrt{2}s+1)$	$1/(s+5)(s^2+\sqrt{2}s+1)$	$1/(s+1)(s^2+\sqrt{2}s+1)$
CANCELLED POLE	$s=-\infty$	$1/(s+5)$	$1/(s+1)$
G(z)	$\frac{0.01873(z+0.93552)}{(z-1)(z-0.8187308)}$	Same as Case 1	Same as Case 1
Go(z)	$\frac{0.01791(z+0.93552)}{z(z^2-1.7189z+0.0754)}$	$\frac{0.0113(z+0.9355)}{(z-0.367)(z^2-1.719z+0.753)}$	$\frac{0.00325(z+0.9355)}{(z-0.848)(z^2-1.7z+0.7536)}$
C1(z)	$\frac{0.9561398}{z^2+0.0997z+0.052}$	$\frac{0.604(z-0.3679)}{z^2-0.6305z+0.1494}$	$\frac{0.173605(z-0.8187)}{z^2-1.537z+0.62}$
C2(z)	$\frac{3.41178z-2.453919}{z^2+0.09978z+0.05252}$	$\frac{1.5403z-1.1609}{z^2-0.63059z+0.149427}$	$\frac{0.17356z-0.142143}{z^2-1.53764z+0.6201}$

	CASE 1	CASE 2	CASE3
C.C SIMULATION RESULTS FOR CONTROL SIGNAL U	Fig. 4.4	Fig. 4.10	Fig. 4.16
C.C SIMULATION RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 4.5	Fig. 4.11	Fig. 4.17
PMIMELINE PLOTTER RESULTS FOR CONTROL SIGNAL U	Fig. 4.6 Fig. 4.8	Fig. 4.12 Fig. 4.14	Fig. 4.18 Fig. 4.20
PMIMELINE PLOTTER RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 4.7 Fig. 4.9	Fig. 4.13 Fig. 4.15	Fig. 4.19 Fig. 4.21
NOISE TO OUTPUT FREQUENCY RESPONSE	Fig. 4.23	Fig. 4.23	Fig. 4.23
NOISE TO INPUT FREQUENCY RESPONSE	Fig. 4.22	Fig. 4.22	Fig. 4.22

4.3 EXAMPLE 2: A NONMINIMUM PHASE SYSTEM

Consider an open-loop plant

$$G(s)=(s-1)/s/(s+2)$$

which is implemented on the analog computer as shown in Fig. 4.2. The design procedure follows that of the Example 1, and the results are shown in Table 2.

4.4 EXAMPLE 3: A TYPE 0 SYSTEM

Consider an open-loop plant:

$$G(s)= 1/(s^2+0.5s+1)$$

which is implemented on the analog computer, as shown in Fig. 4.3. A type 0 plant can be made to act as a type 1 system, for type 1 system performance is required to track a constant reference input with zero error. Type 1 behavior can be achieved by requiring that the polynomial A(z) have the form $A(z)=(z-1)A'(z)$ [6]. Thus, equation (2.3.8) can be written as

$$(z-1)\ A'(z)\cdot D(z)+M(z)\cdot N(z)=Dp(z)\cdot \overline{D}p(z)=F(z)$$

If one follows the procedure of the two previous examples, the result will be four linear algebraic equations in three unknowns. For a solution to exist in general, there must be at least as many unknowns as equations. Thus,

deg $A'(z)=2$, deg $M(z)=2$, deg $Dp(z)=3$ and deg $\overline{D}p(z)=2$

It means one more cancelled pole and one more control pole should be chosen.

TABLE 2

THE RESULTS OF 3 CASES IN EXAMPLE 2

	CASE 1	CASE 2	CASE 3
CONTROL POLES	$1/s(s^2+\sqrt{2}s+1)$	$1/(s+5)(s^2+\sqrt{2}s+1)$	$1/(s+1)(s^2+\sqrt{2}s+1)$
CANCELLED POLE	$s=-\infty$	$1/(s+5)$	$1/(s+1)$
G(z)	$\frac{0.14726(z-1.22387)}{(z-1)(z-0.6703208)}$	Same as Case 1	Same as Case 1
Go(z)	$\frac{-0.15511(z-1.22387)}{z(z^2-1.7189z+0.0754)}$	$\frac{-0.0978(z-1.2238)}{(z-0.367)(z^2-1.719z+0.753)}$	$\frac{-0.0282(z-1.2238)}{(z-0.818)(z^2-1.7z+0.7536)}$
C1(z)	$\frac{-1.053306*z}{z^2-0.0048z+1.2906}$	$\frac{-0.664(z-0.3679)}{z^2-0.7842z+0.3354}$	$\frac{-0.191(z-0.8187)}{z^2-1.686z+0.76}$
C2(z)	$\frac{-2.34258z+1.2908}{z^2-0.0048z+1.2906}$	$\frac{1.5403z-1.1609}{z^2-0.7842z+0.3354}$	$\frac{0.17356z-0.142143}{z^2-1.686z+0.76}$

	CASE 1	CASE 2	CASE3
C.C SIMULATION RESULTS FOR CONTROL SIGNAL U	Fig. 4.24	Fig. 4.30	Fig. 4.36
C.C SIMULATION RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 4.25	Fig. 4.31	Fig. 4.37
PMIMELINE PLOTTER RESULTS FOR CONTROL SIGNAL U	Fig. 4.26 Fig. 4.28	Fig. 4.32 Fig. 4.34	Fig. 4.38 Fig. 4.40
PMIMELINE PLOTTER RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 4.27 Fig. 4.29	Fig. 4.33 Fig. 4.35	Fig. 4.39 Fig. 4.41
NOISE TO OUTPUT FREQUENCY RESPONSE	Fig. 4.43	Fig. 4.43	Fig. 4.43
NOISE TO INPUT FREQUENCY RESPONSE	Fig. 4.42	Fig. 4.42	Fig. 4.42

Set $A(z)=(z-1)\cdot(z^2+A_1*z+A_0)$,

$M(z)=M_2z^2+M_1z+M_0$,

$D(z)=z^2+a_1z+a_0$,

$N(z)=b_1z+b_0$,

and $F(z)=Z^5+\beta_5Z^4+\beta_4Z^3+\beta_3Z^2+\beta_2Z+\beta_1$.

By matching the coefficients, A(z) and M(z) can be found from the following matrix :

$$\begin{bmatrix} -\bar{a}_0 & \bar{b}_0 & 0 & 0 & 0 \\ \bar{a}_0-\bar{a}_1 & \bar{b}_1 & -\bar{a}_0 & \bar{b}_0 & 0 \\ \bar{a}_1-1 & 0 & \bar{a}_0-\bar{a}_1 & \bar{b}_1 & \bar{b}_0 \\ 1 & 0 & \bar{a}_1-1 & 0 & b_1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} A_0 \\ M_0 \\ A_1 \\ M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3+\bar{a}_0 \\ \beta_4-\bar{a}_0+a_1 \\ \beta_5-\bar{a}_1+1 \end{bmatrix}$$

Results from four different cases are shown in Table 3.

TABLE 3

THE RESULTS OF 4 CASES IN EXAMPLE 3

	CASE 1	CASE 2
CONTROL POLES	$1/(s+5)(s^2+\sqrt{2}s+1)$	$1/(s+1)(s^2+\sqrt{2}s+1)$
CANCELLED POLES	$1/(s+5)^2$	$s=-\infty$ $s=-\infty$
G(z)	$\frac{0.01928(z+0.96717)}{z^2-1.866899z+0.9048374}$	$\frac{0.01928(z+0.96717)}{z^2-1.866899z+0.9048374}$
Go(z)	$\frac{0.011138(z+0.96717)}{(z-0.36788)(z^2-1.7189z+0.0754)}$	$\frac{0.0031998(z+0.96717)}{(z-0.81873)(z^2-1.7189z+0.7536)}$
C1(z)	$\frac{0.5775z^2-0.4249z+0.078161}{z^3-0.9556z^2+0.1751z-0.21941}$	$\frac{0.1659*z^2}{z^3-0.6707z^2-0.4949z+0.16367}$
C2(z)	$\frac{9.9869z^2-18.388z+8.632319}{z^3-0.9556z^2+0.1751z-0.21941}$	$\frac{8.78868z^2-16.5625z+7.940732}{z^3-0.6707z^2-0.4949z+0.16367}$
	CASE 1	**CASE 2**
C.C SIMULATION RESULTS FOR CONTROL SIGNAL U	Fig. 44	Fig. 50
C.C SIMULATION RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 45	Fig. 51
PMIMELINE PLOTTER RESULTS FOR CONTROL SIGNAL U	Fig. 46 Fig. 48	Fig. 55 Fig. 54
PMIMELINE PLOTTER RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 47 Fig. 49	Fig. 53 Fig. 55
NOISE TO OUTPUT FREQUENCY RESPONSE	Fig. 69	Fig. 69
NOISE TO INPUT FREQUENCY RESPONSE	Fig. 68	Fig. 68

TABLE 3 (CONT.)

THE RESULTS OF 4 CASES IN EXAMPLE 3

	CASE 3	CASE 4
CONTROL POLES	$1/(s+1)(s^2+\sqrt{2}s+1)$	$1/(s+1)(s^2+\sqrt{2}s+1)$
CANCELLED POLES	$1/(s+5)^2$	$1/(s+1)^2$
G(z)	$\frac{0.01928(z+0.96717)}{z^2-1.866899z+0.9048374}$	$\frac{0.01928(z+0.96717)}{z^2-1.866899z+0.9048374}$
Go(z)	$\frac{0.0031998(z+0.96715)}{(z-0.81873)(z^2-1.7189z+0.7536)}$	$\frac{0.0031998(z+0.96715)}{(z-0.81873)(z^2-1.7189z+0.7536)}$
C1(z)	$\frac{0.1659*(z-0.367879)^2}{z^3-1.4065z^2+0.56541z-0.1589}$	$\frac{0.1659(z^2-1.63746z+0.67032)}{z^3-2.3082z^2+1.7693z-0.46111}$
C2(z)	$\frac{3.4917z^2-6.6574z+3.231989}{z^3-1.4065z^2+0.56541z-0.1589}$	$\frac{0.16771z^2-0.35674z+0.19945}{z^3-2.3082z^2+1.7693z-0.46111}$

	CASE 3	CASE 4
C.C SIMULATION RESULTS FOR CONTROL SIGNAL U	Fig. 56	Fig. 62
C.C SIMULATION RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 57	Fig. 63
PMIMELINE PLOTTER RESULTS FOR CONTROL SIGNAL U	Fig. 58 Fig. 60	Fig. 64 Fig. 66
PMIMELINE PLOTTER RESULTS FOR OUTPUT STEP RESPONSE Y	Fig. 59 Fig. 61	Fig. 65 Fig. 67
NOISE TO OUTPUT FREQUENCY RESPONSE	Fig. 69	Fig. 69
NOISE TO INPUT FREQUENCY RESPONSE	Fig. 68	Fig. 68

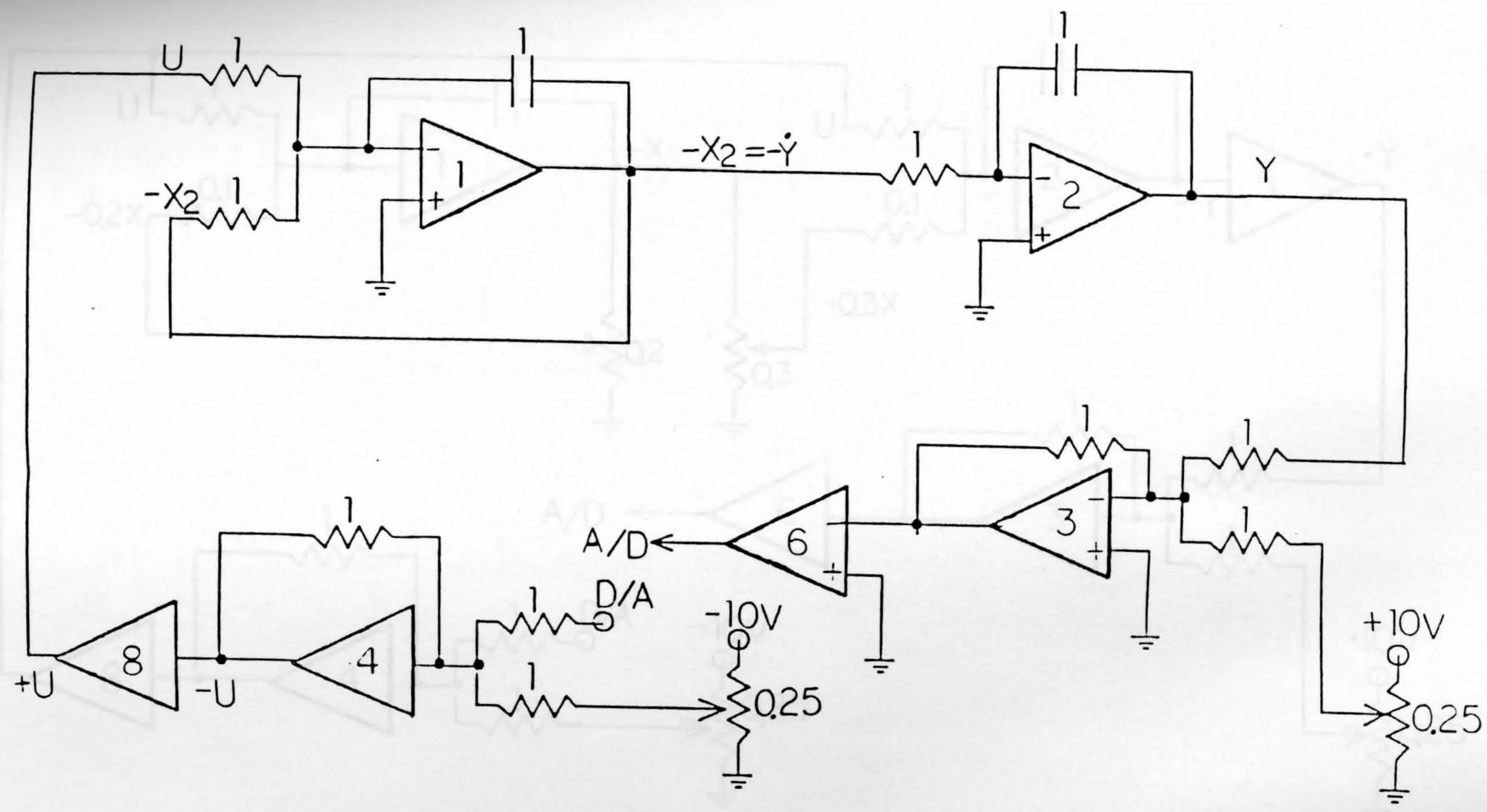

Figure 4.1 Analog plant circuit for Example 1

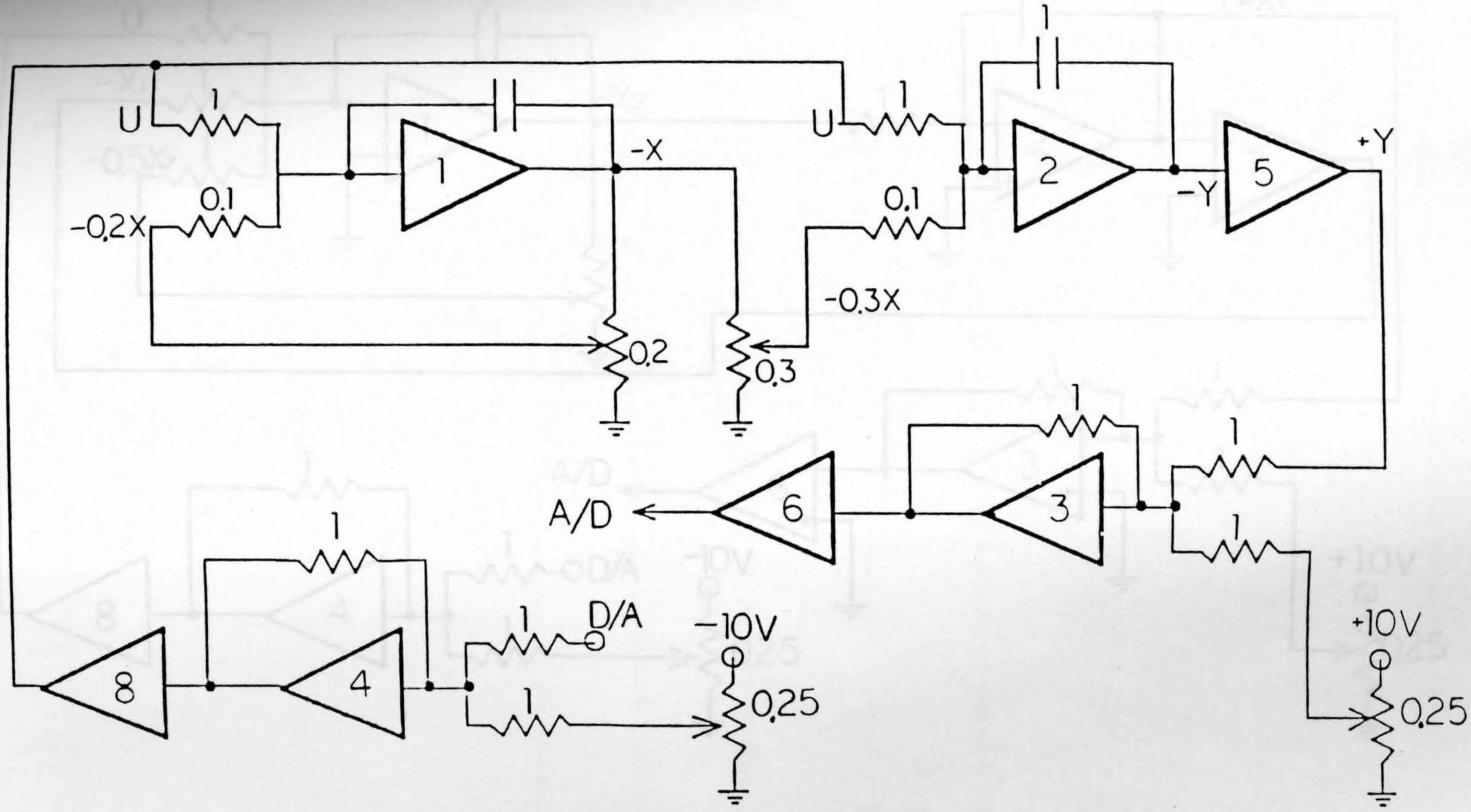

Figure 4.2 Analog plant circuit for Example 2

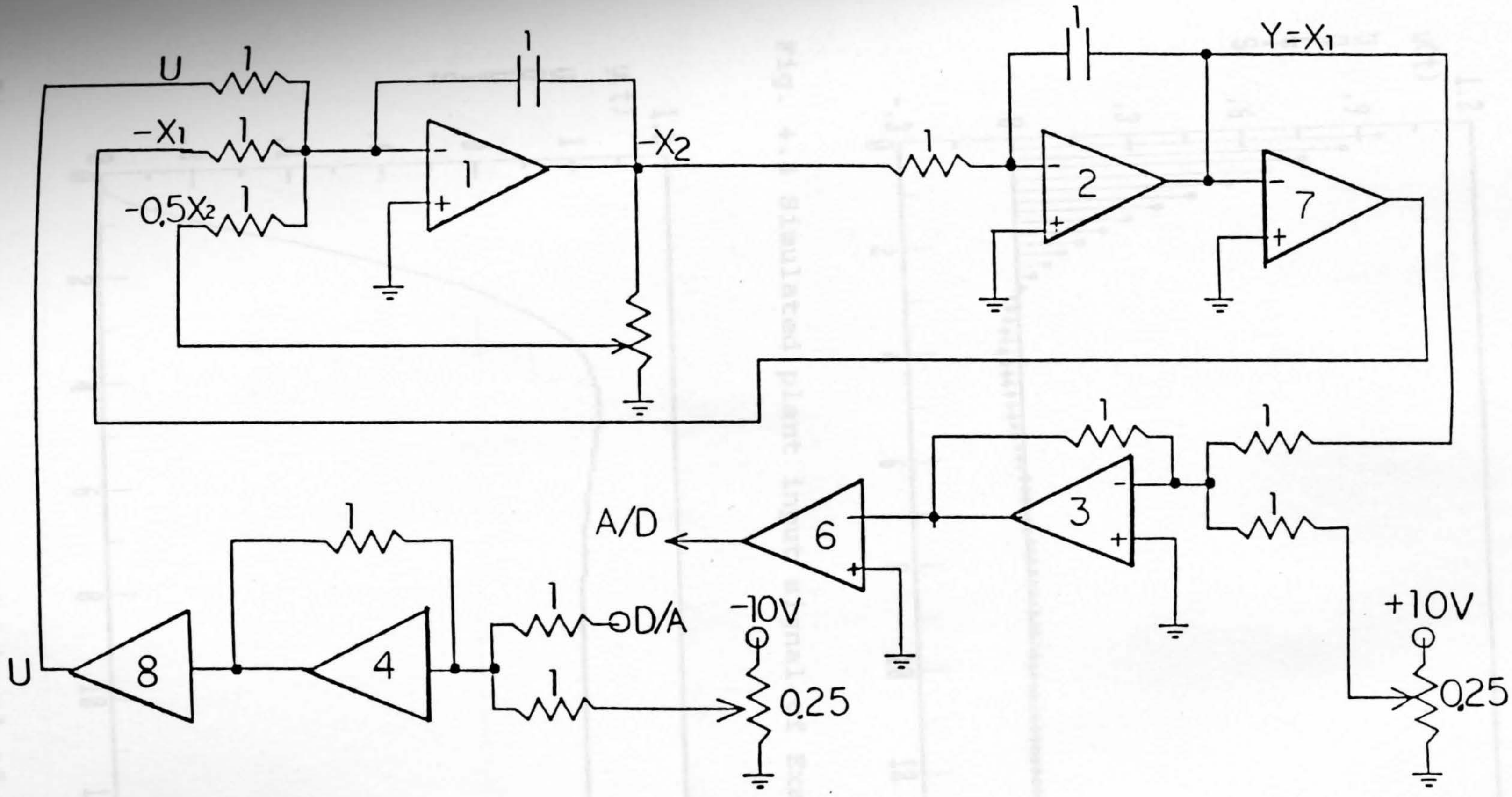

Figure 4.3 Analog plant circuit for Example 3

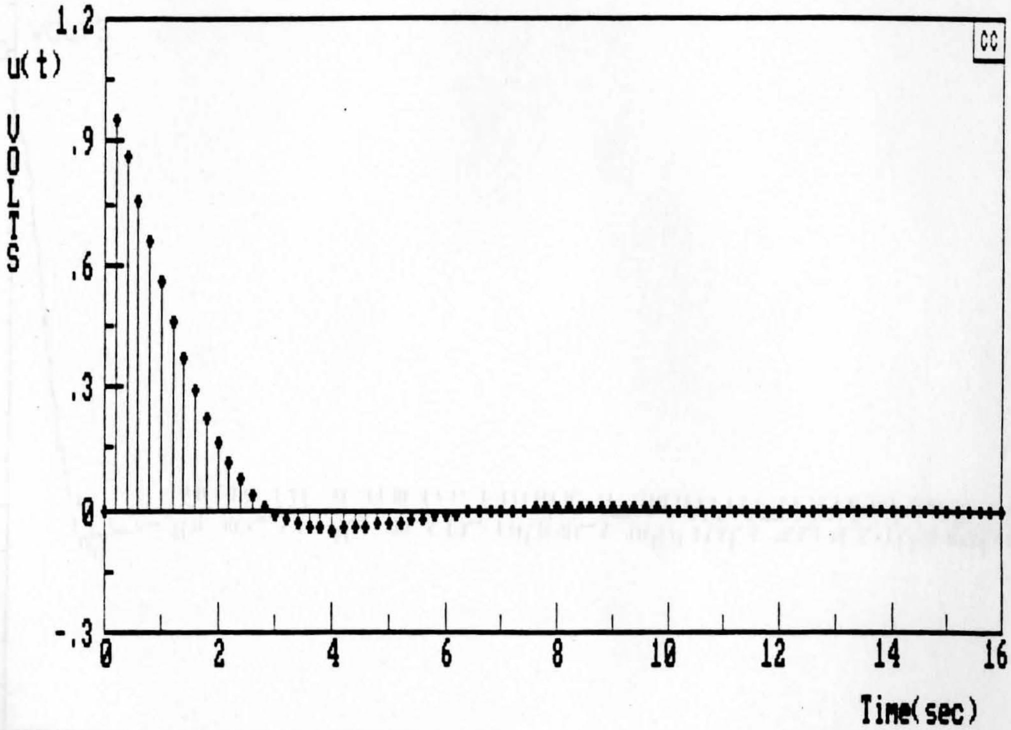

Fig. 4.4 Simulated plant input signal of Example 1 Case 1

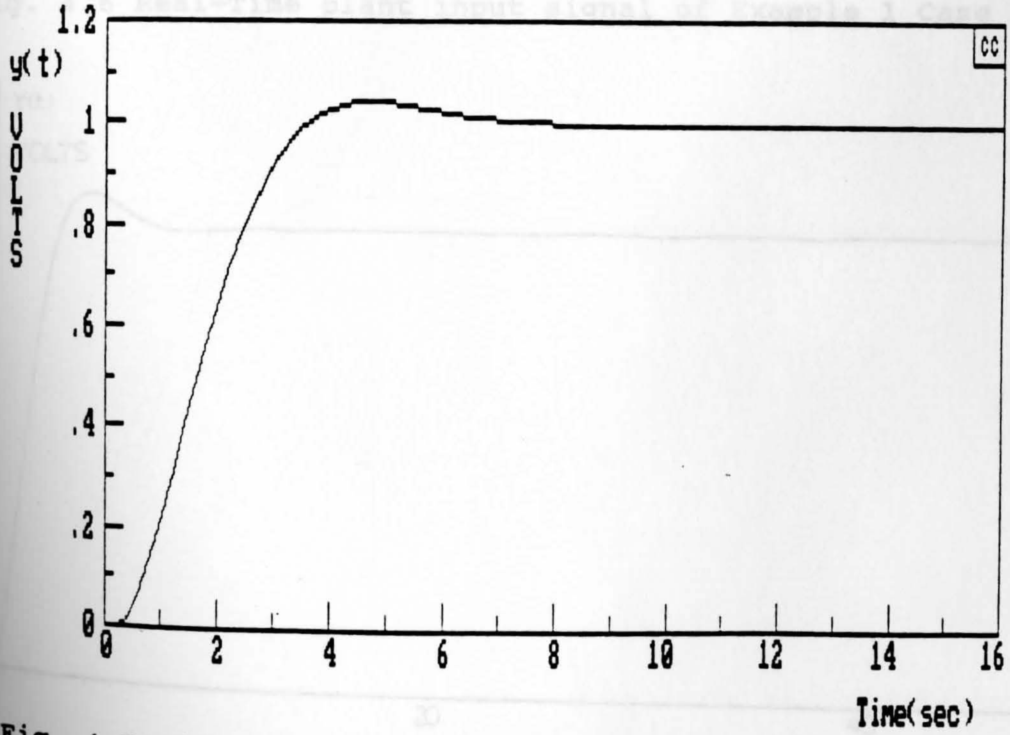

Fig. 4.5 Simulated plant output signal of Example 1 Case 1

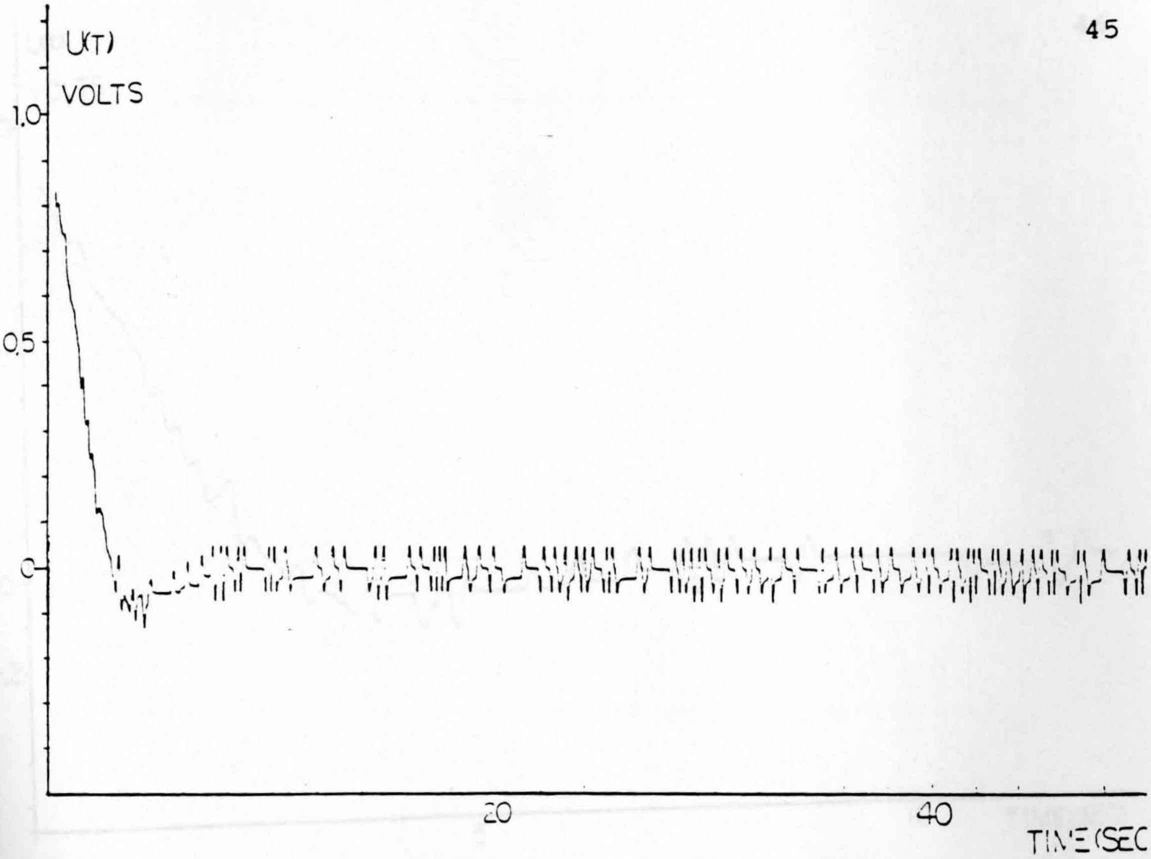

Fig. 4.6 Real-Time plant input signal of Example 1 Case 1

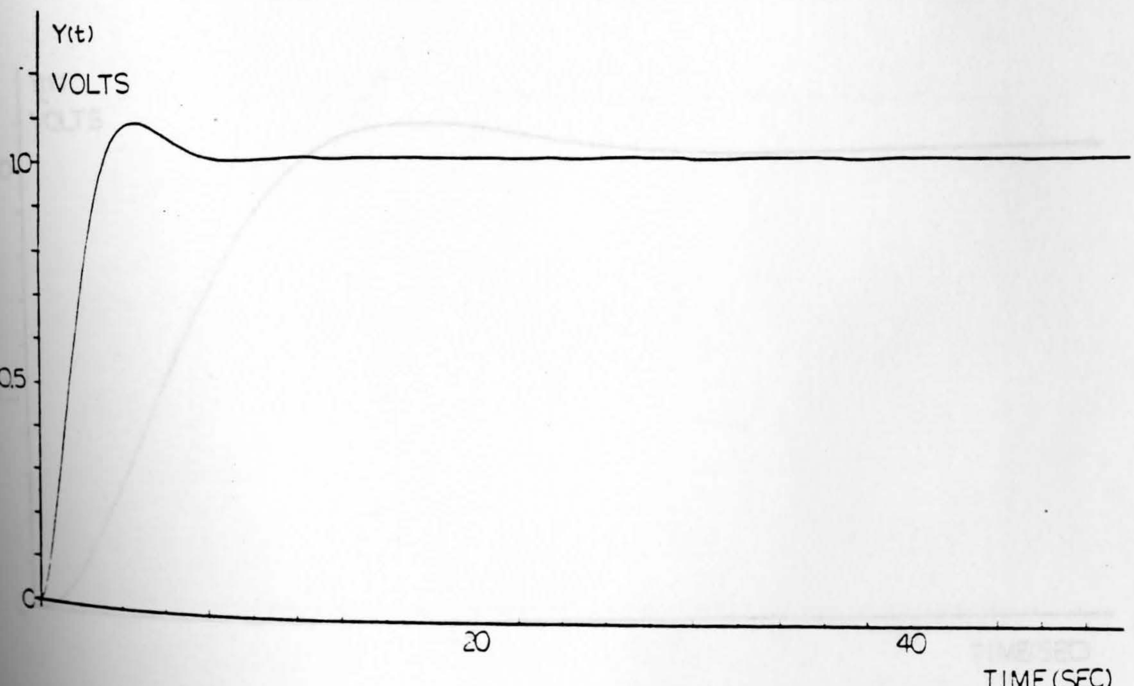

Fig. 4.7 Real-Time plant output signal of Example 1 Case 1

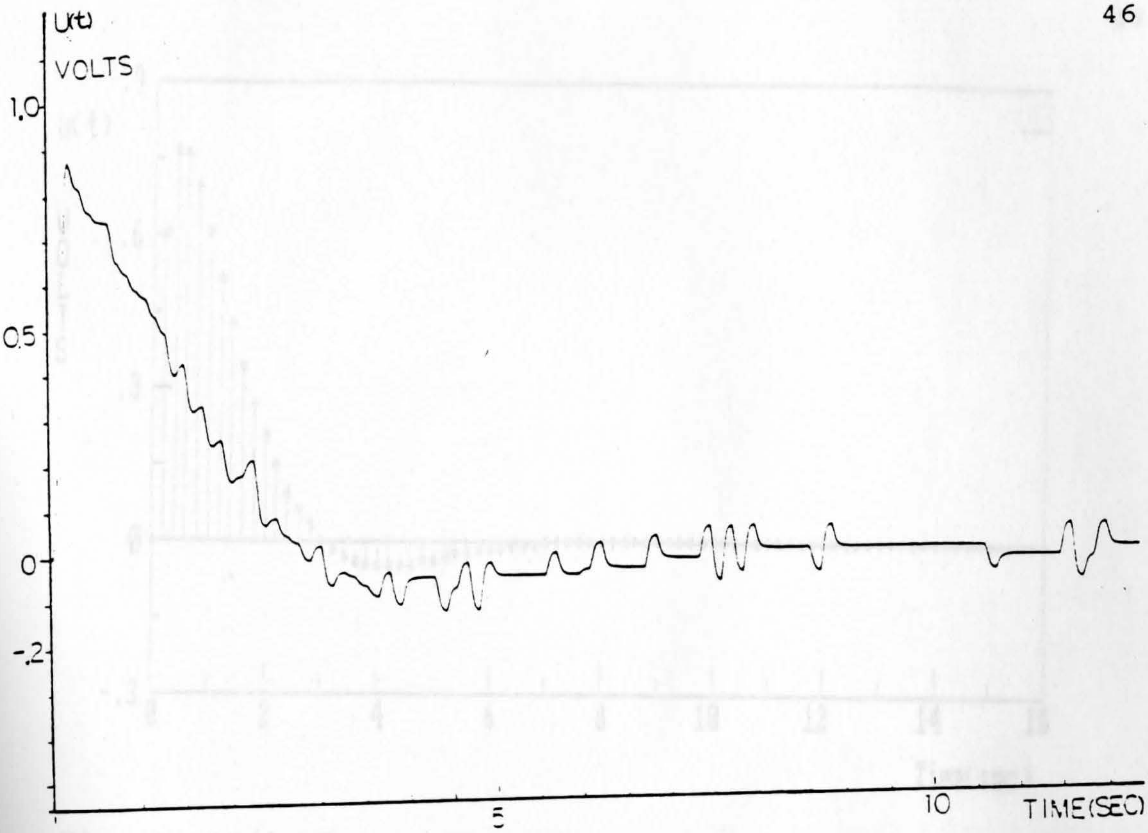

Fig. 4.8 Real-time plant input signal of Example 1 Case 1

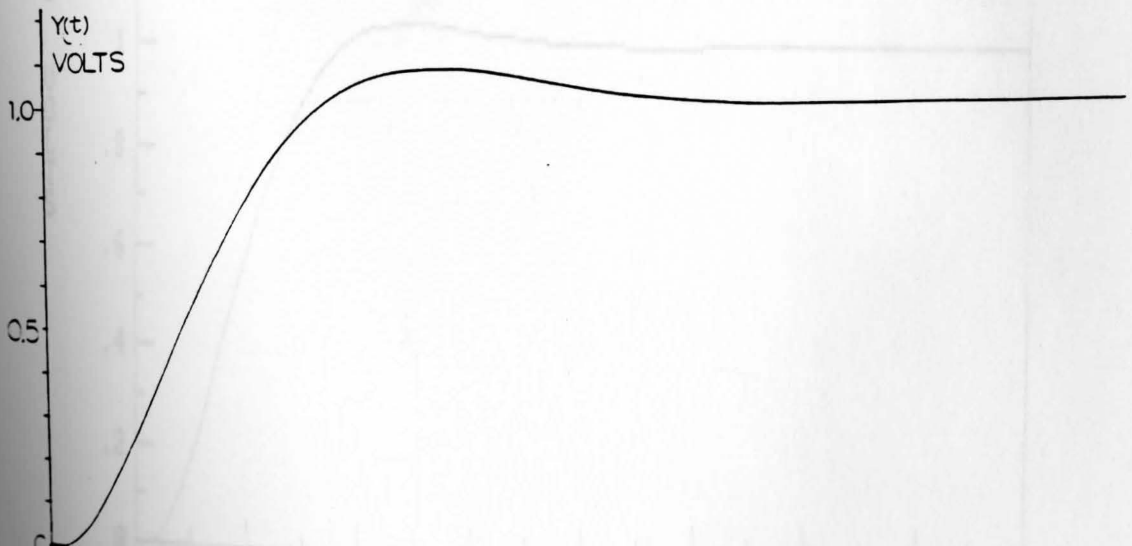

Fig. 4.9 Real-time plant output signal of Example 1 Case 1

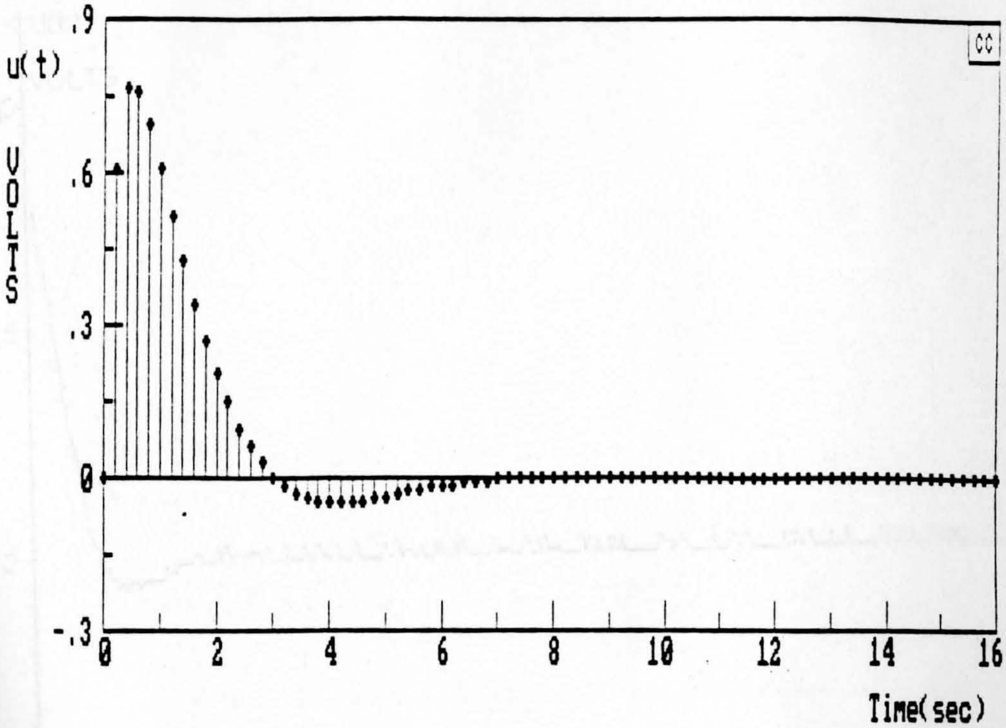

Fig. 4.10 Simulated plant input signal of Example 1 Case 2

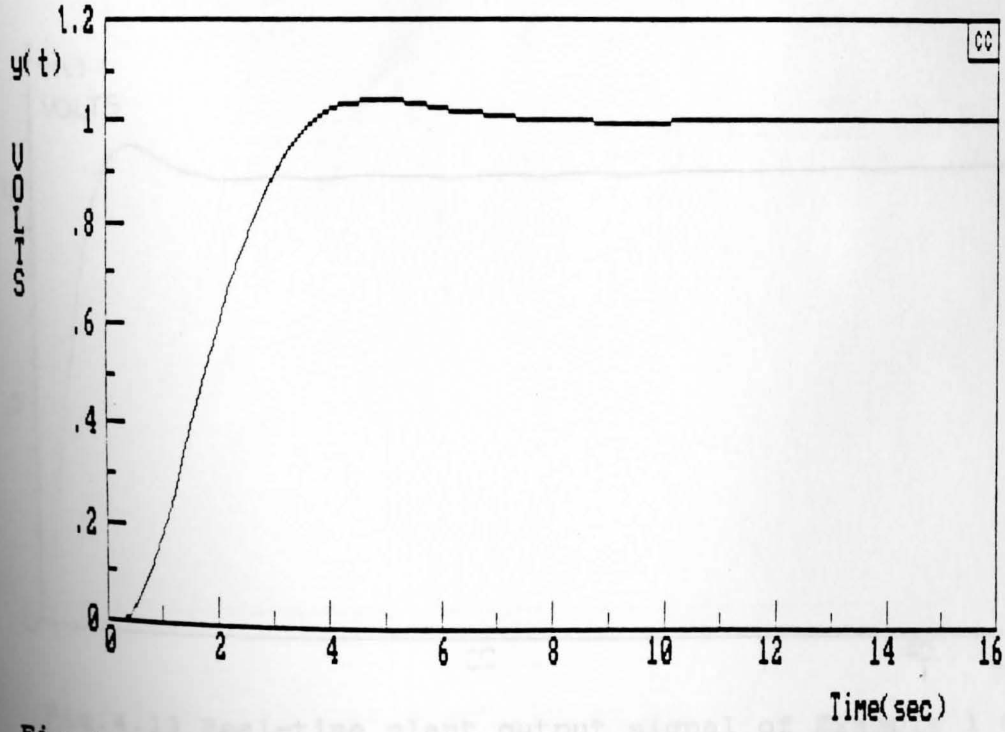

Fig.4.11 Simulated plant output signal of Example 1 Case 2

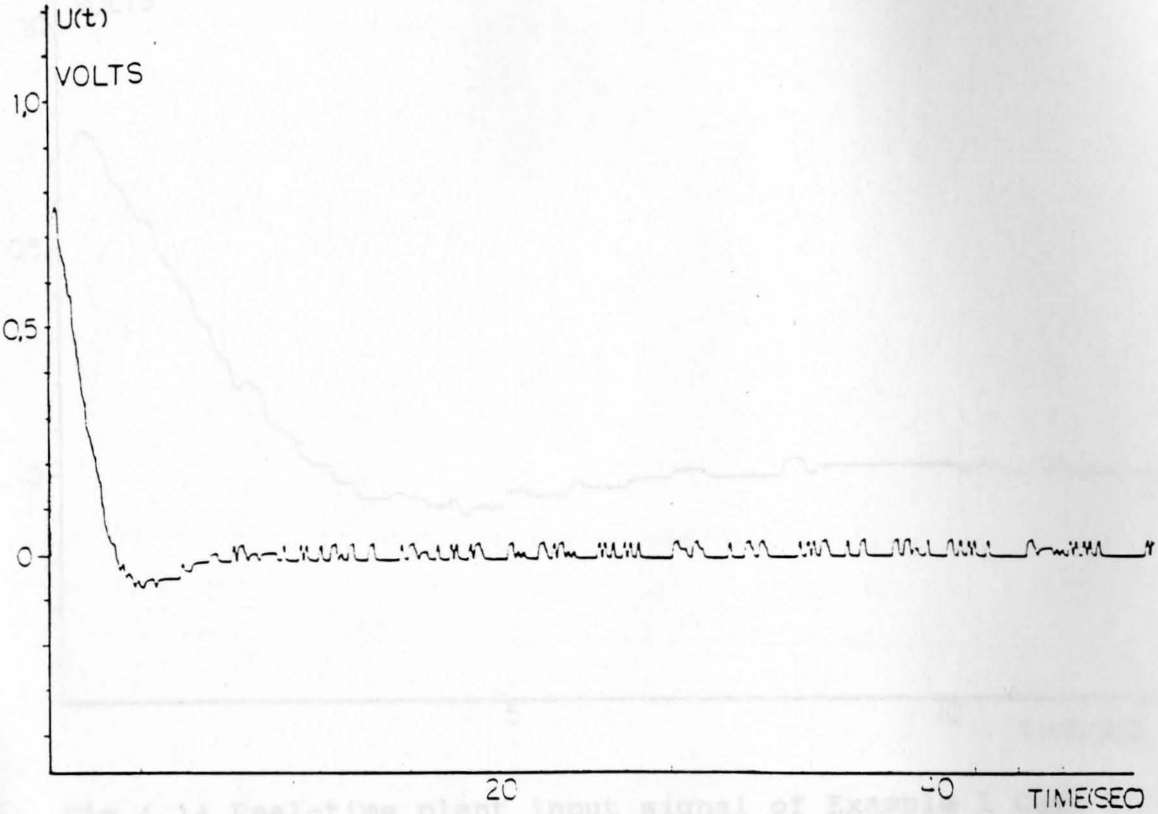

Fig.4.12 Real-time plant input signal of Example 1 Case 2

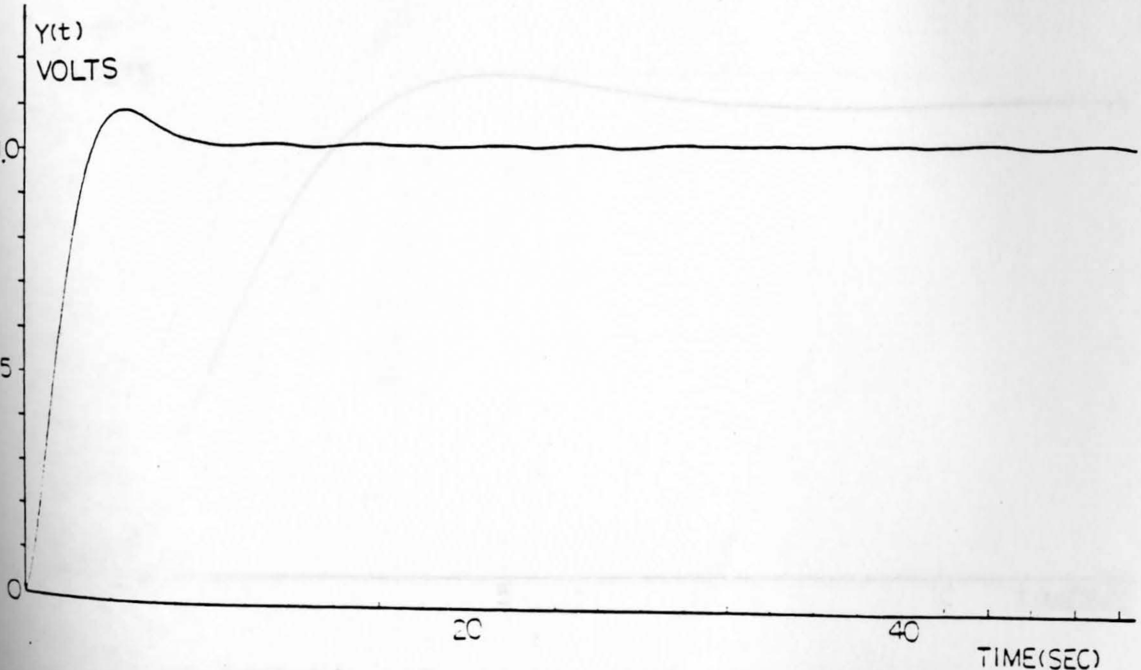

Fig.4.13 Real-time plant output signal of Example 1 Case 2

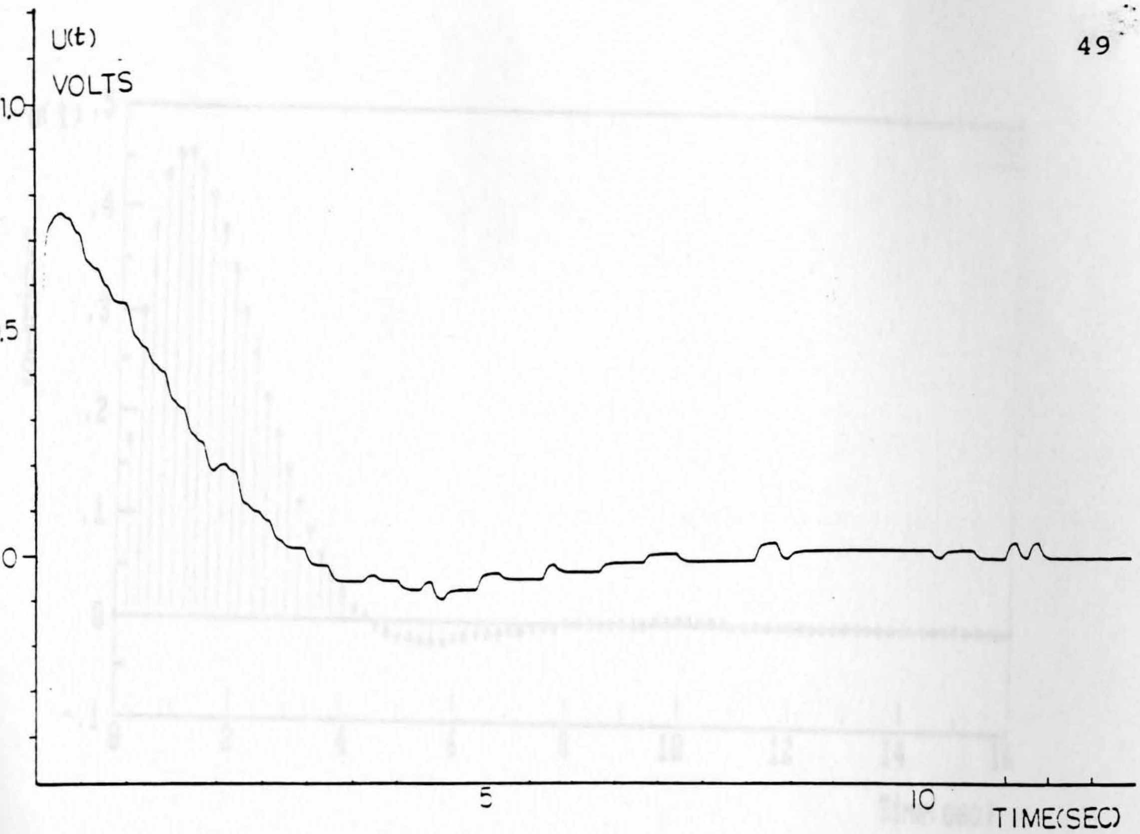

Fig.4.14 Real-time plant input signal of Example 1 Case 2

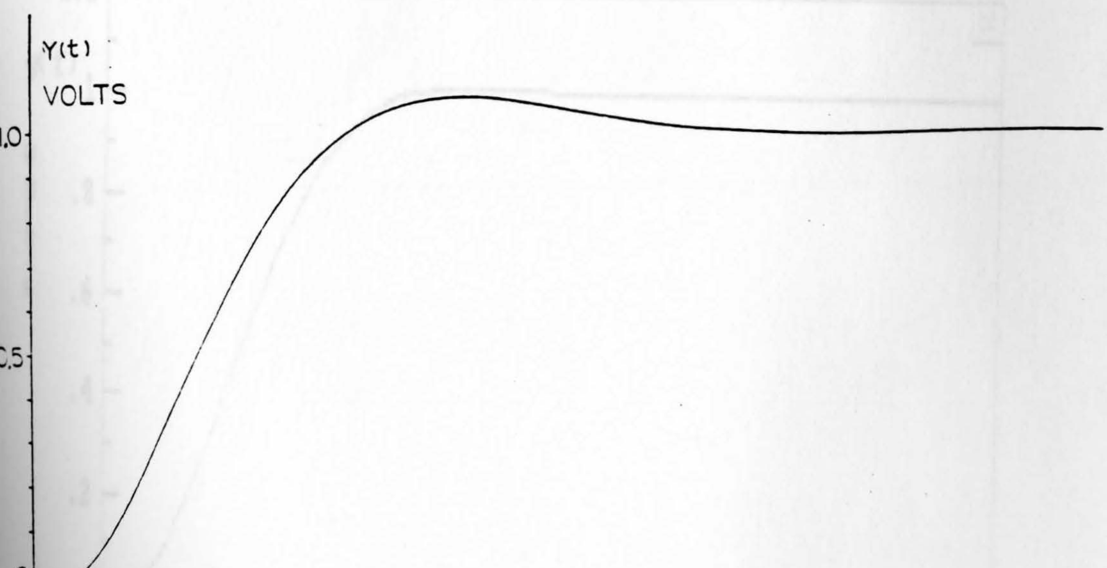

Fig.4.15 Real-time plant output signal of Example 1 Case 2

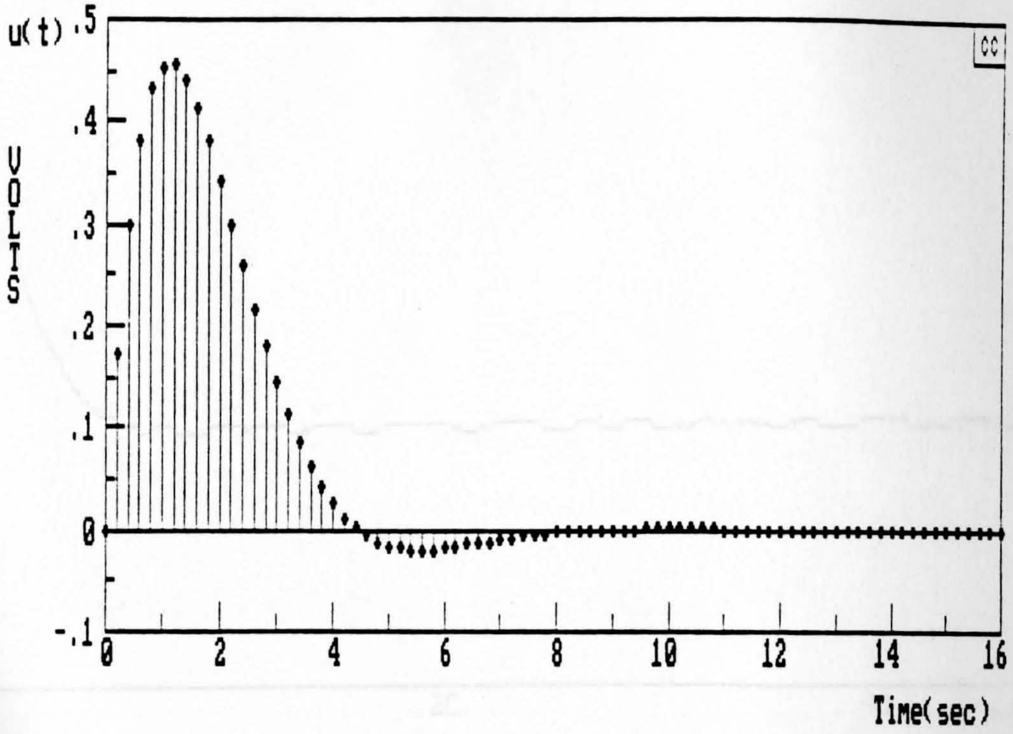

Fig.4.16 Simulated plant input signal of Example 1 Case 3

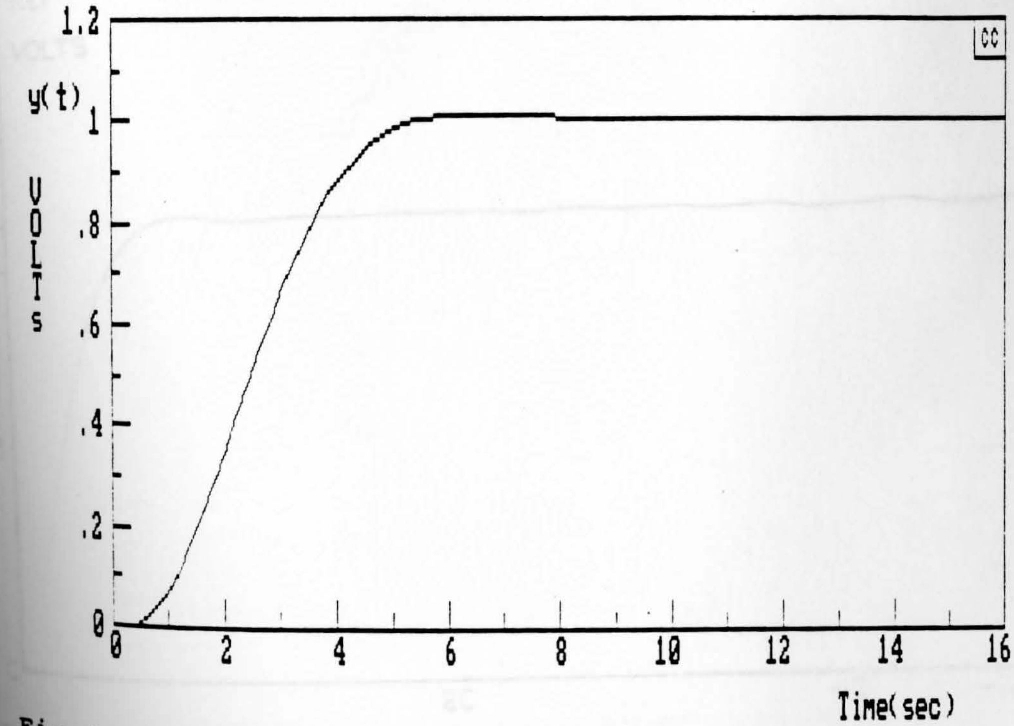

Fig.4.17 Simulated plant output signal of Example 1 Case 3

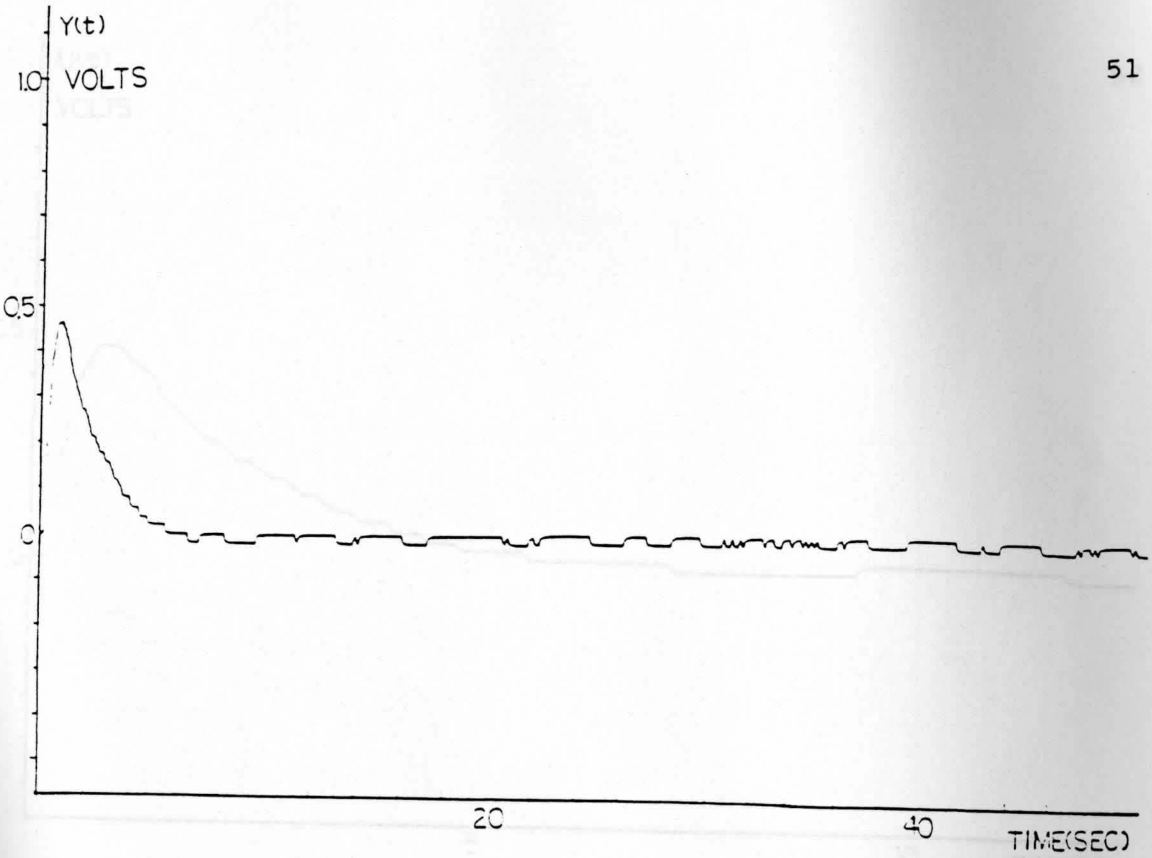

Fig.4.18 Real-time plant input signal of Example 1 Case 3

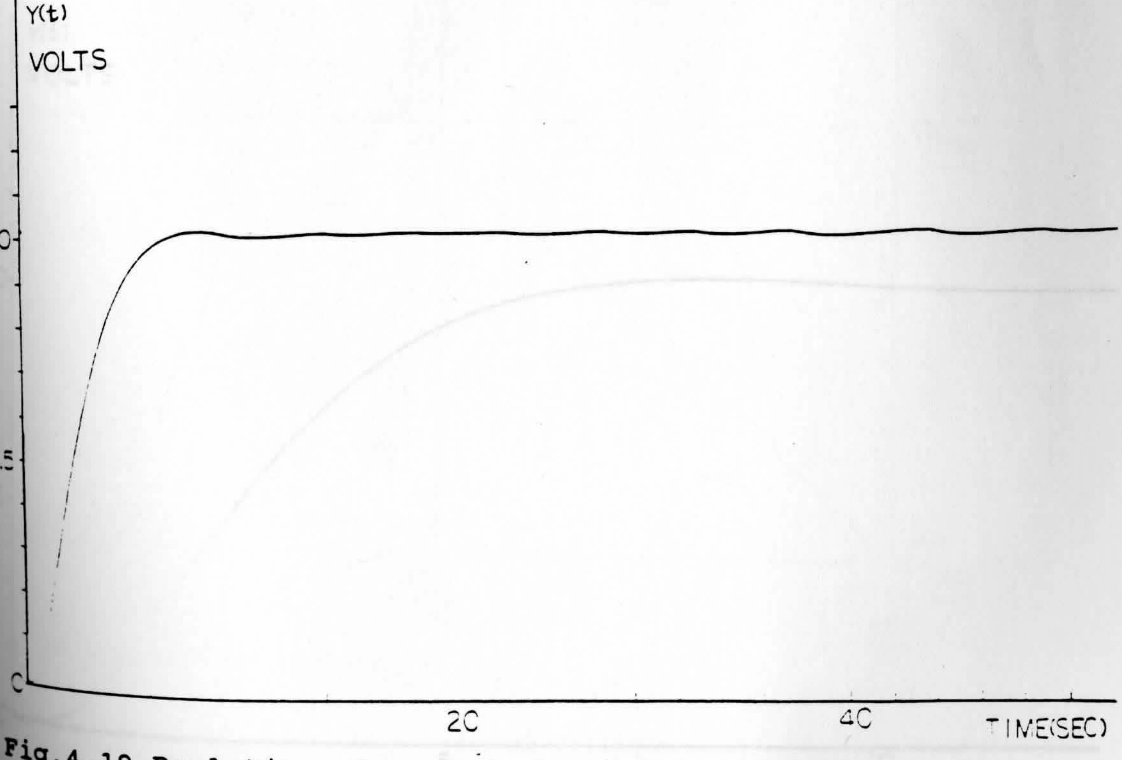

Fig.4.19 Real-time plant output signal of Example 1 Case 3

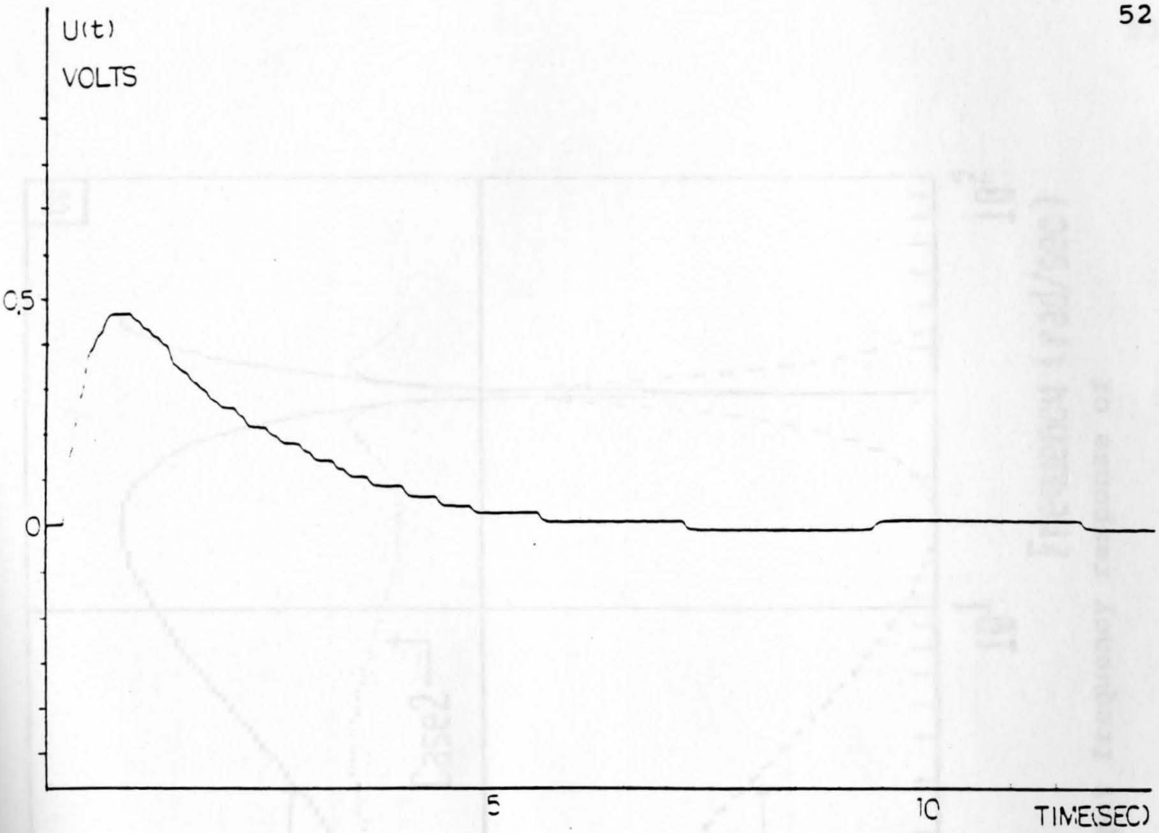

Fig.4.20 Real-time plant input signal of Example 1 Case 3

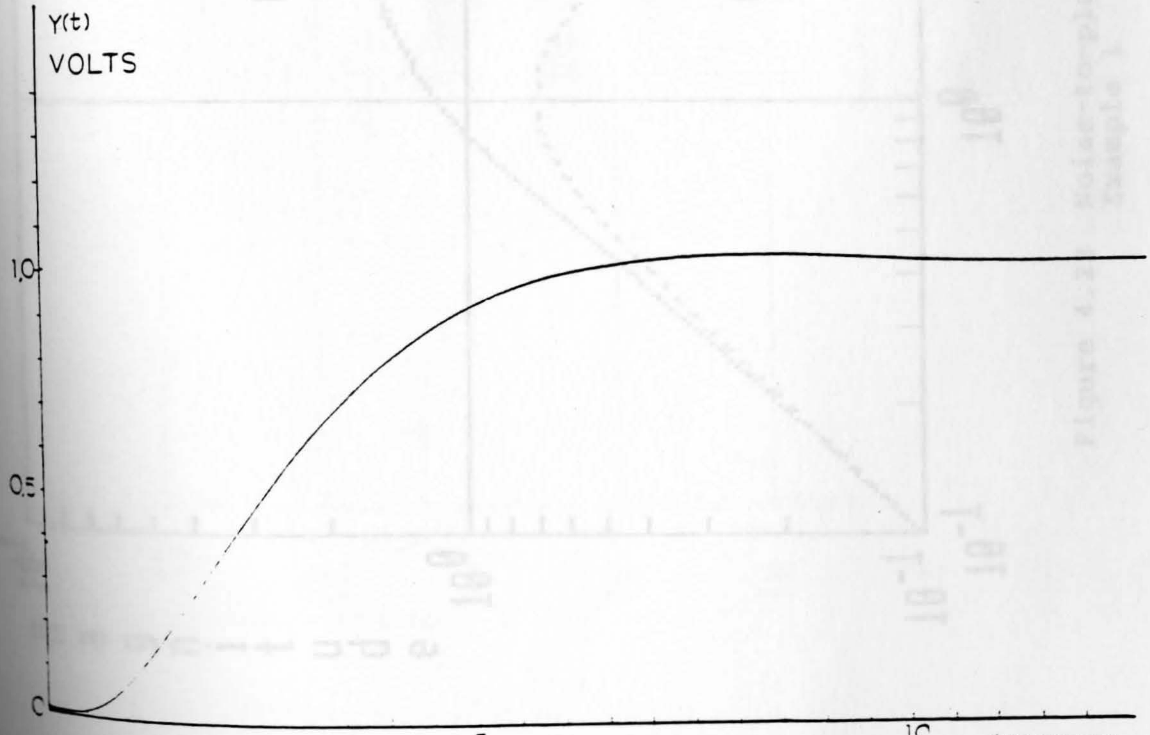

Fig.4.21 Real-time plant output signal of Example 1 Case 3

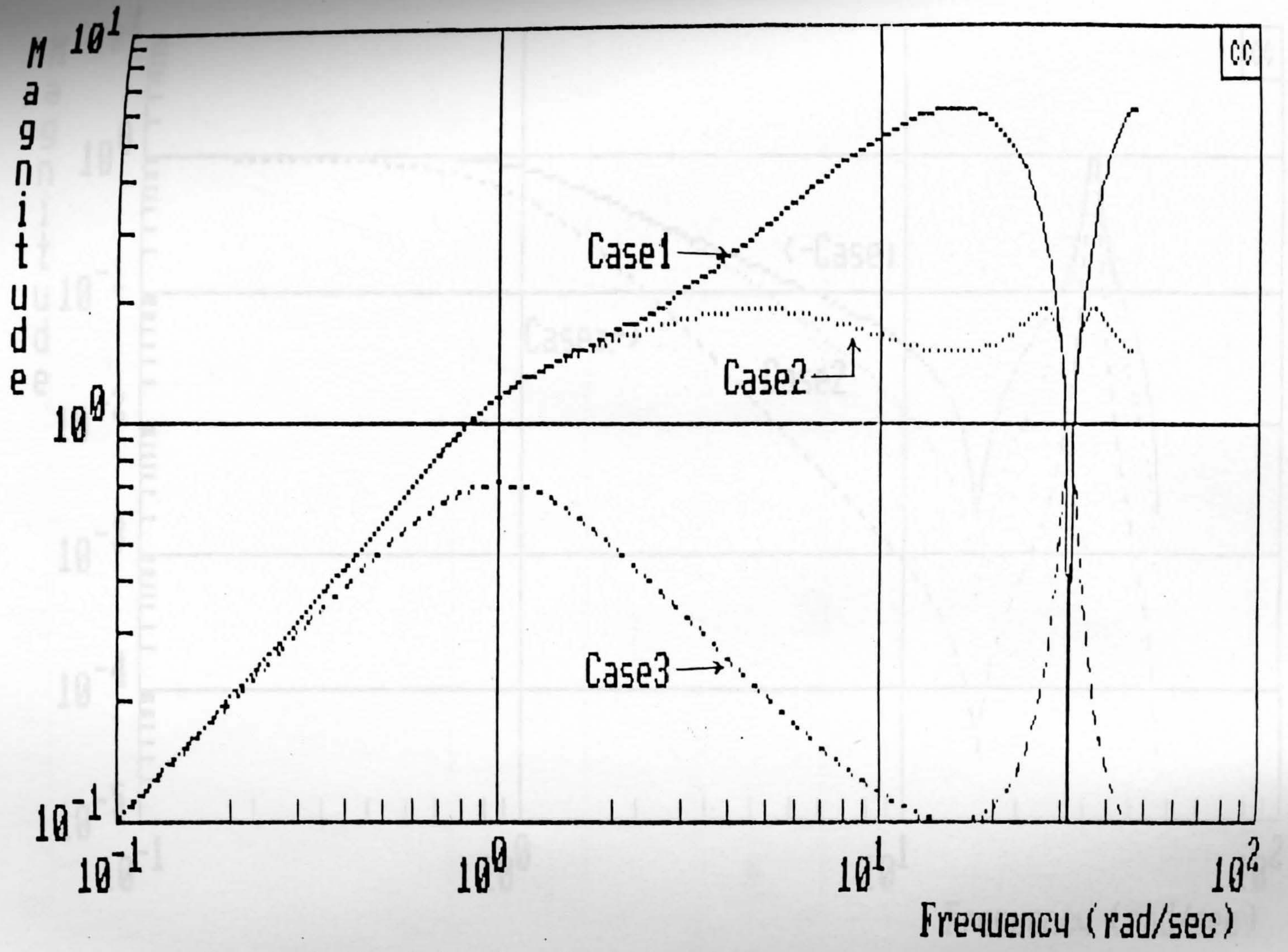

Figure 4.22 Noise-to-plant input frequency response of Example 1

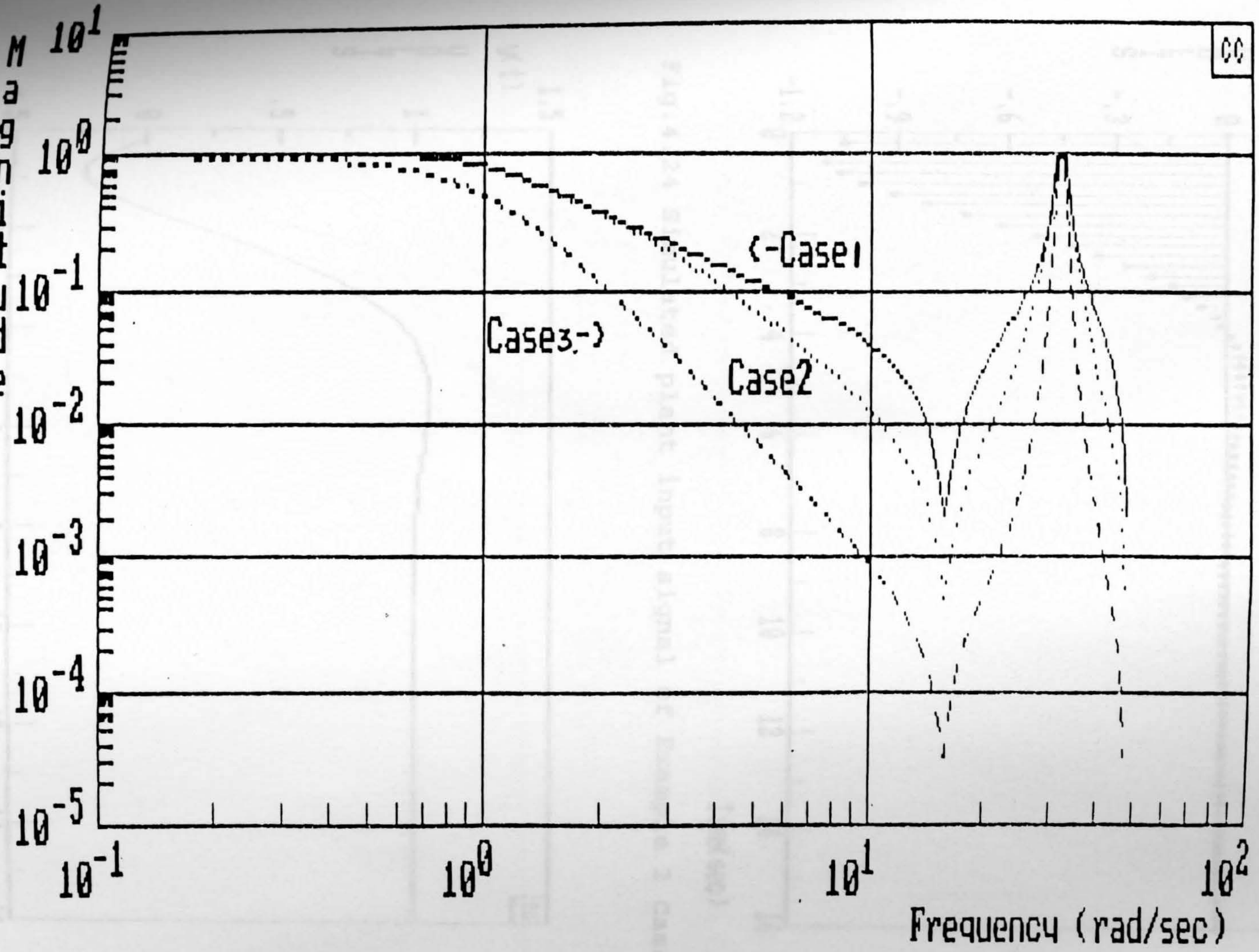

Figure 4.23 Noise-to-plant output frequency response of Example 1

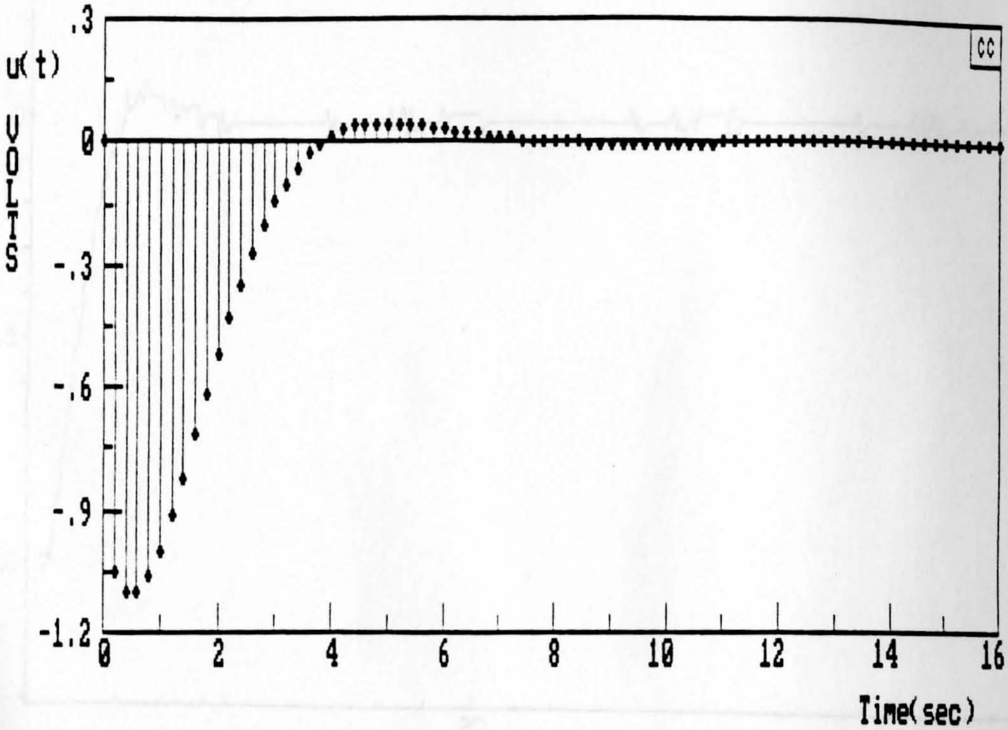

Fig.4.24 Simulated plant input signal of Example 2 Case 1

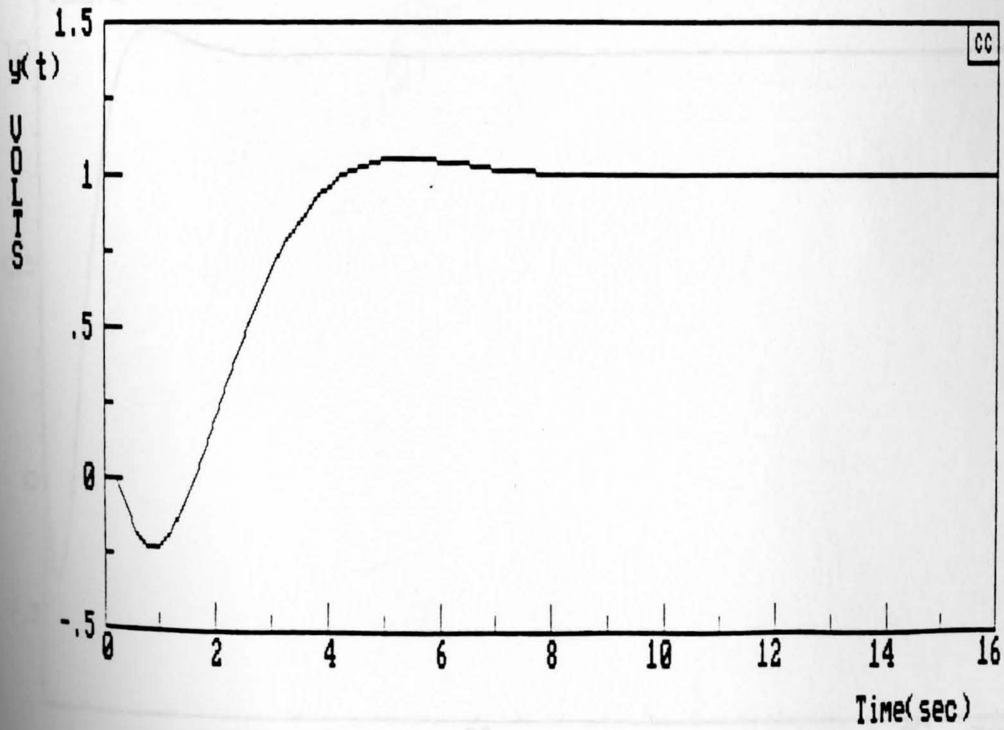

Fig.4.25 Simulated plant output signal of Example 2 Case 1

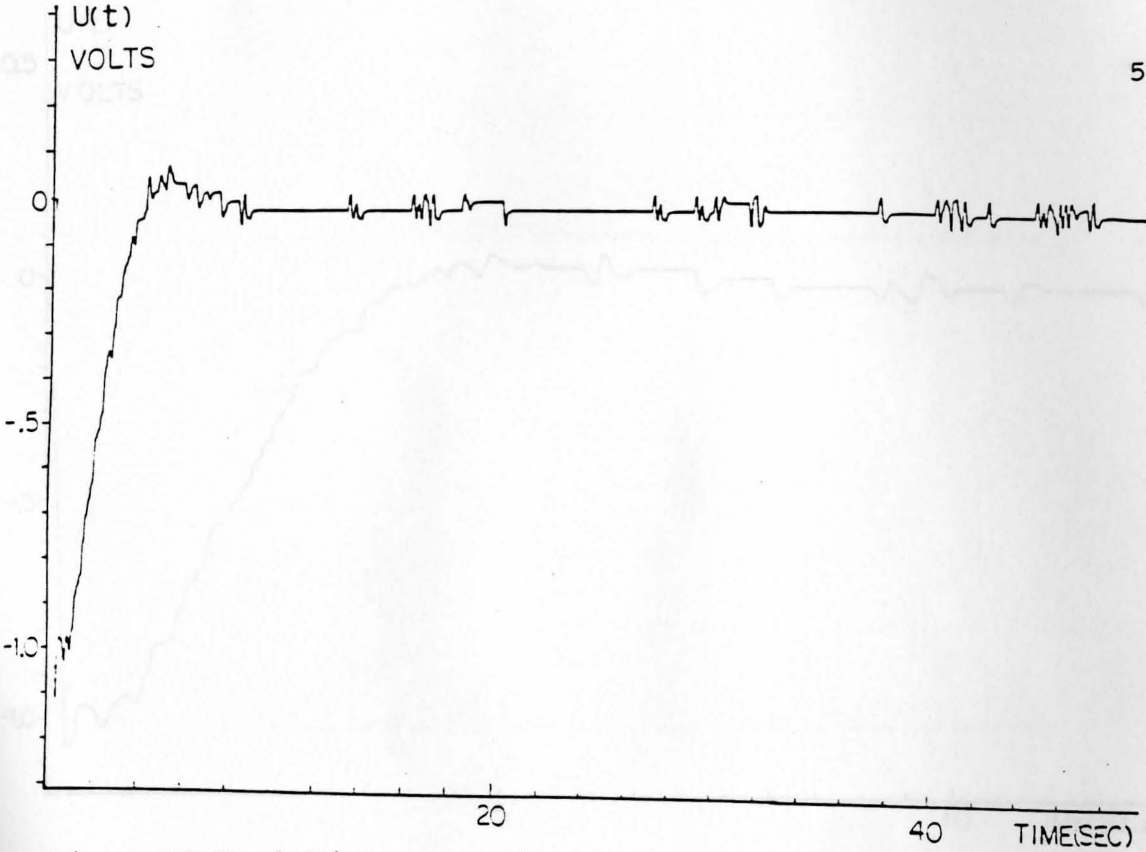

Fig.4.26 Real-Time plant input signal of Example 2 Case 1

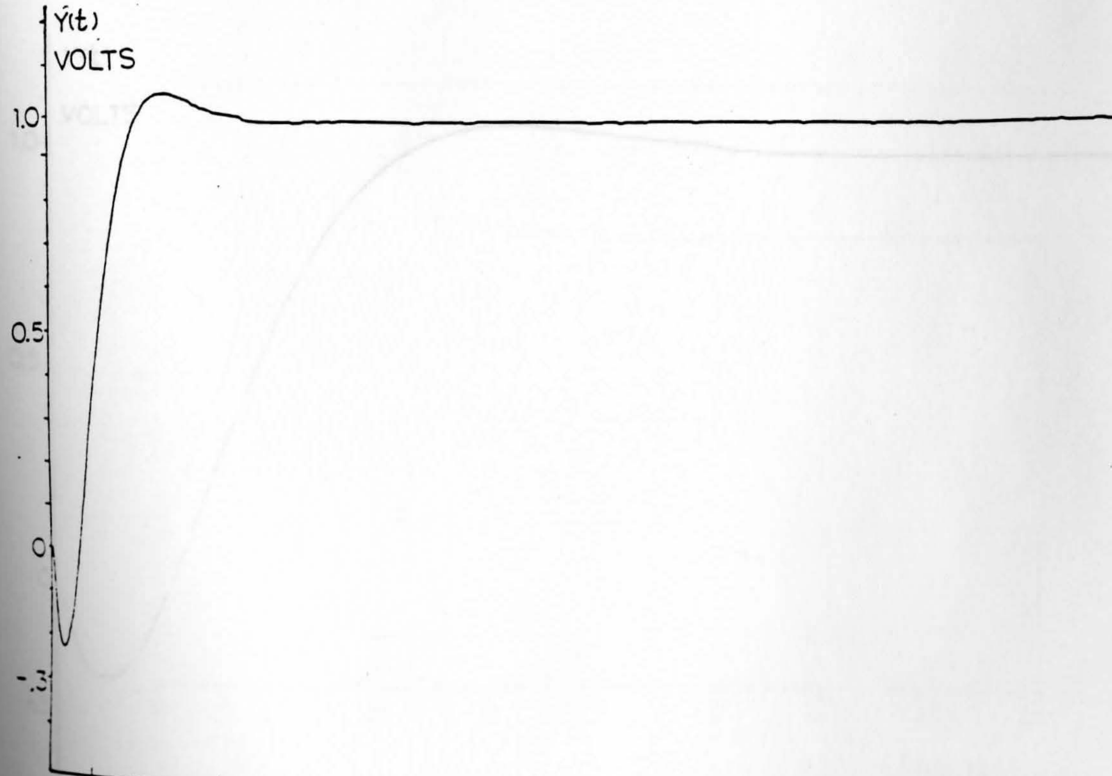

Fig.4.27 Real-Time plant output signal of Example 2 Case 1

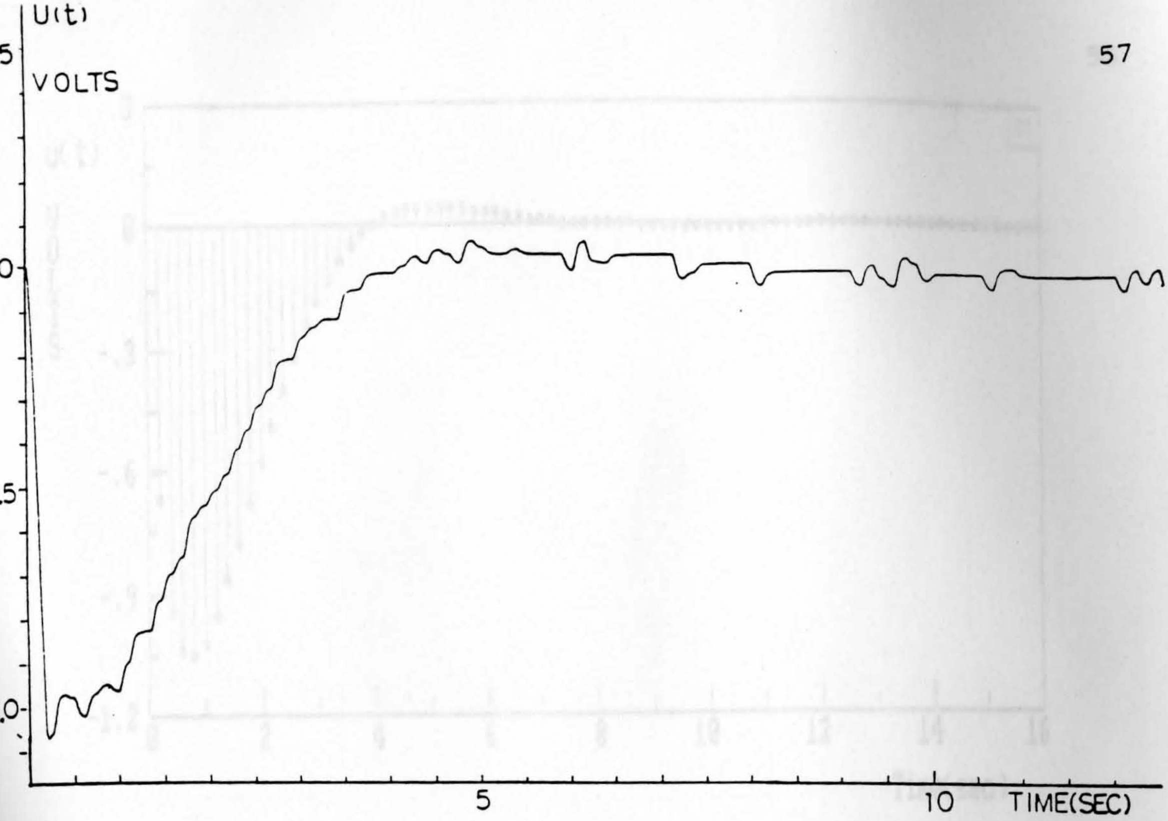

Fig.4.28 Real-time plant input signal of Example 2 Case 1

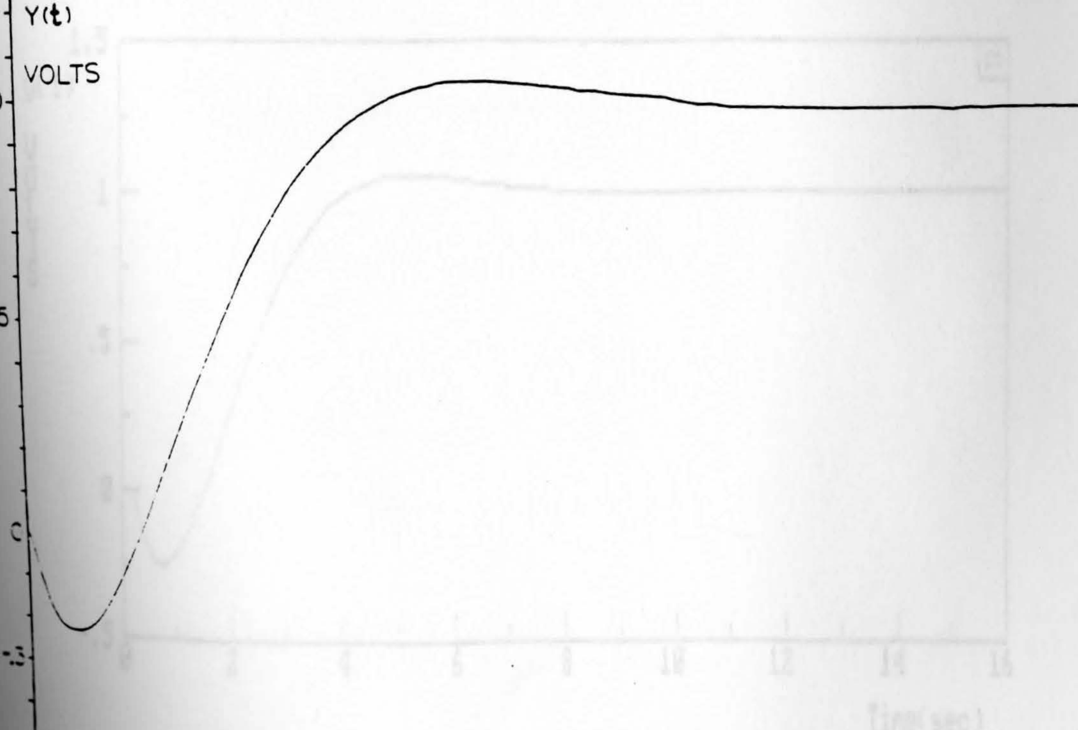

Fig.4.29 Real-time plant output signal of Example 2 Case 1

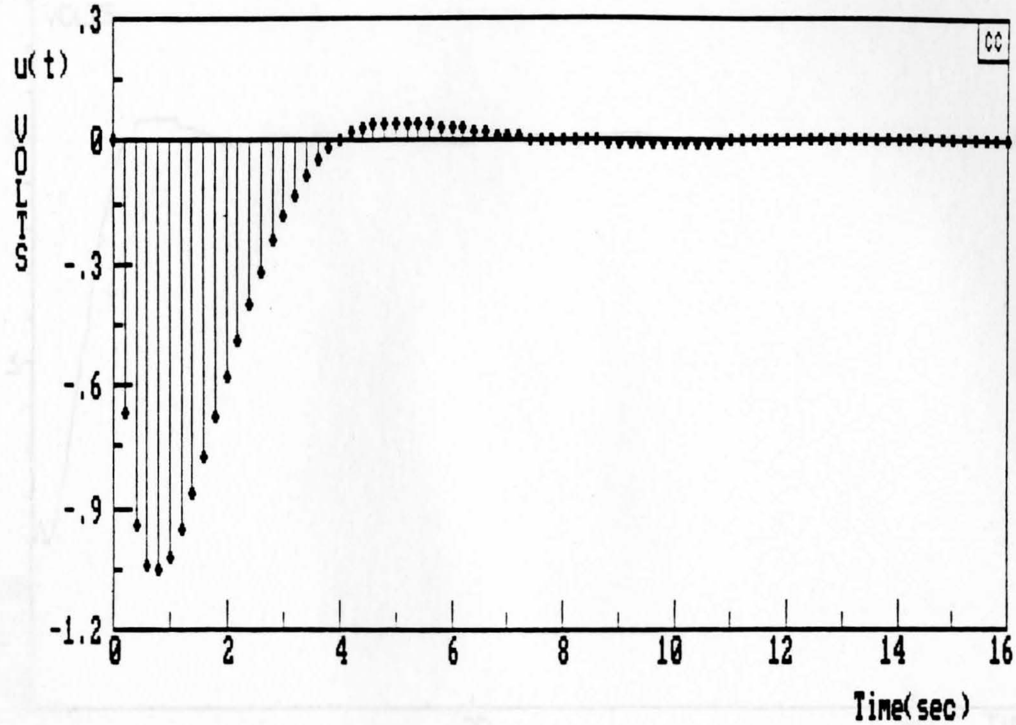

Fig.4.30 Simulated plant input signal of Example 2 Case 2

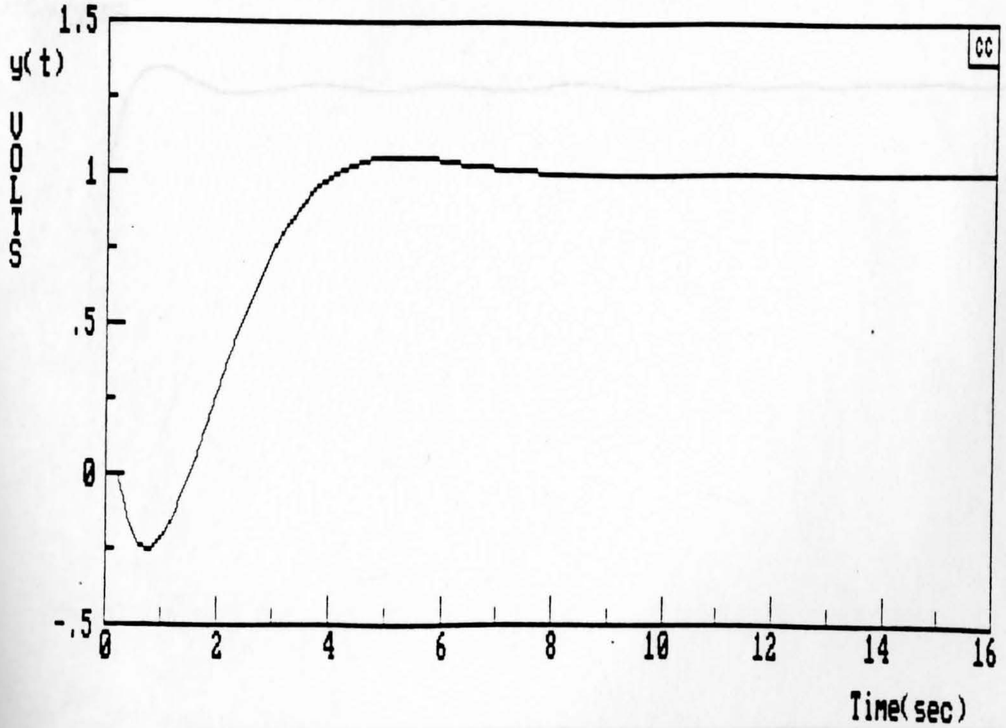

Fig.4.31 Simulated plant output signal of Example 2 Case 2

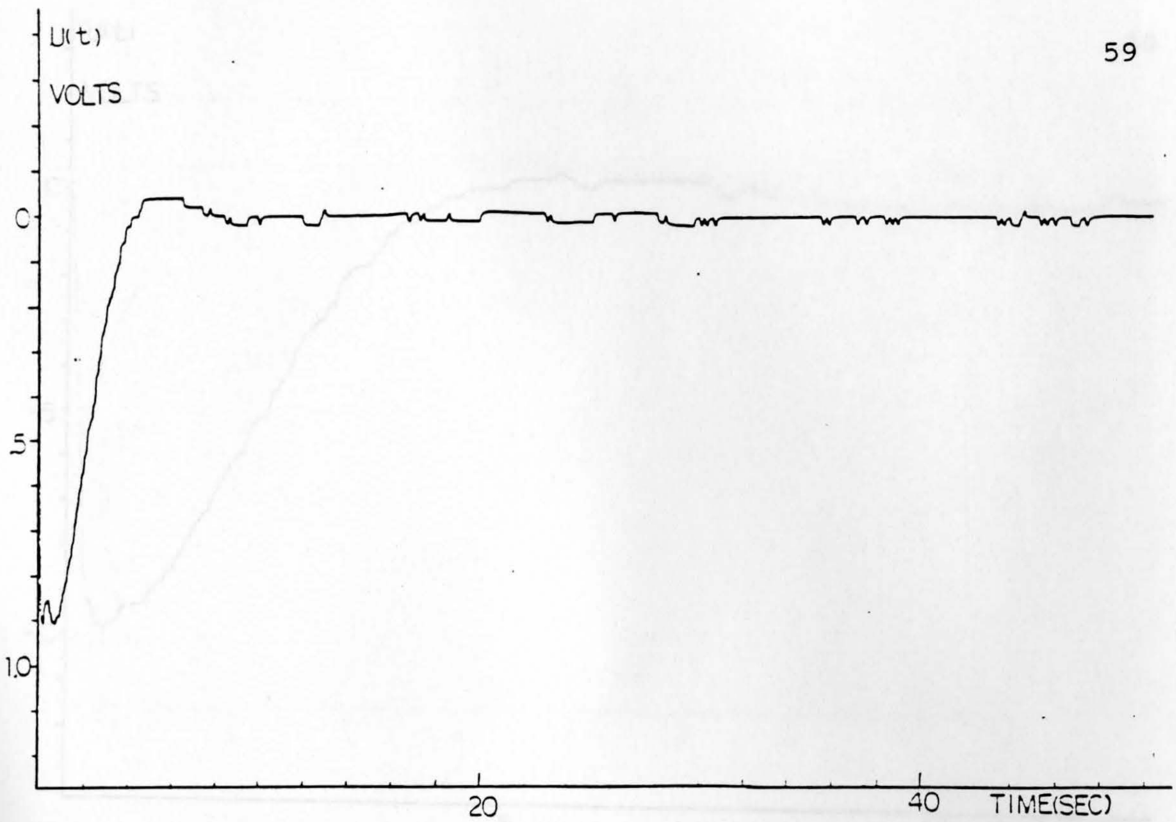

Fig.4.32 Real-time plant input signal of Example 2 Case 2

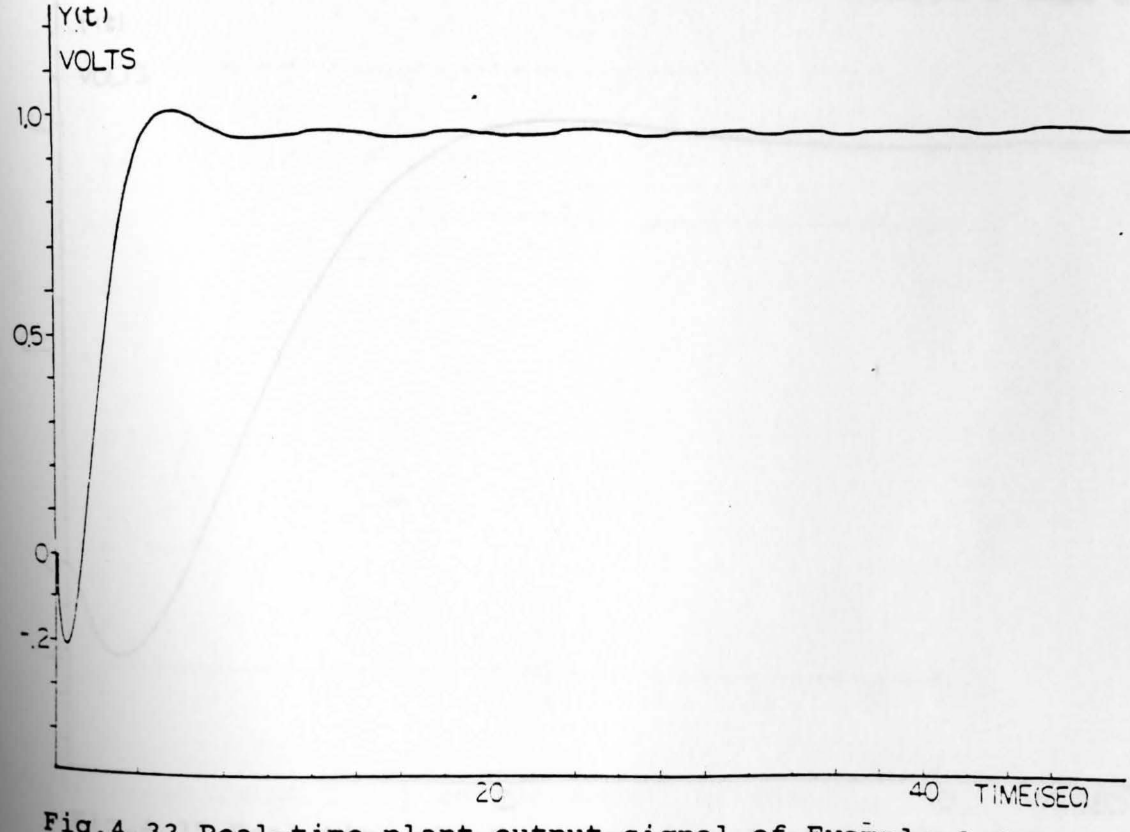

Fig.4.33 Real-time plant output signal of Example 2 Case 2

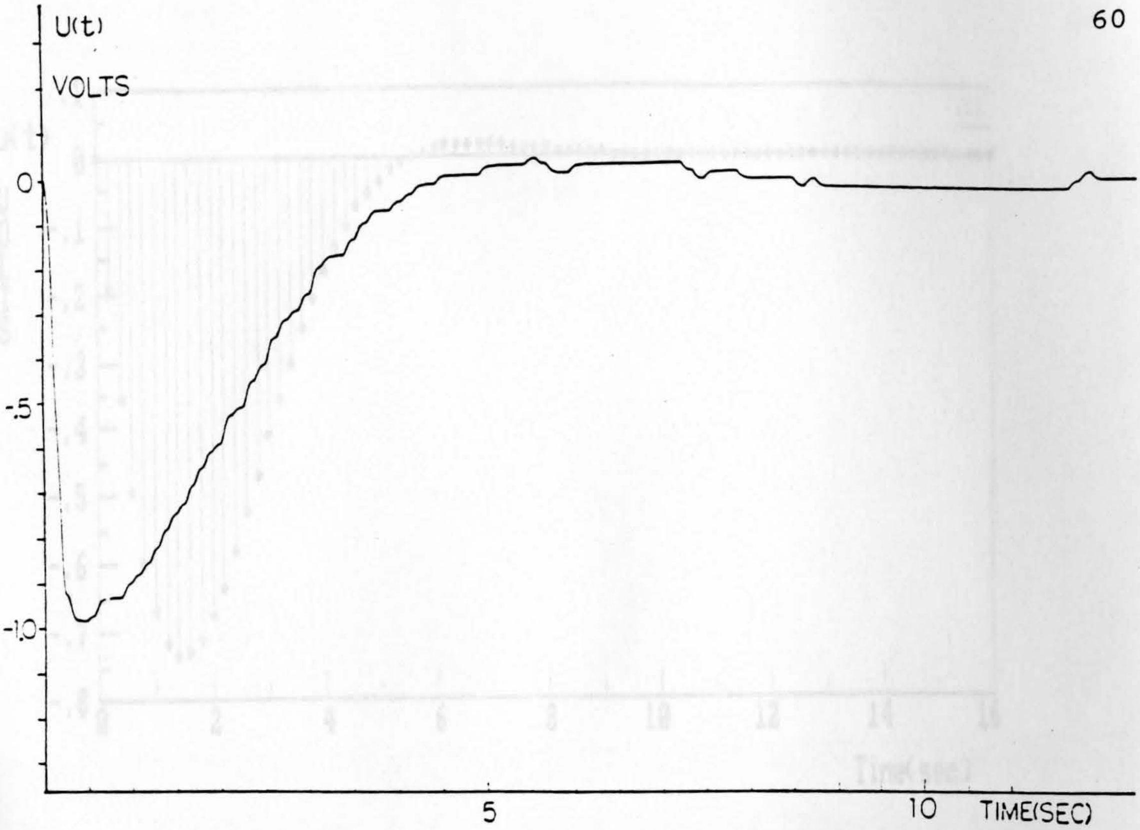

Fig.4.34 Real-time plant input signal of Example 2 Case 2

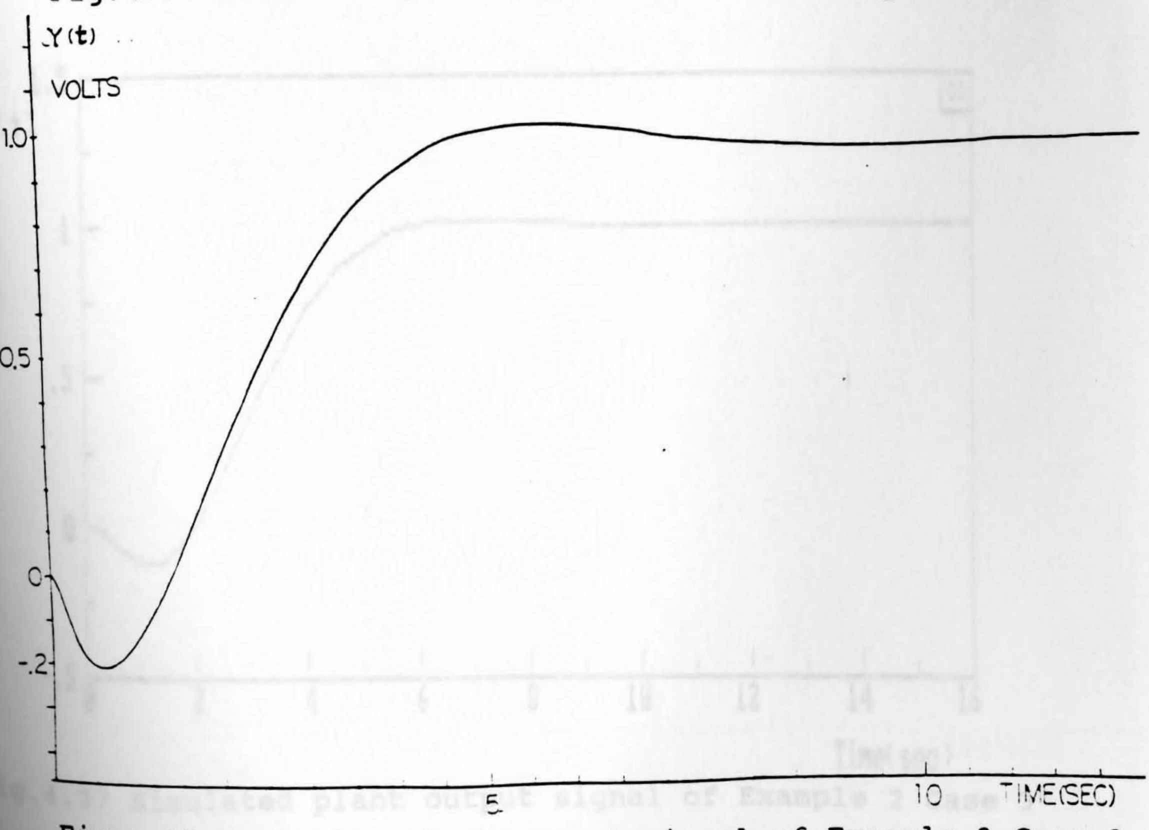

Fig.4.35 Real-time plant output signal of Example 2 Case 2

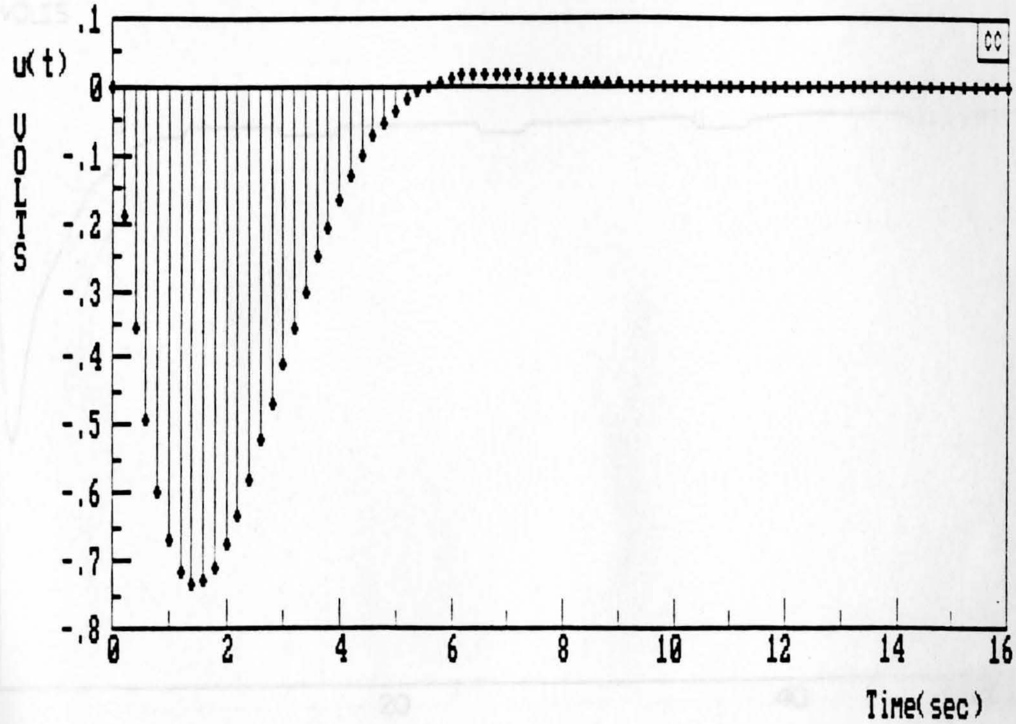

Fig.4.36 Simulated plant input signal of Example 2 Case 3

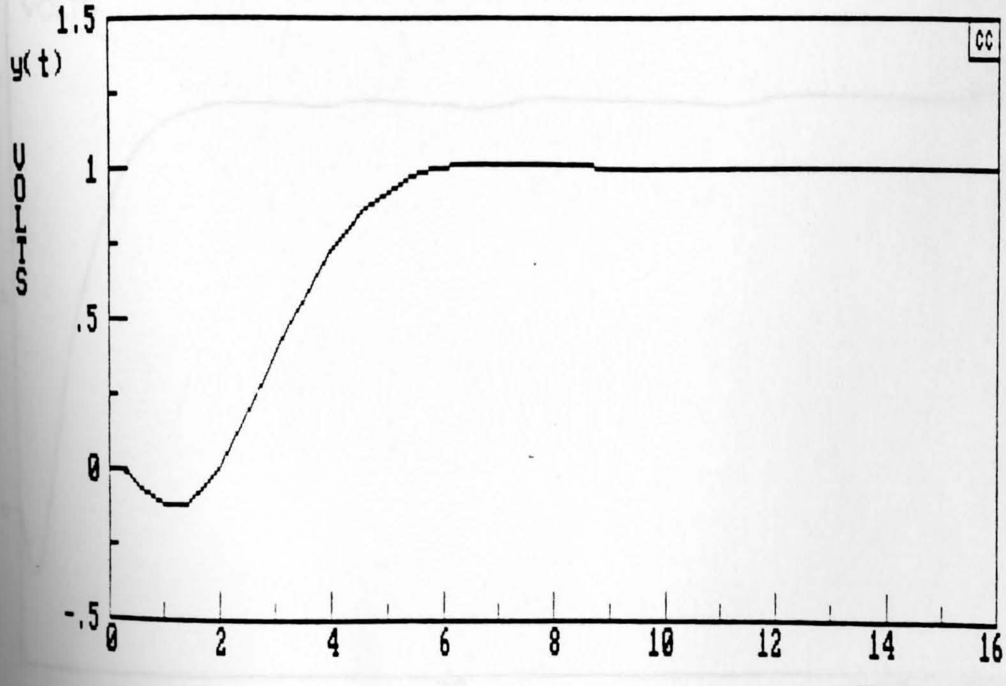

Fig.4.37 Simulated plant output signal of Example 2 Case 3

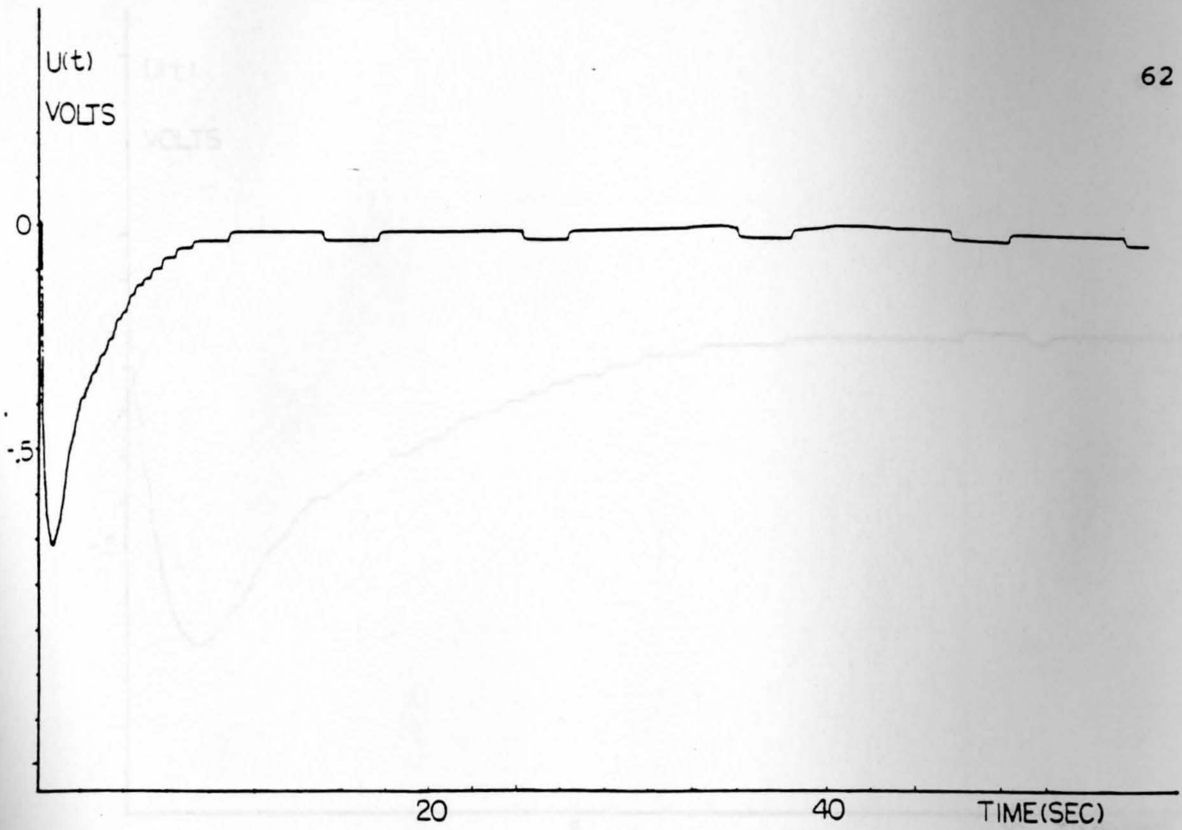

Fig.4.38 Real-time plant input signal of Example 2 Case 3

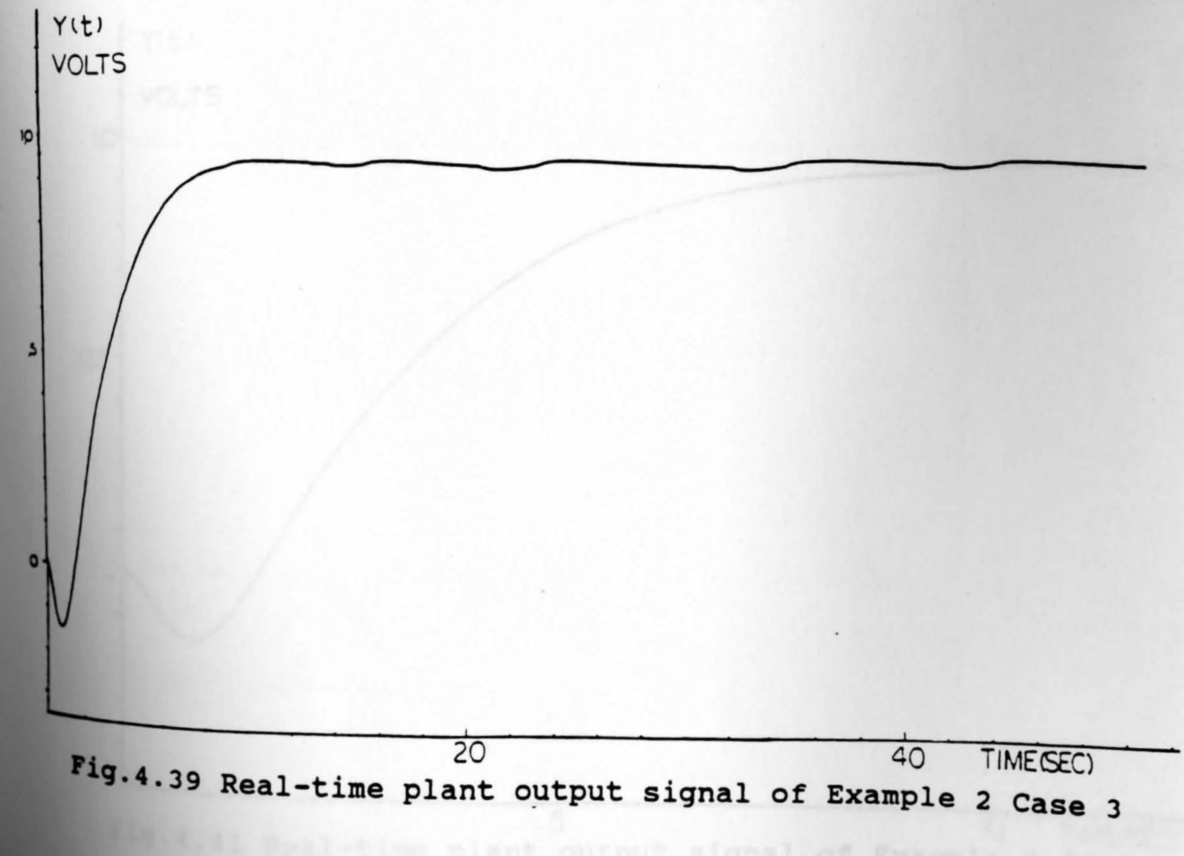

Fig.4.39 Real-time plant output signal of Example 2 Case 3

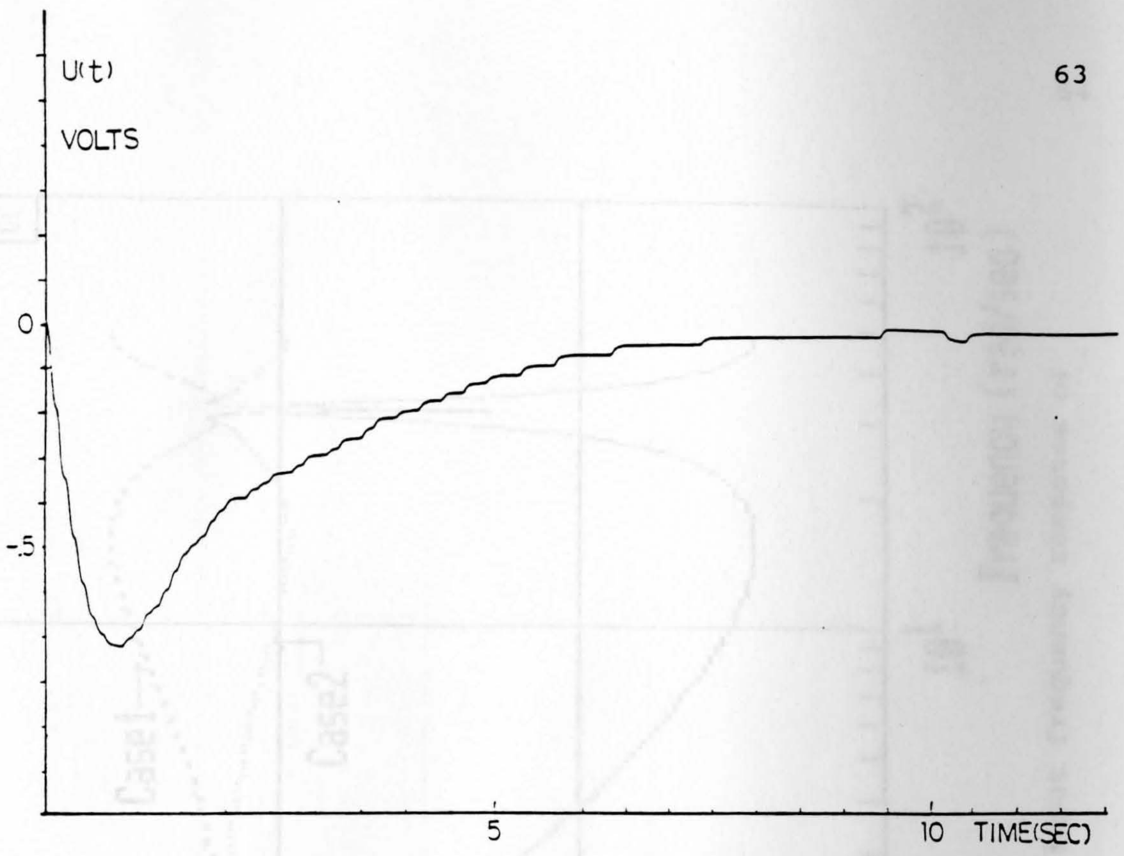

Fig.4.40 Real-time plant input signal of Example 2 Case 3

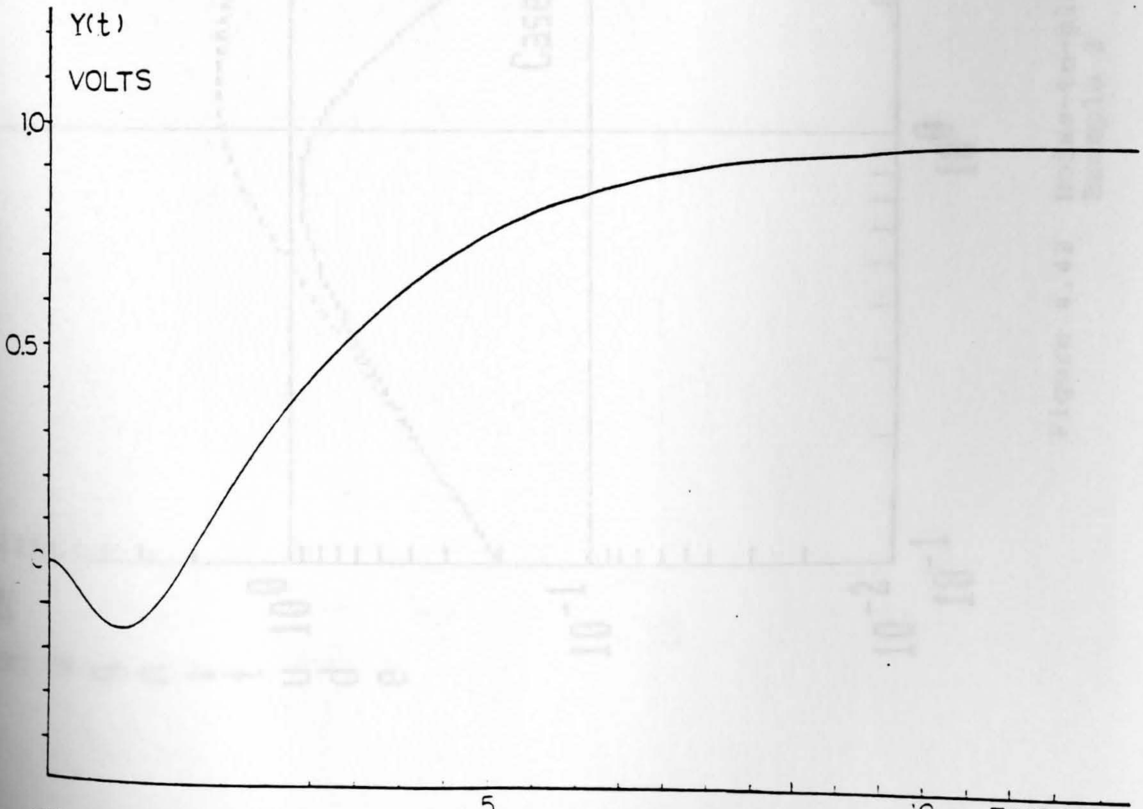

Fig.4.41 Real-time plant output signal of Example 2 Case 3

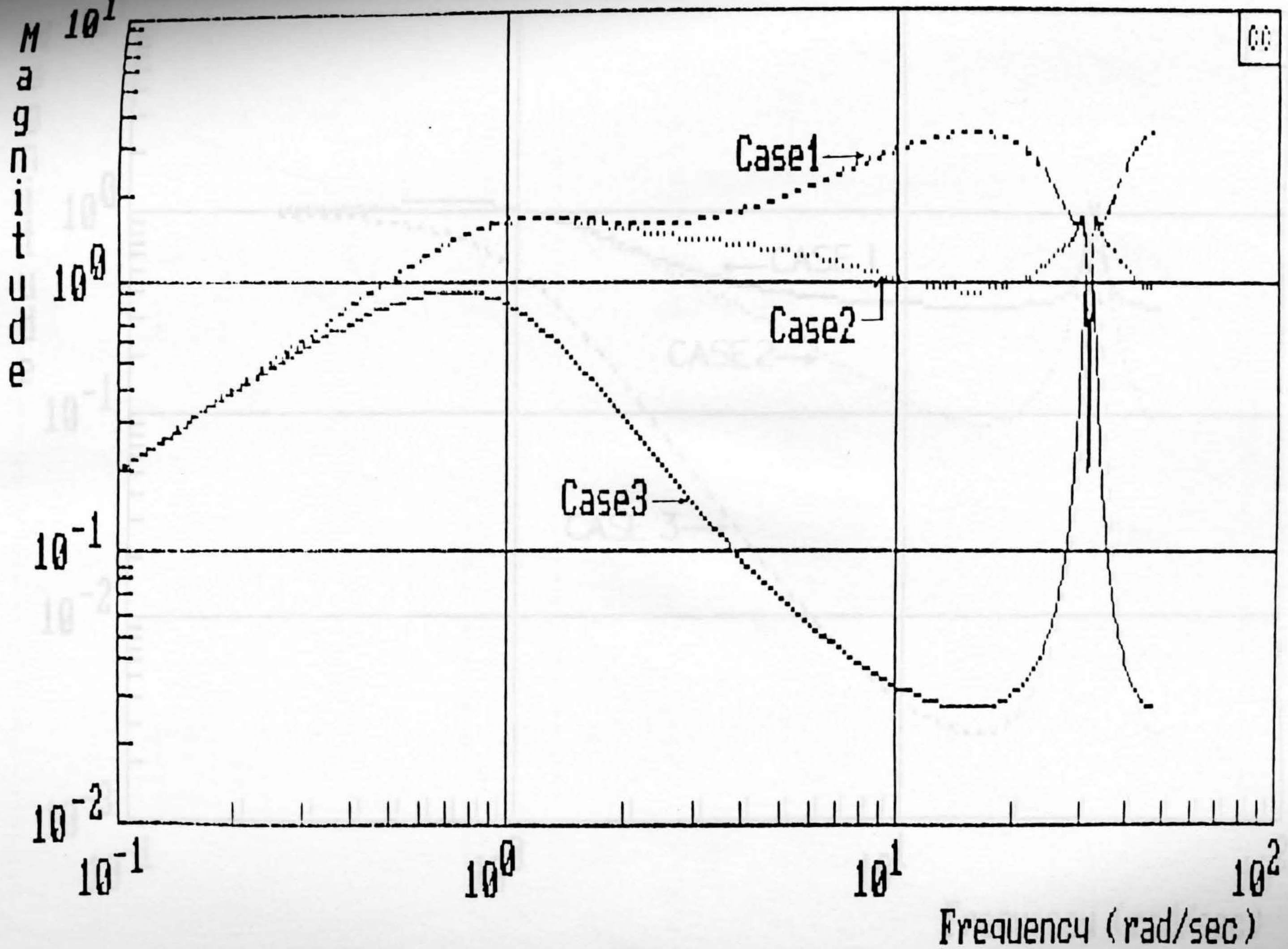

Figure 4.42 Noise-to-plant input frequency response of Example 2

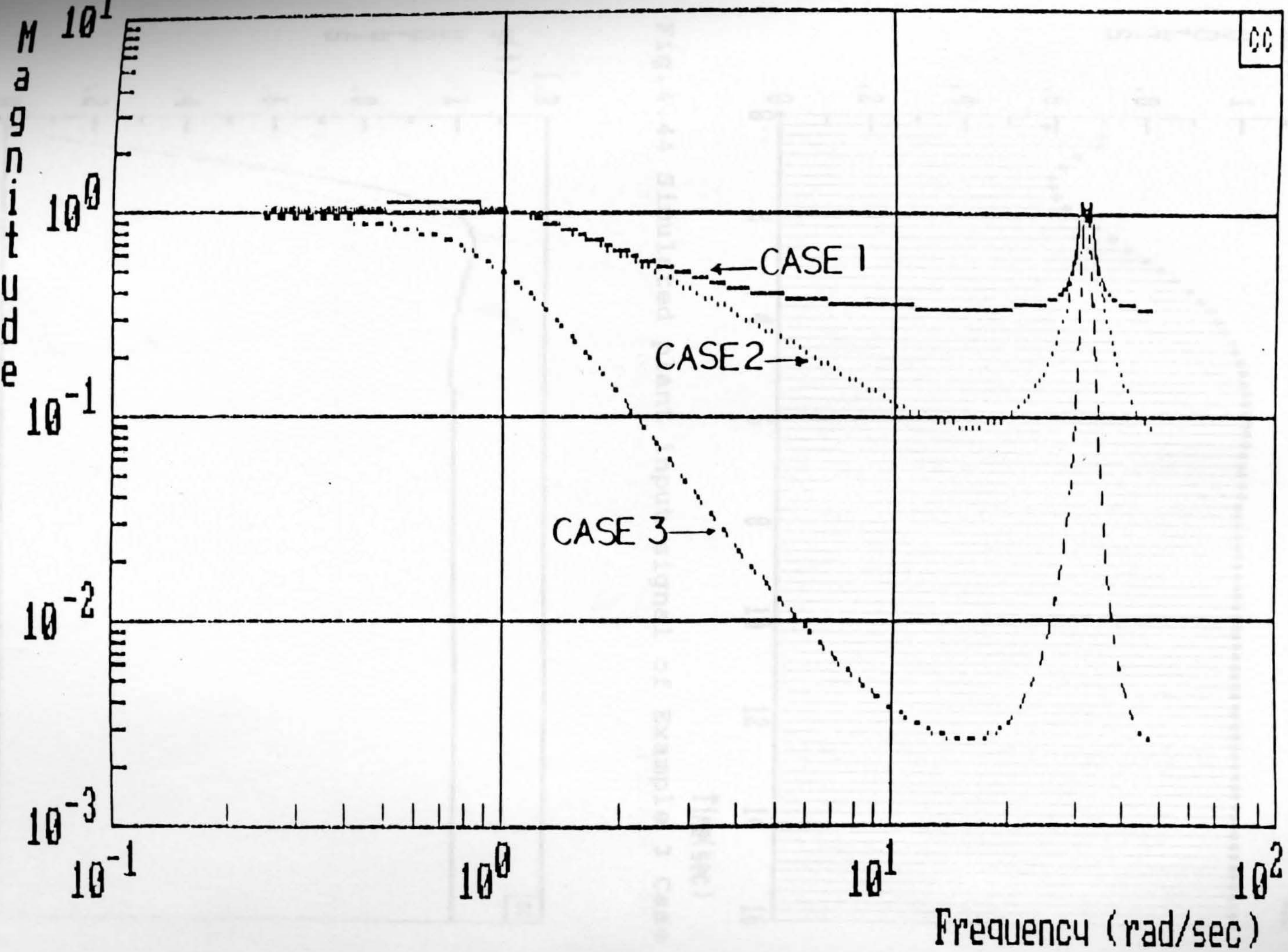

Figure 4.43 Noise-to-plant output frequency response of Example 2

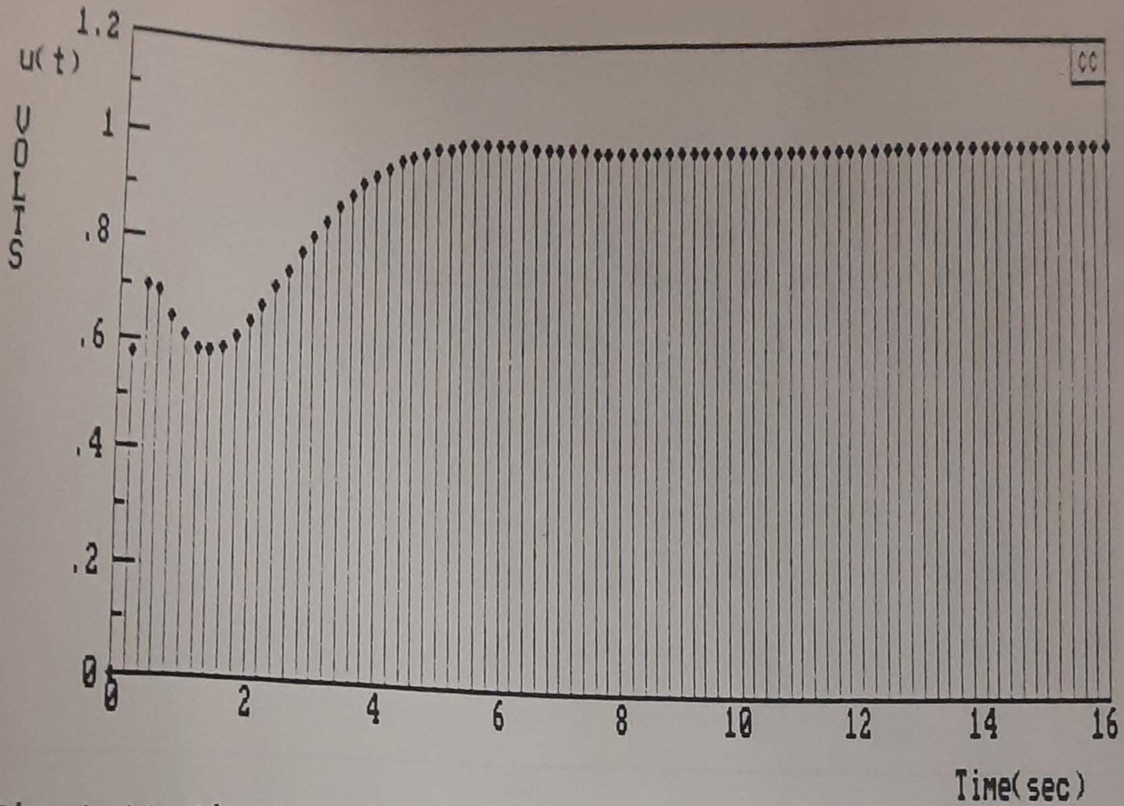

Fig.4.44 Simulated plant input signal of Example 3 Case 1

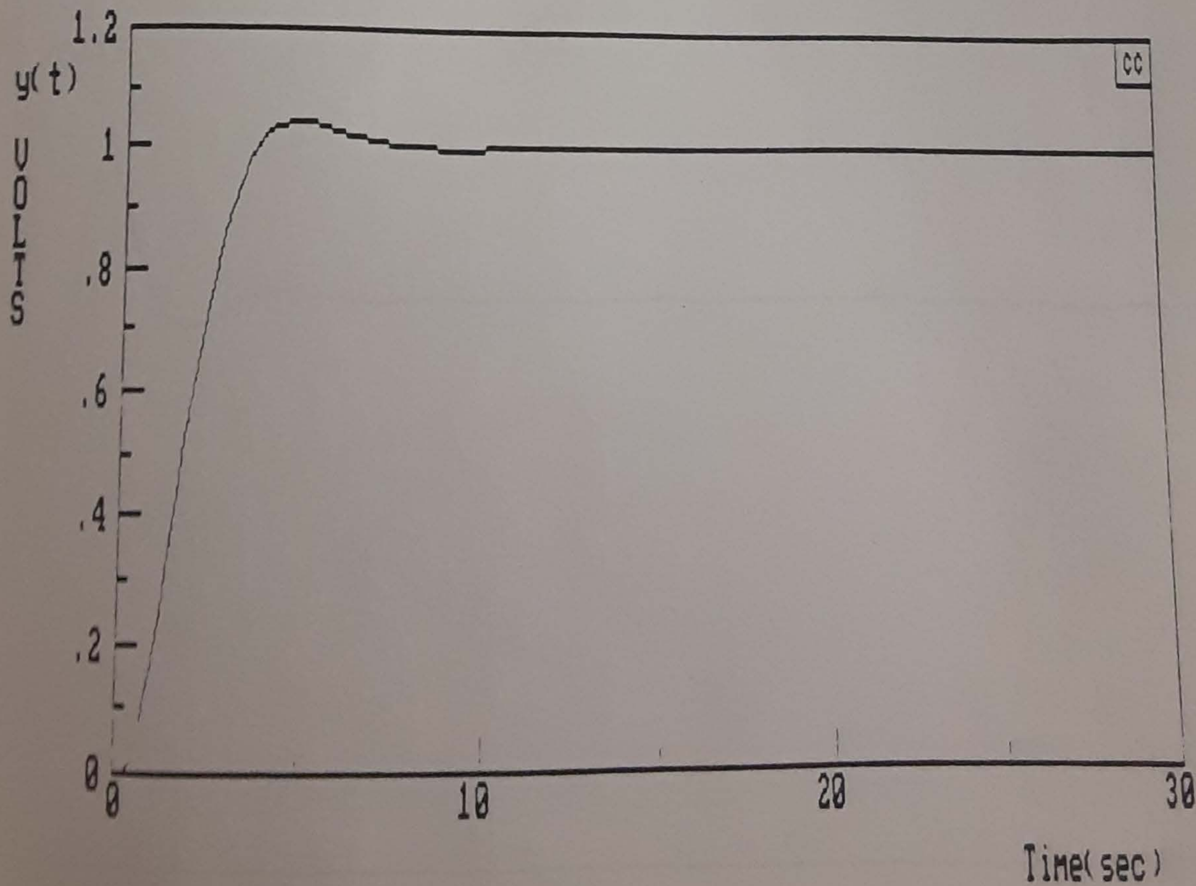

Fig.4.45 Simulated plant output signal of Example 3 Case 1

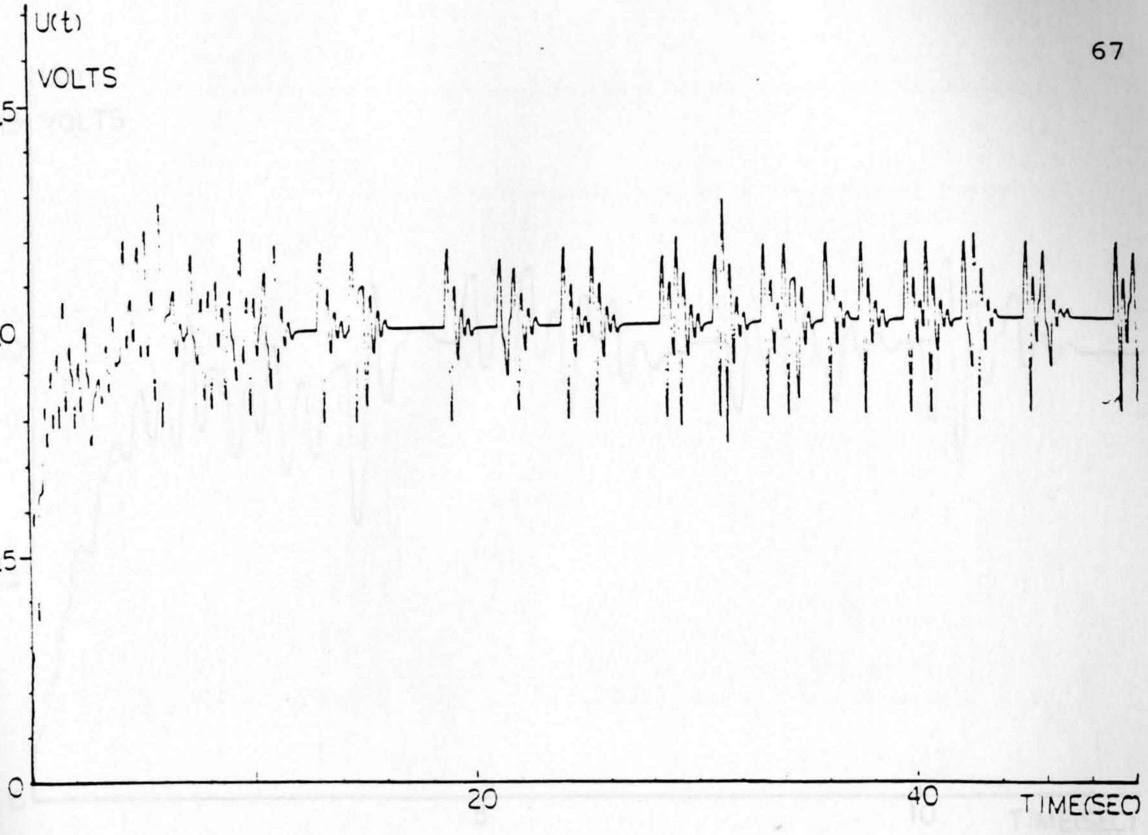

Fig.4.46 Real-Time plant input signal of Example 3 Case 1

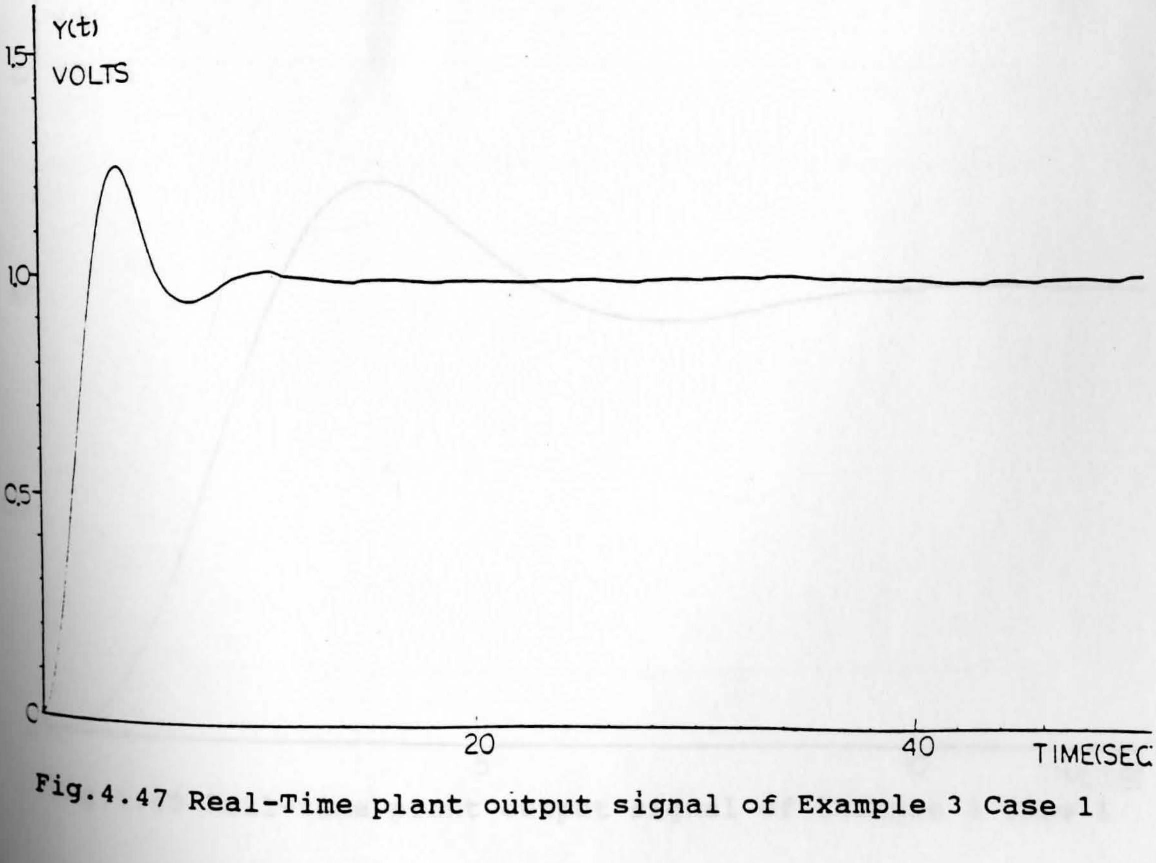

Fig.4.47 Real-Time plant output signal of Example 3 Case 1

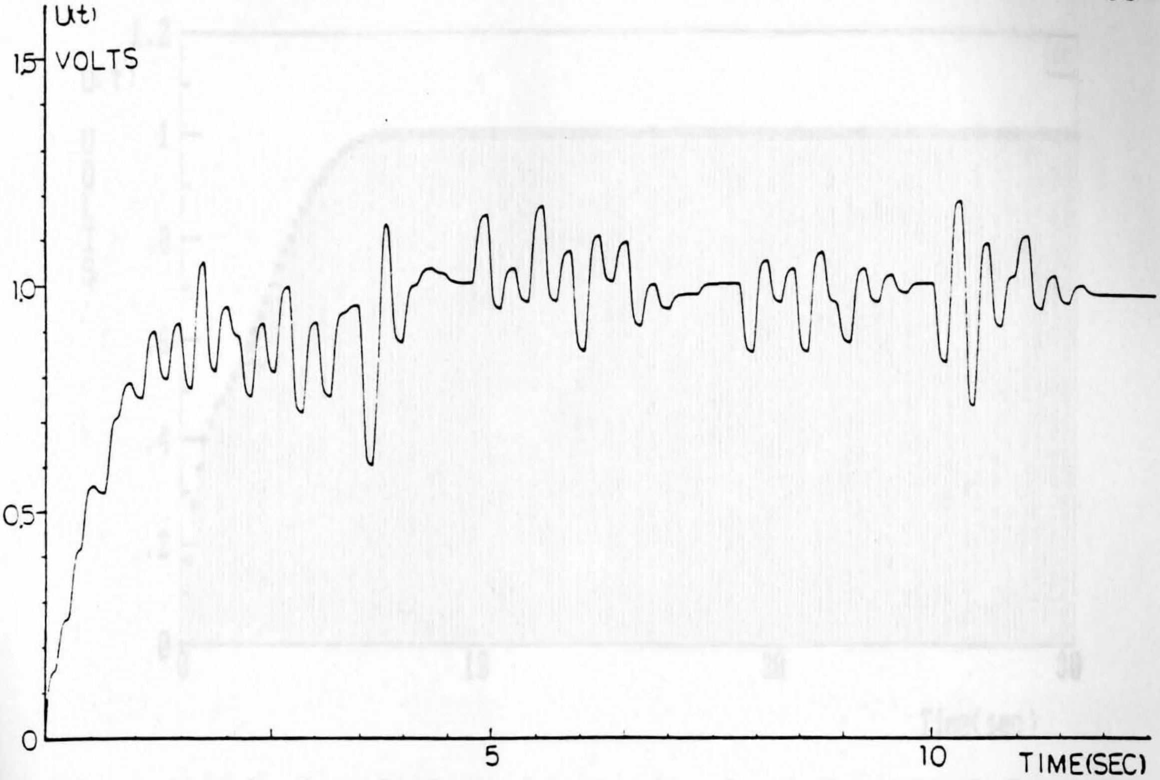

Fig.4.48 Real-time plant input signal of Example 3 Case 1

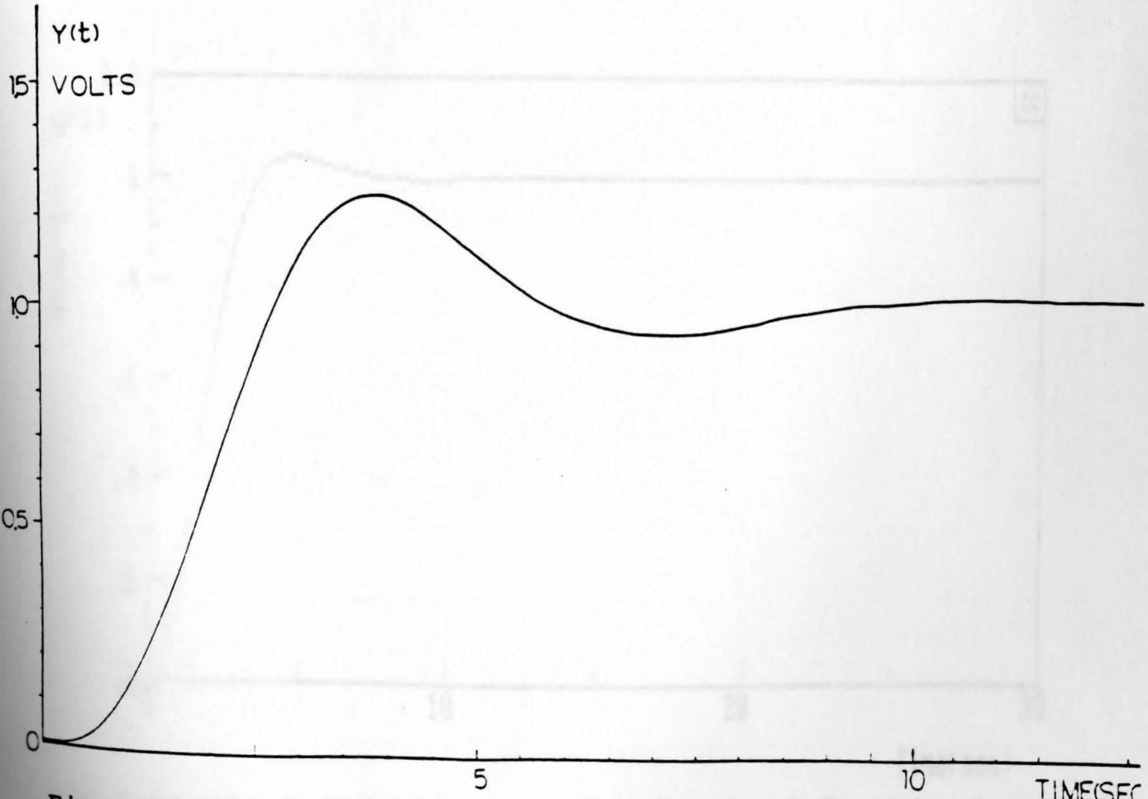

Fig.4.49 Real-time plant output signal of Example 3 Case 1

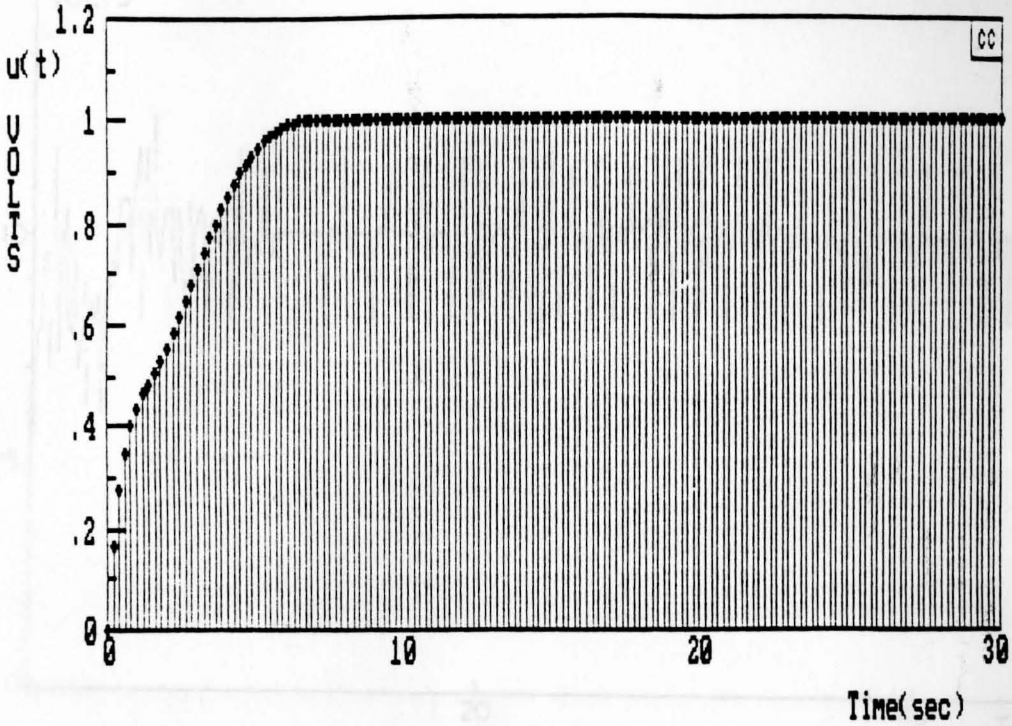

Fig.4.50 Simulated plant input signal of Example 3 Case 2

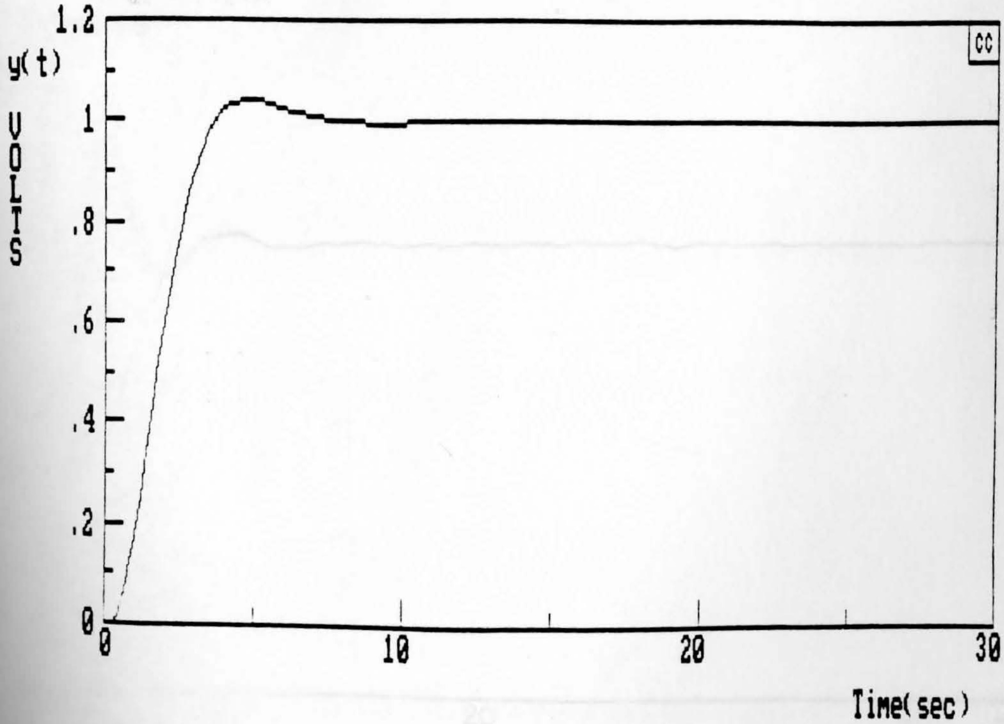

Fig.4.51 Simulated plant output signal of Example 3 Case 2

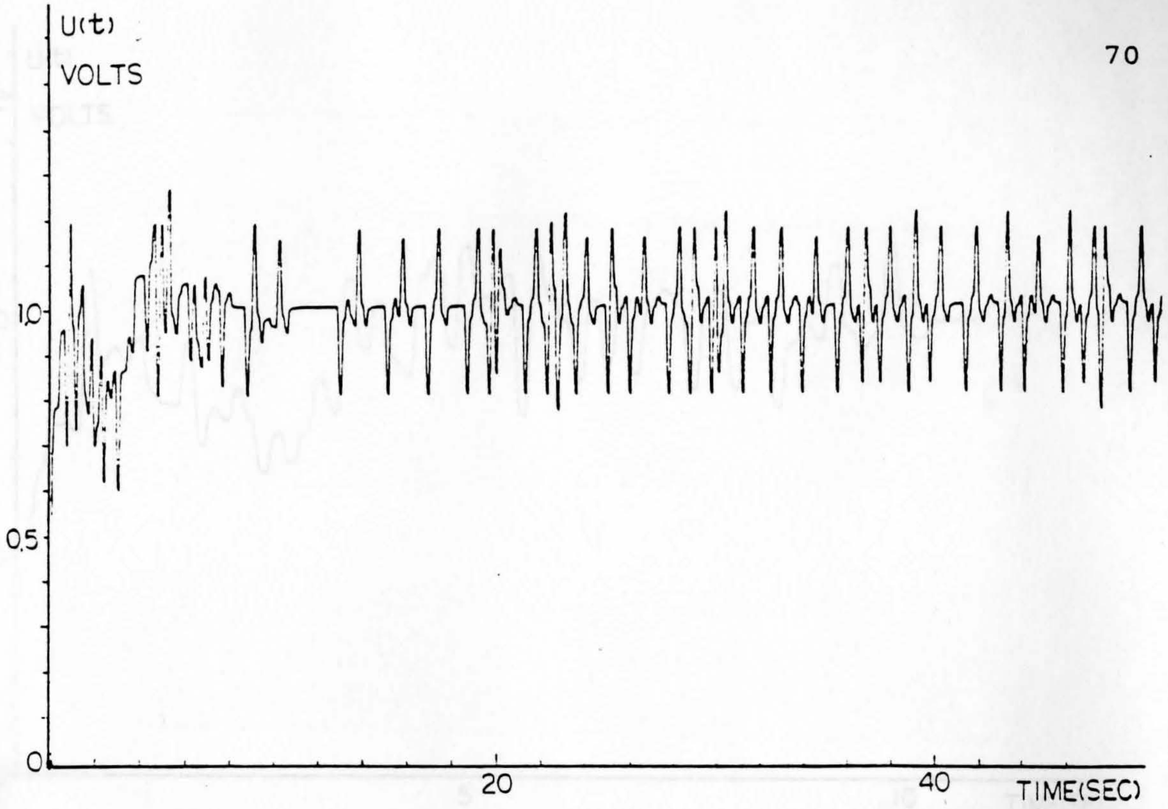

Fig.4.52 Real-time plant input signal of Example 3 Case 2

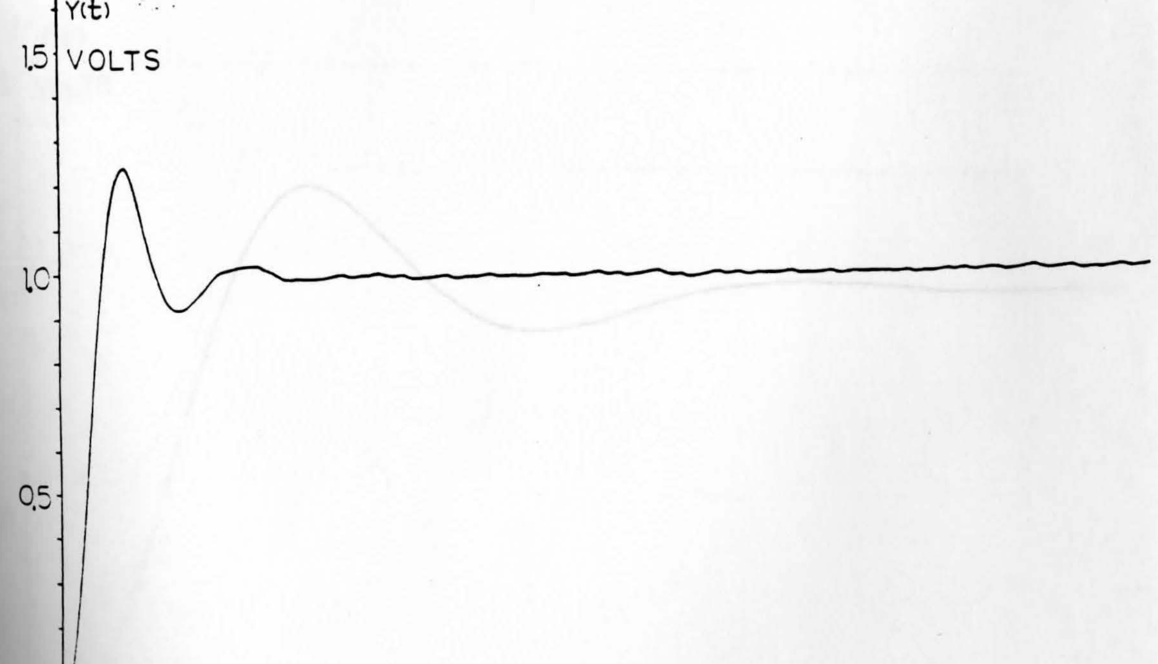

Fig.4.53 Real-time plant output signal of Example 3 Case 2

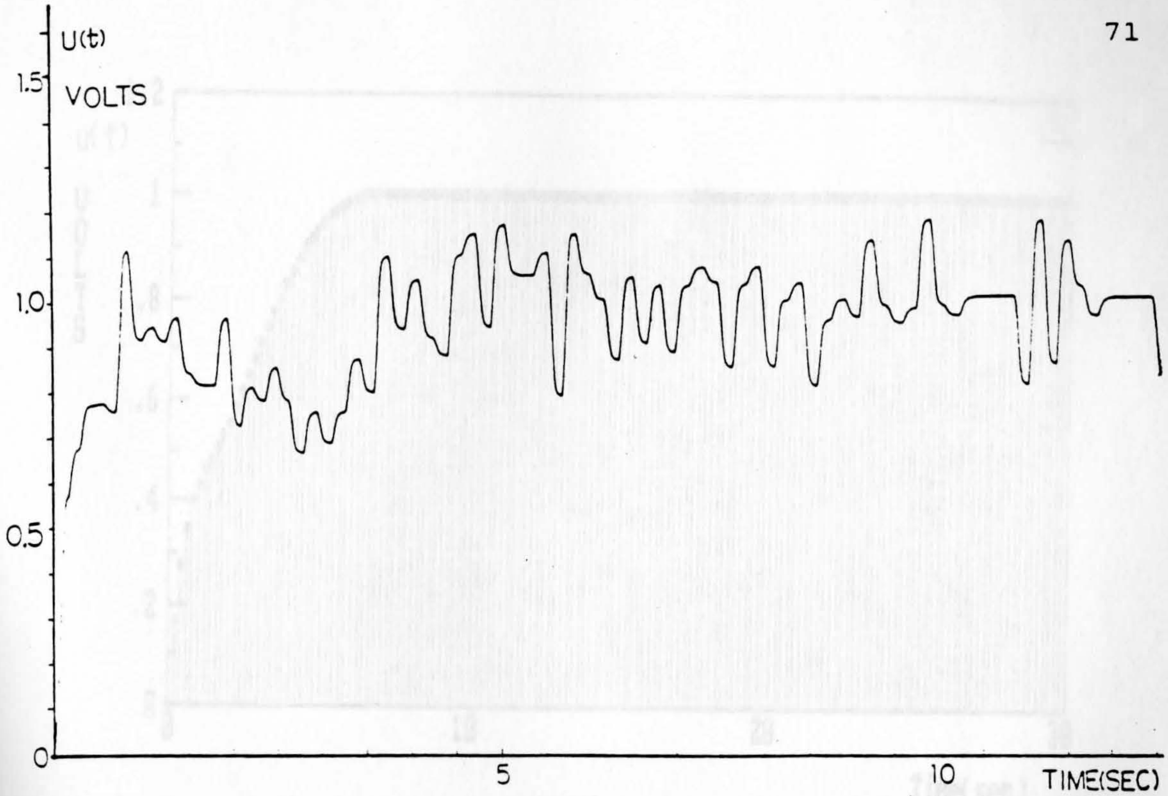

Fig.4.54 Real-time plant input signal of Example 3 Case 2

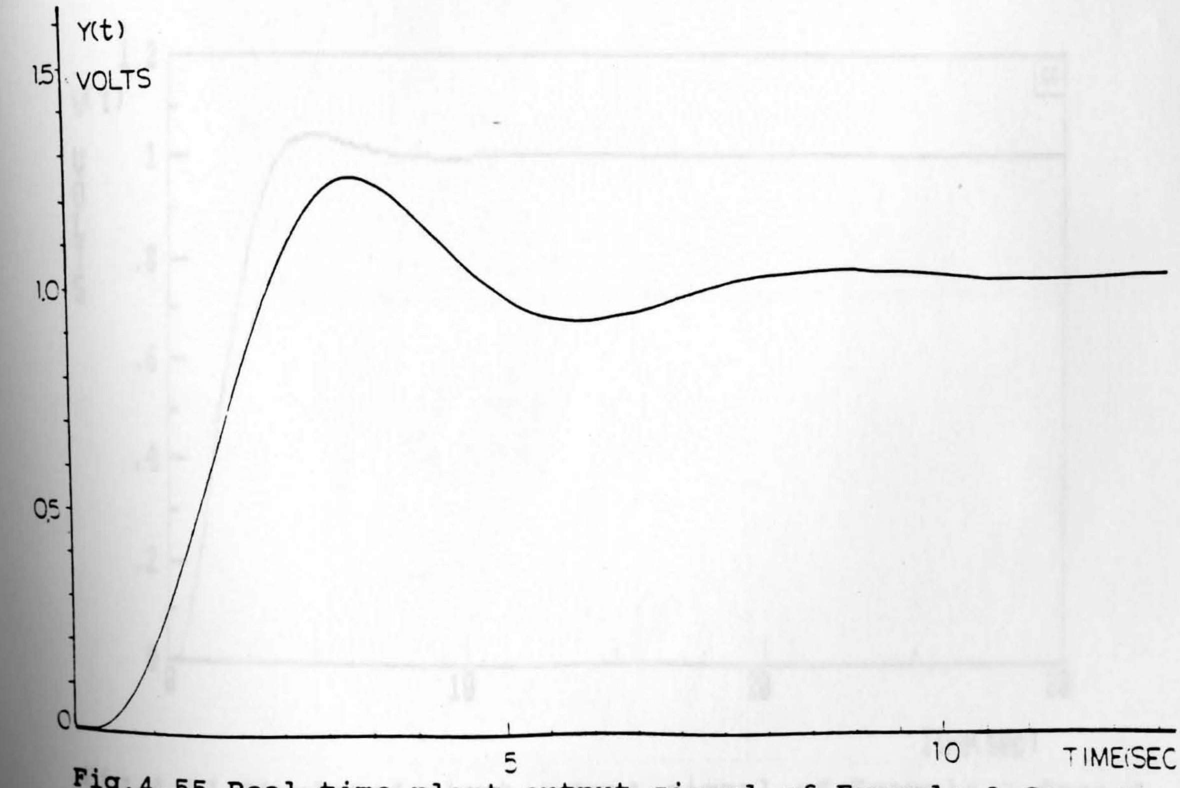

Fig.4.55 Real-time plant output signal of Example 3 Case 2

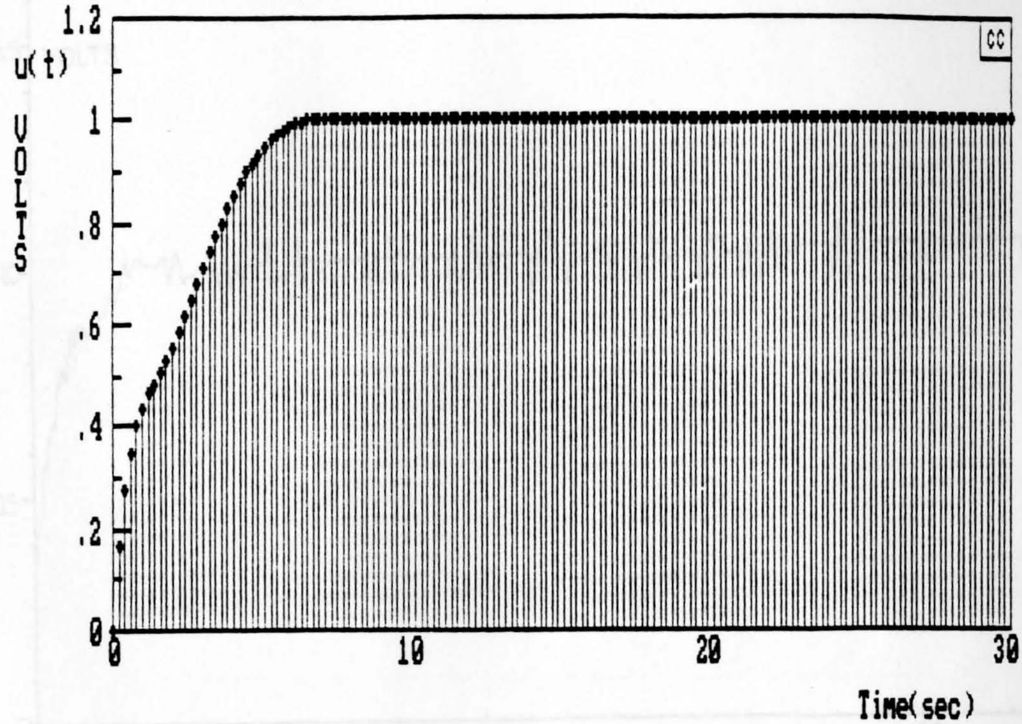

Fig.4.56 Simulated plant input signal of Example 3 Case 3

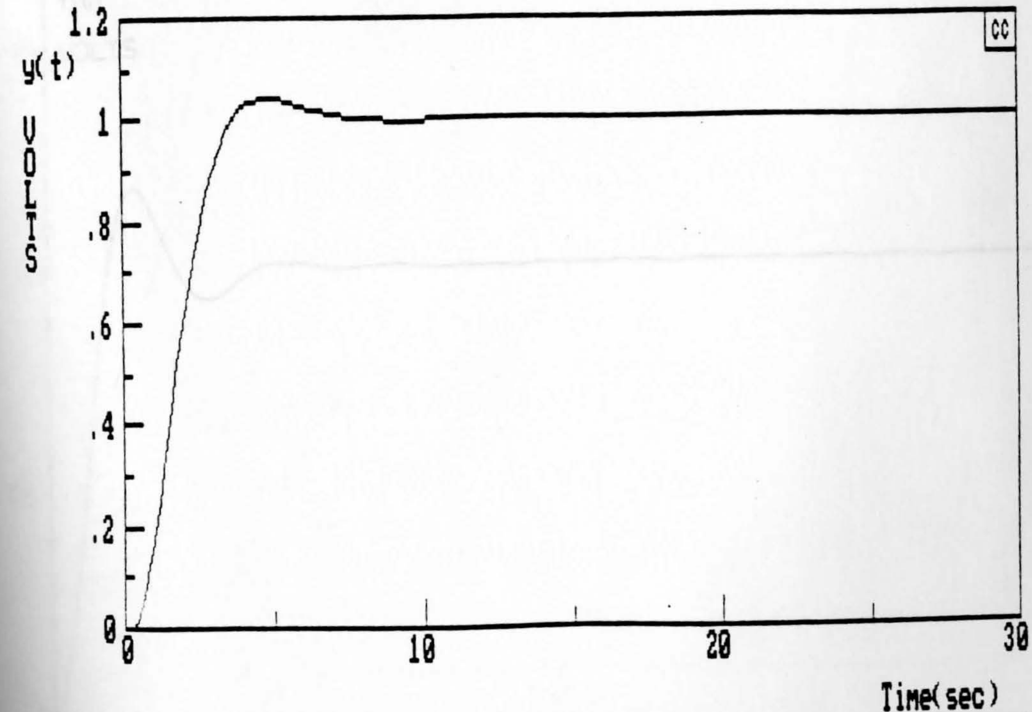

Fig.4.57 Simulated plant output signal of Example 3 Case 3

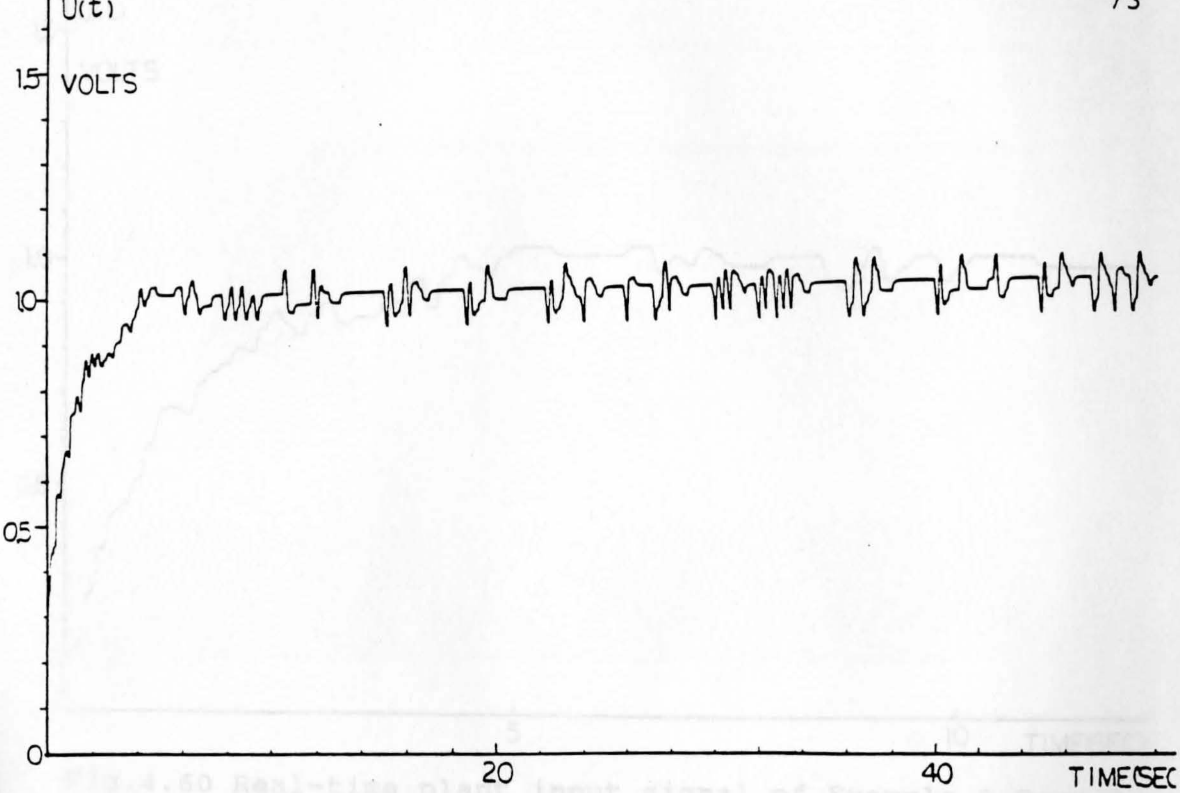

Fig.4.58 Real-time plant input signal of Example 3 Case 3

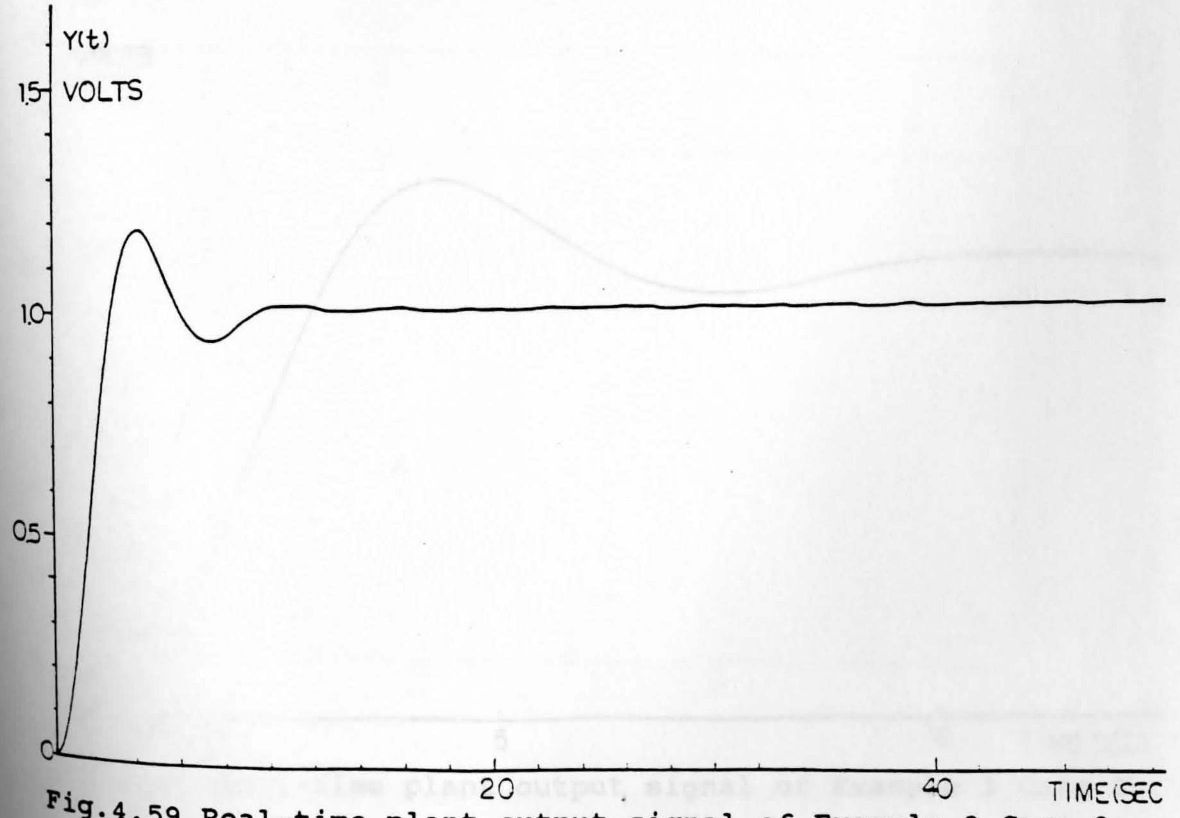

Fig.4.59 Real-time plant output signal of Example 3 Case 3

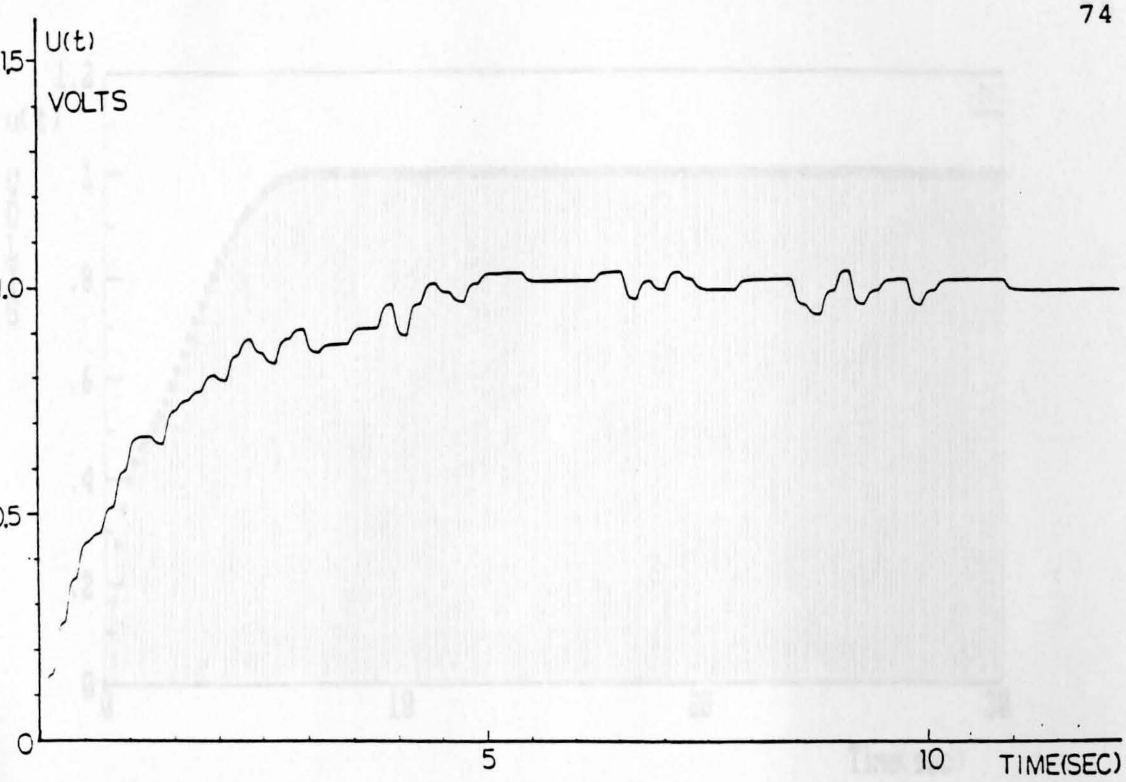

Fig.4.60 Real-time plant input signal of Example 3 Case 3

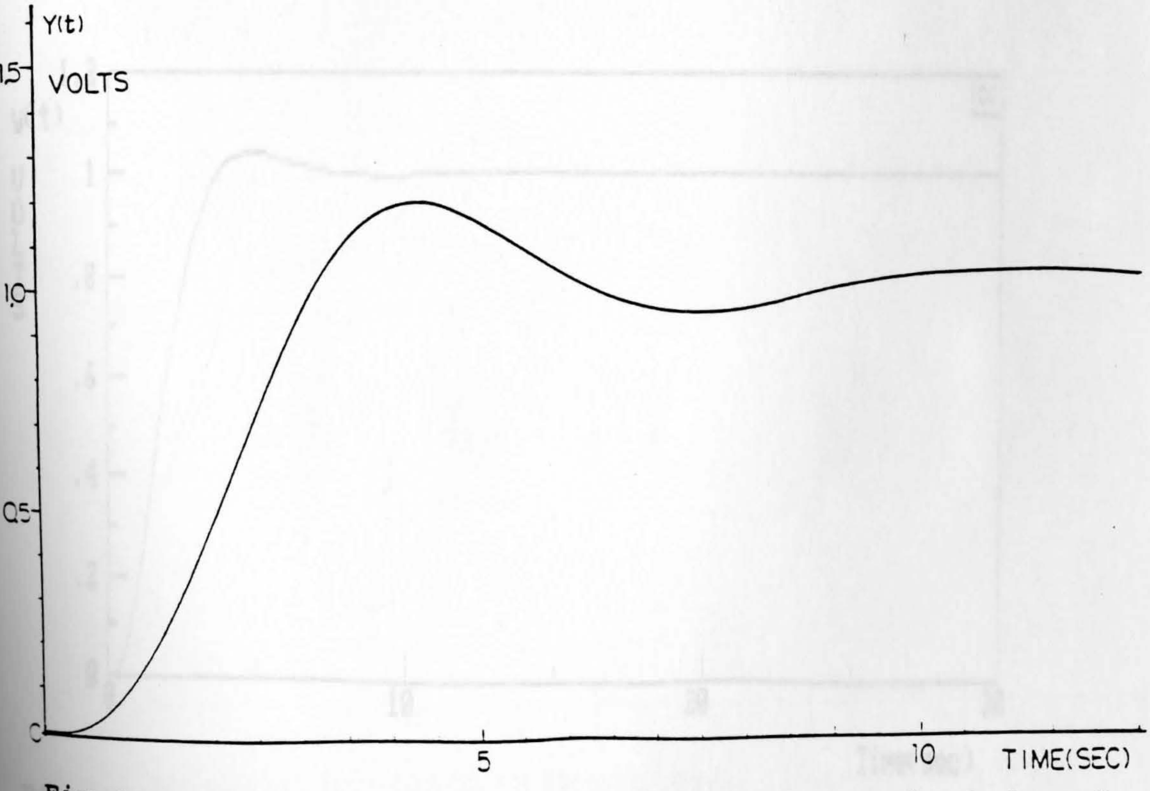

Fig.4.61 Real-time plant output signal of Example 3 Case 3

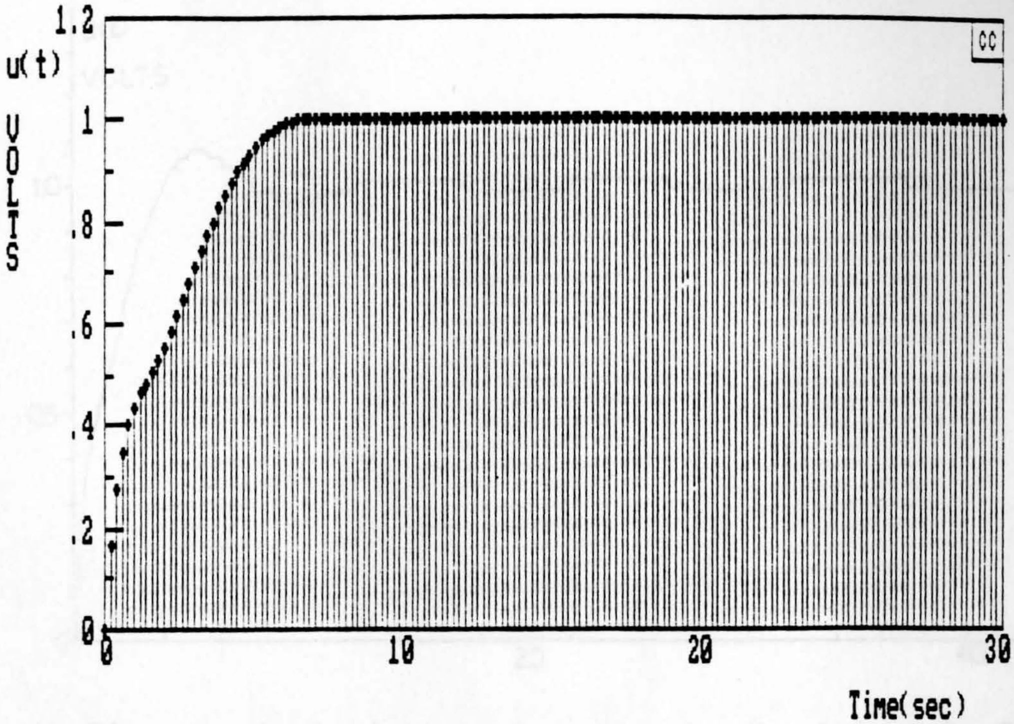

Fig.4.62 Simulated plant input signal of Example 3 Case 4

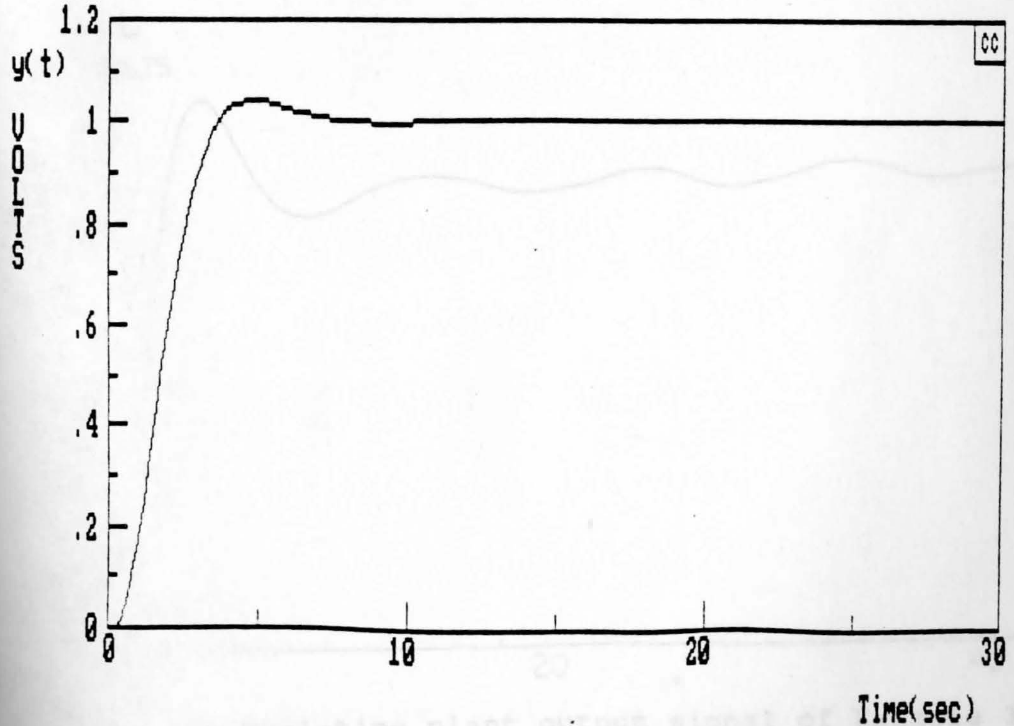

Fig.4.63 Simulated plant output signal of Example 3 Case 4

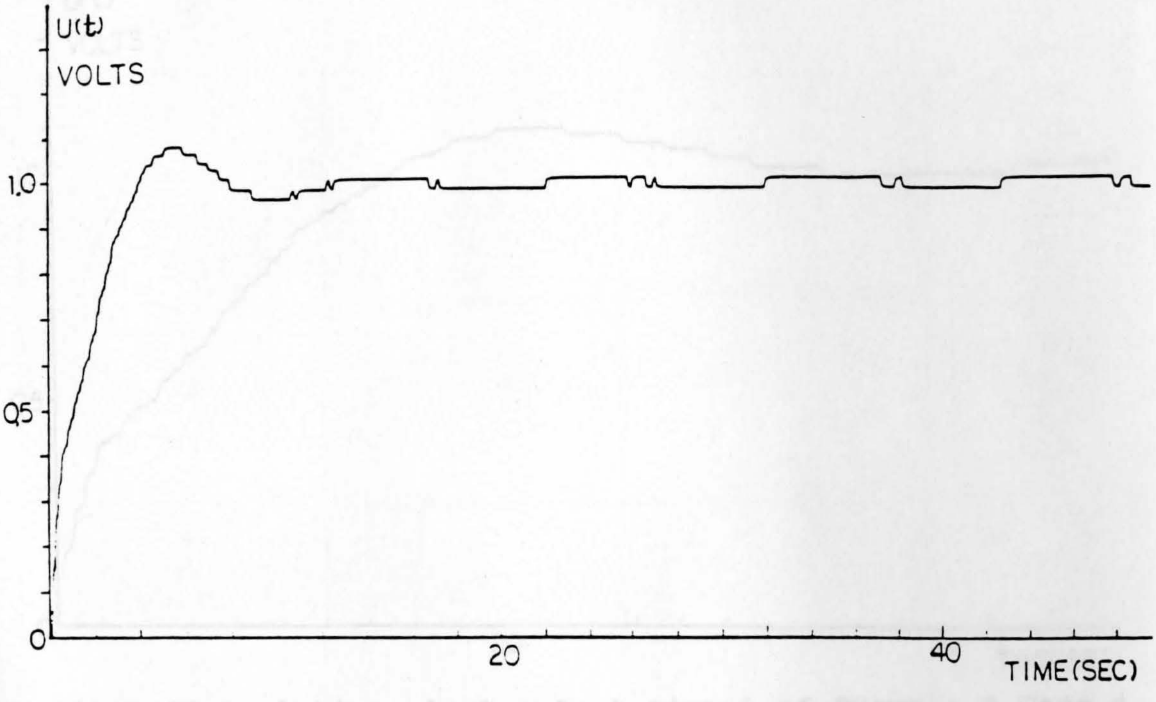

Fig.4.64 Real-time plant input signal of Example 3 Case 4

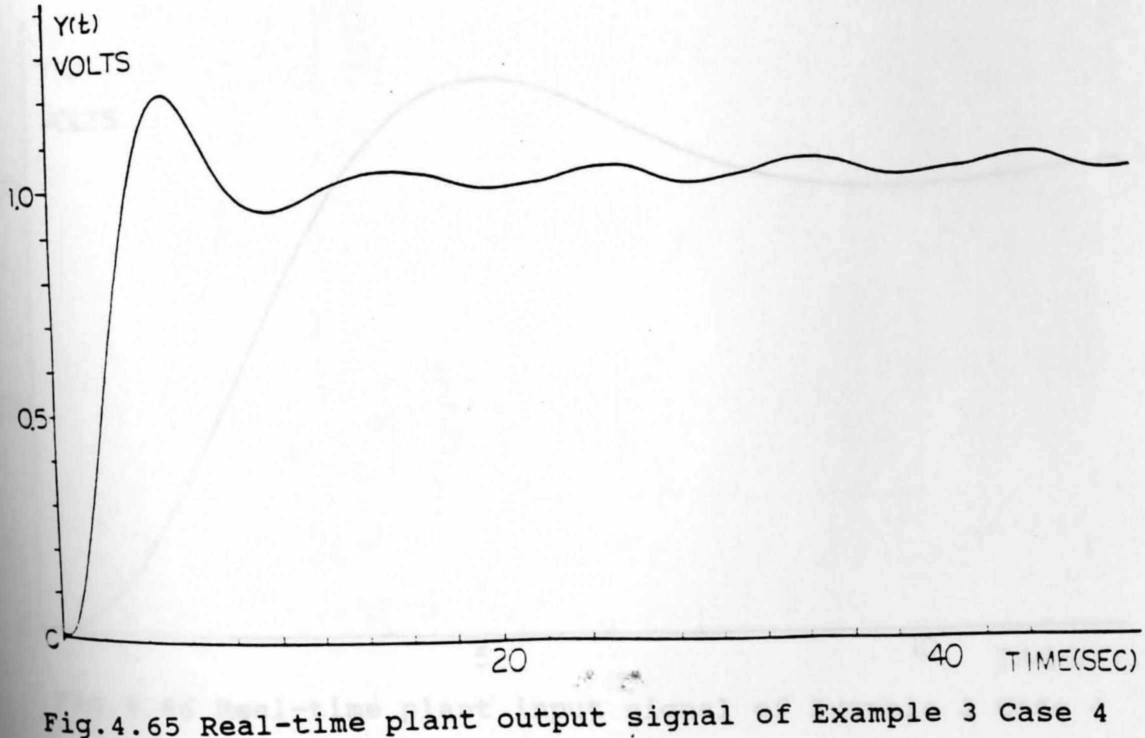

Fig.4.65 Real-time plant output signal of Example 3 Case 4

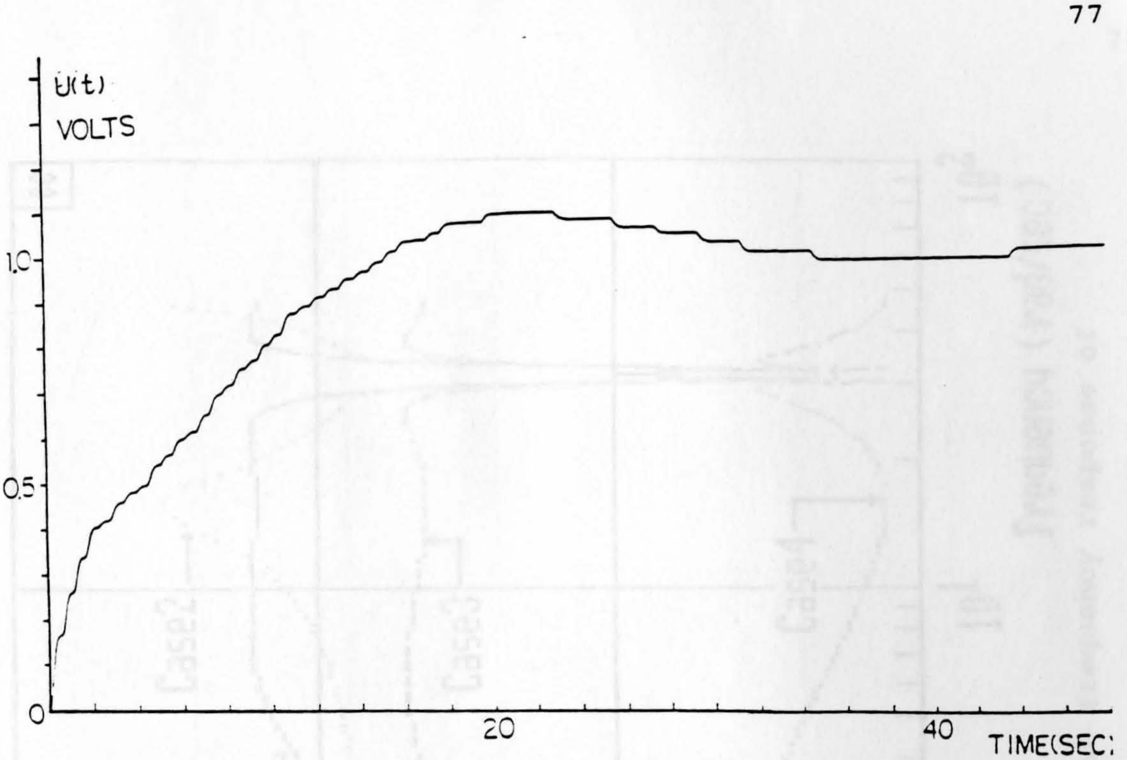

Fig.4.67 Real-time plant output signal of Example 3 Case 4

Fig.4.66 Real-time plant input signal of Example 3 Case 4

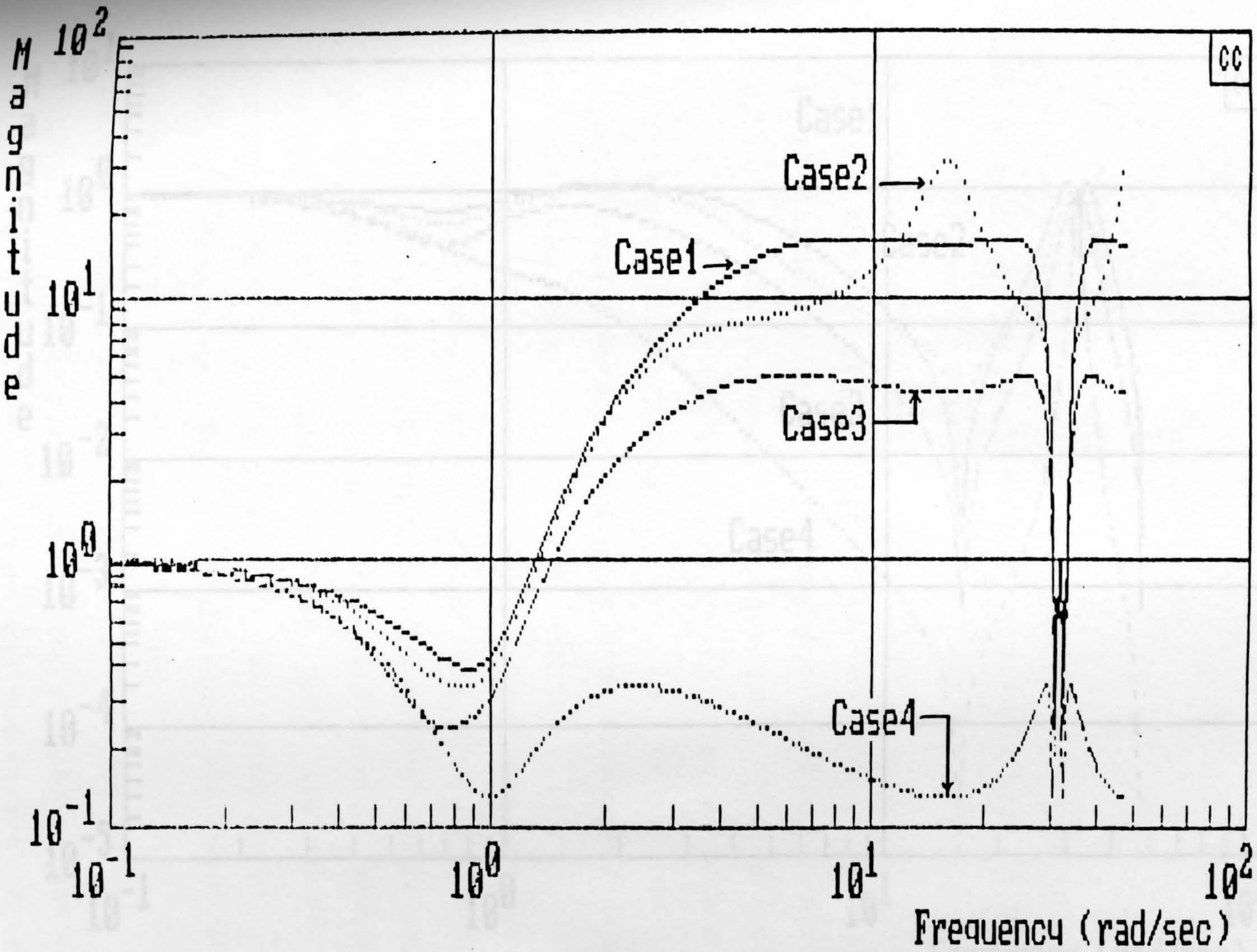

Figure 4.68 Noise-to-plant input frequency response of Example 3

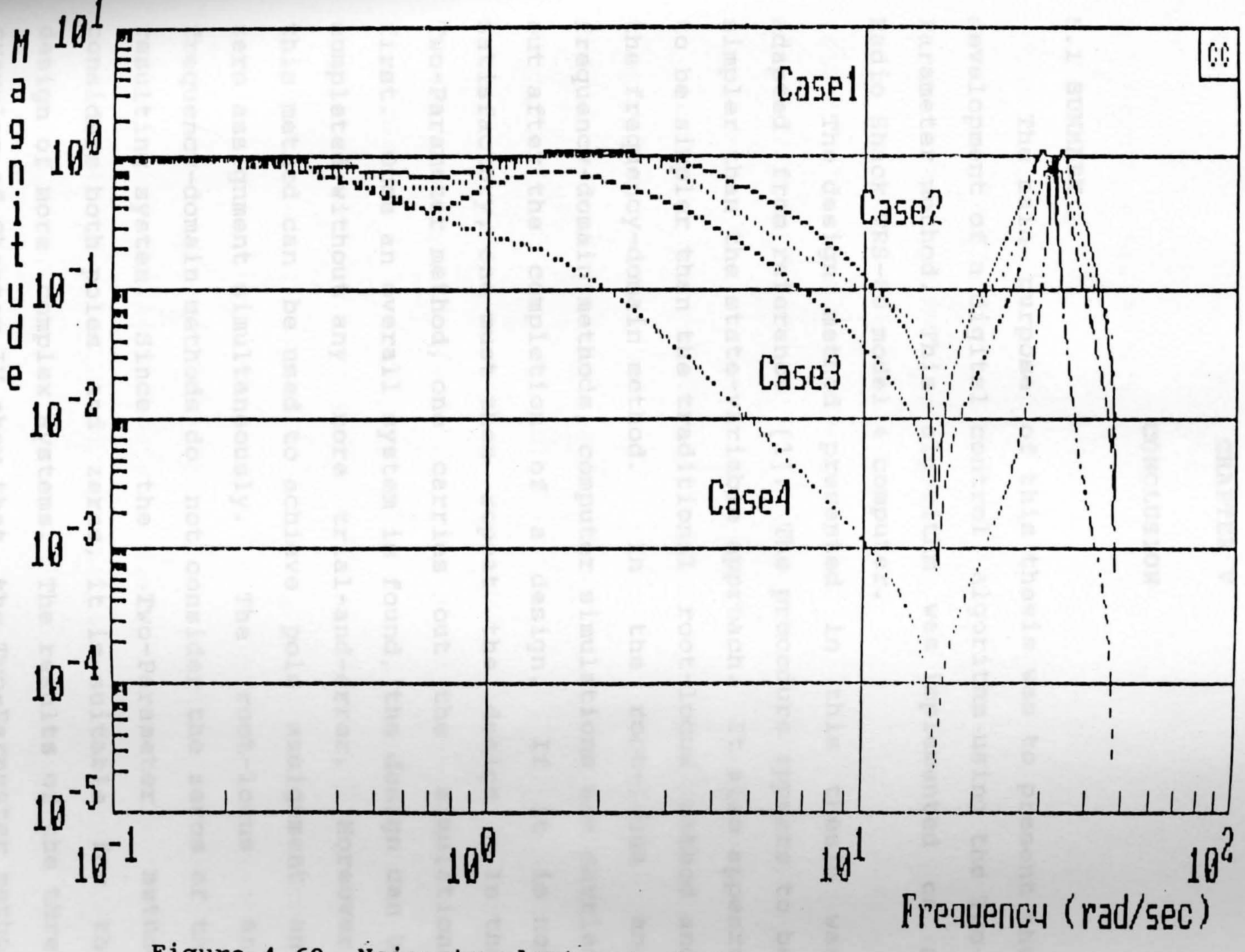

Figure 4.69 Noise-to-plant output frequency response of Example 3

CHAPTER V

CONCLUSION

5.1 SUMMARY

The main purpose of this thesis was to present the development of a digital control algorithm using the Two-Parameter method. This algorithm was implemented on a Radio Shack TRS-80 model 4 computer.

The design method presented in this thesis was adapted from reference [1]. The procedure appears to be simpler than the state-variable approach. It also appears to be simpler than the traditional root-locus method and the frequency-domain method. In the root-locus and frequency-domain methods, computer simulations are carried out after the completion of a design. If it is not satisfactory, one must then repeat the design. In the Two-Parameter method, one carries out the simulations first. Once an overall system is found, the design can be completed without any more trial-and-error. Moreover, this method can be used to achieve pole assignment and zero assignment simultaneously. The root-locus and frequency-domain methods do not consider the zeros of the resulting system. Since the Two-Parameter method considers both poles and zeros, it is suitable for the design of more complex systems. The results of the three examples of Chapter IV show that the Two-Parameter method is simple and straightforward. Although the pole-and-zero assignment is arbitrary, the results of the three examples show how the assignment affects the noise reduction.

5.2 RECOMMENDATIONS FOR FUTURE WORK

To improve the results of the digital controller presented in this thesis, a faster computer can be used to make the sampling period as short as possible. In fact, making the sampling period shorter and shorter tends to make the system behave more like the continuous-time system. To reduce the quantization error in A/D and D/A converters, A/D and D/A converters with higher resolution are also needed.

For noise rejection, a prefilter can be placed before the analog plant. The study of the effects of a prefilter bandwidth is presented in the reference [12]. In the digital case, the prefilter may also be implemented digitally. However, it would probably be sampled at a faster rate than the rate at which the controller operates This results in a multirate design in which the control signal is updated at the faster rate, while the output is sampled at the slower rate. See [17] for one design of such a controller. The opposite multirate sampling has also been considered [13]; here the control is updated at the slower rate and the output is sampled at the faster rate. Both types of multirate designs could be considered in conjunction with the Two-Parameter method.

APPENDIX A

The Two-Parameter design procedure, which discussed in Chapter II and Chapter IV, can be accomplished by using the following BASIC program.

```
10  'ENTER THE COEFFICIENTS FOR THE EQUATION G(z) and
20  'Go(z).
30  'where              b1*z+b0
40  '          G(z)=--------------   is open-loop transfer
50  '               z^2+a1*z+a0      function
70  '
80  '
90  '                      kz+z0
100 '     Go(z)=----------------------------  is closed-loop
110 '           z^3+a'2*z^2+a'1*z +a'0        transfer function
120 '
130 KEY OFF:COLOR 14,1,1:CLS
140 DATA C=A/,C=C*B,END
141 '
142 'ENTER THE COEFFICIENTS FOR EACH EQUATION
143 '
150 INPUT "Input from keyboard --- ",A$
160 IF A$="Y" OR A$="y" THEN 180
180 INPUT "Input b1 --- ",B1
190 INPUT "Input b0 --- ",B0
210 INPUT "Input a1 --- ",A1
220 INPUT "Input a0 --- ",A0
230 INPUT "Input a'2--- ",AP2
240 INPUT "Input a'1--- ",AP1
250 INPUT "Input a'0--- ",AP0
260 INPUT "Input d1 --- ",D1
261 '
262 'input the data to matrix a
263 '
280 OPEN "A.MAT" FOR OUTPUT AS #1
290 PRINT #1,5,5
300 PRINT #1,A0:PRINT #1,B0:PRINT #1,0 :PRINT #1,0 :PRINT
    #1,0
310 PRINT #1,A1:PRINT #1,B1:PRINT #1,A0:PRINT #1,B0:PRINT
    #1,0
320 PRINT #1,1 :PRINT #1,0 :PRINT #1,A1:PRINT #1,B1:PRINT
    #1,A0
330 PRINT #1,0 :PRINT #1,0 :PRINT #1,1 :PRINT #1,0 :PRINT
    #1,A1
340 PRINT #1,0 :PRINT #1,0 :PRINT #1,0 :PRINT #1,0 :PRINT
    #1,1
```

```
360 CLOSE #1
361 '
362 ' enter the data for matrix b
363 '
370 OPEN "B.MAT" FOR OUTPUT AS #1
380 PRINT #1,5,1
390 PRINT #1,D1*AP0
400 PRINT #1,AP0+AP1*D1
410 PRINT #1,AP1+AP2*D1
420 PRINT #1,AP2+D1
430 PRINT #1,1
450 CLOSE #1
455 '
460 REM ----- MATRIX OPERATION -----
465 '
470 ON ERROR GOTO 2140
480 DEFINT I,J,K
490 READ AA$
500 IF AA$="END" OR AA$="end" THEN 2560
510 GOSUB 540
520 GOTO 490
530 '----------------------
540 N=LEN(AA$)
550 N1=INSTR(AA$,"="):IF N1>0 THEN 570
560 NUM=1:GOTO 720
570 N2=INSTR(AA$,"=-"):IF N2=0 THEN 590
580 NUM=3:GOTO 720
590 N2=INSTR(AA$,"'"):IF N2=0 THEN 610
600 NUM=4:GOTO 720
610 N2=INSTR(AA$,"/"):IF N2=0 THEN 630
620 NUM=5:GOTO 720
630 N2=INSTR(AA$,"+"):IF N2=0 THEN 650
640 NUM=6:GOTO 720
650 N2=INSTR(AA$,"-"):IF N2=0 THEN 670
660 NUM=7:GOTO 720
670 N2=INSTR(AA$,"*"):IF N2=0 THEN 690
680 NUM=8:GOTO 720
690 N2=INSTR(AA$,";"):IF N2=0 THEN 710
700 NUM=8:GOTO 720
710 NUM=2
720 IF N1=0 THEN A1$=AA$ :A2$="" :A3$="" :GOTO 800
730 A3$=LEFT$(AA$,N1-1)
740 IF N2=0 THEN A1$=RIGHT$(AA$,N-N1) :A2$="" :GOTO 800
750 IF N2=N1 THEN A1$=RIGHT$(AA$,N-N2-1) :A2$="" :GOTO 800
760 A1$=MID$(AA$,N1+1,N2-N1-1)
770 IF N2=N THEN A2$="" :GOTO 800
780 A2$=RIGHT$(AA$,N-N2)
790 IF NUM=8 AND VAL(A1$)><0 THEN NUM=9 :GOTO 870
800 OPEN A1$+".MAT" FOR INPUT AS #1
810 INPUT #1,I1,J1
820 DIM GAUA(I1,J1)
830 FOR I=1 TO I1
840   FOR J=1 TO J1 :INPUT #1,GAUA(I,J) :NEXT J
850 NEXT I
860 CLOSE #1
```

```
870 IF A2$="" THEN 950
880 OPEN A2$+".MAT" FOR INPUT AS #1
890 INPUT #1,I2,J2
900 DIM GAUB(I2,J2)
910 FOR I=1 TO I2
920   FOR J=1 TO J2 :INPUT #1,GAUB(I,J) :NEXT J
930 NEXT I
940 CLOSE #1
950 P=0
960 ON NUM GOSUB
    1130,1230,1300,1370,1440,1720,1800,1880,1990
970 IF P=1 THEN 1090
980 OPEN A3$+".MAT" FOR OUTPUT AS #1
990 PRINT #1,I3,J3
1000 PRINT "Matrix ";A3$;" is:"
1010 FOR I=1 TO I3
1020   FOR J=1 TO J3
1030     PRINT USING "  ##.###^^^^";GAUC(I,J);
1040     PRINT #1,GAUC(I,J)
1050   NEXT J
1060   PRINT:PRINT
1070 NEXT I
1080 CLOSE #1
1090 ERASE GAUA
1100 IF A2$><"" THEN ERASE GAUB
1110 IF A3$><"" THEN ERASE GAUC
1120 RETURN
1130 REM ----- ERASE A MATRIX -----
1140 FOR I=1 TO I1
1150   FOR J=1 TO J1 :PRINT USING "
       ##.###^^^^";GAUA(I,J); :NEXT J
1160   PRINT
1170 NEXT I
1180 PRINT :PRINT "Erase Matrix ";A1$; :INPUT " or not?
     ",A$
1190 PRINT
1200 IF A$="Y" OR A$="y" THEN KILL A1$+".MAT"
1210 P=1
1220 RETURN
1230 REM ----- EQUIVALENT -----
1240 I3=I1:J3=J1
1250 DIM GAUC(I3,J3)
1260 FOR I=1 TO I3
1270   FOR J=1 TO J3 :GAUC(I,J)=GAUA(I,J) :NEXT J
1280 NEXT I
1290 RETURN
1300 REM ----- NEGATIVE -----
1310 I3=I1:J3=J1
1320 DIM GAUC(I3,J3)
1330 FOR I=1 TO I3
1340   FOR J=1 TO J3 :GAUC(I,J)=-GAUA(I,J) :NEXT J
1350 NEXT I
1360 RETURN
1370 REM ----- TRANSPOSE -----
1380 I3=J1:J3=I1
1390 DIM GAUC(I3,J3)
```

```
1400 FOR I=1 TO I3
1410   FOR J=1 TO J3 :GAUC(I,J)=GAUA(J,I) :NEXT J
1420 NEXT I
1430 RETURN
1440 REM ----- INVERSE MATRIX -----
1450 IF I1><J1 THEN 2090
1460 I3=I1:J3=J1
1470 DIM GAUC(I3,J3)
1480 FOR I=1 TO I3
1490   FOR J=1 TO J3 :GAUC(I,J)=0 :NEXT J
1500   GAUC(I,I)=1
1510 NEXT I
1520 FOR I=1 TO I1
1530   IF GAUA(I,I)><0 THEN 1610
1540   K=I+1:IF K>I1 THEN 1570
1550   IF GAUA(K,I)><0 THEN 1580
1560   K=K+1:IF K<=I1 THEN 1550
1570   PRINT :PRINT "No inverse exists." :PRINT :P=1
       :RETURN
1580   FOR J=1 TO J1
1590     GAUA(I,J)=GAUA(I,J)+GAUA(K,J)
       :GAUC(I,J)=GAUC(I,J)+GAUC(K,J)
1600   NEXT J
1610   X=1/GAUA(I,I)
1620   FOR J=1 TO J1 :GAUA(I,J)=GAUA(I,J)*X
       :GAUC(I,J)=GAUC(I,J)*X :NEXT J
1630   FOR K=1 TO I1
1640     IF K=I THEN 1690
1650     X=-GAUA(K,I)
1660     FOR J=1 TO J1
1670       GAUA(K,J)=GAUA(K,J)+GAUA(I,J)*X
         :GAUC(K,J)=GAUC(K,J)+GAUC(I,J)*X
1680     NEXT J
1690   NEXT K
1700 NEXT I
1710 RETURN
1720 REM ----- ADDITION -----
1730 IF I1><I2 OR J1><J2 THEN 2090
1740 I3=I1:J3=J1
1750 DIM GAUC(I3,J3)
1760 FOR I=1 TO I3
1770   FOR J=1 TO J3 :GAUC(I,J)=GAUA(I,J)+GAUB(I,J) :NEXT
         J
1780 NEXT I
1790 RETURN
1800 REM ----- SUBSTRACTION -----
1810 IF I1><I2 OR J1><J2 THEN 2090
1820 I3=I1:J3=J1
1830 DIM GAUC(I3,J3)
1840 FOR I=1 TO I3
1850   FOR J=1 TO J3 :GAUC(I,J)=GAUA(I,J)-GAUB(I,J) :NEXT
     J
1860 NEXT I
1870 RETURN
1880 REM ----- MULTIPLICATION -----
1890 IF J1><I2 THEN 2090
```

```
1900 I3=I1:J3=J2
1910 DIM GAUC(I3,J3)
1920 FOR I=1 TO I3
1930   FOR J=1 TO J3
1940     GAUC(I,J)=0
1950     FOR K=1 TO J1
     :GAUC(I,J)=GAUC(I,J)+GAUA(I,K)*GAUB(K,J)  :NEXT K
1960   NEXT J
1970 NEXT I
1980 RETURN
1990 REM ----- MULITPLY A CONSTANT -----
2000 M=VAL(A1$)
2010 I1=I2:J1=I2
2020 DIM GAUA(I1,J1)
2030 FOR I=1 TO I1
2040   FOR J=1 TO J1 :GAUA(I,J)=0 :NEXT J
2050   GAUA(I,I)=M
2060 NEXT I
2070 GOSUB 1880
2080 RETURN
2090 REM----- DIMENSION INCONSISTENCY -----
2100 BEEP
2110 PRINT "Dimension inconsistency."
2120 P=1
2130 RETURN
2140 REM ----- SUB FOR INPUT MATRIX -----
2150 CLOSE #1
2160 PRINT ERL
2170 IF ERL=800 THEN 2210
2180 IF ERL=880 THEN 2390
2190 BEEP
2200 PRINT "Something wrong!" :STOP
2210 IF NUM><1 THEN 2230
2220 PRINT "Matrix ";A1$;" is NOT exists." :PRINT :RESUME
     1210
2230 PRINT "Input Matrix ";A1$ :PRINT
2240 INPUT "# of row ---------- ",I1
2250 INPUT "# of column ------- ",J1
2260 DIM GAUA(I1,J1)
2270 FOR I=1 TO I1
2280   FOR J=1 TO J1
2290     PRINT A1$;"(";I;",";J; :INPUT ") = ",GAUA(I,J)
2300   NEXT J
2310 NEXT I
2320 OPEN A1$+".MAT" FOR OUTPUT AS #1
2330 PRINT #1,I1,J1
2340 FOR I=1 TO I1
2350   FOR J=1 TO J1 :PRINT #1,GAUA(I,J) :NEXT J
2360 NEXT I
2370 CLOSE #1
2380 RESUME 870
2390 PRINT "Input Matrix ";A2$ :PRINT
2400 INPUT "# or row ---------- ",I2
2410 INPUT "# or column ------- ",J2
2420 DIM GAUB(I2,J2)
2430 FOR I=1 TO I2
```

```
2440    FOR J=1 TO J2
2450      PRINT A2$;"(";I;",";J; :INPUT ") = ",GAUB(I,J)
2460    NEXT J
2470 NEXT I
2480 OPEN A2$+".MAT" FOR OUTPUT AS #1
2490 PRINT #1,I2,J2
2500 FOR I=1 TO I2
2510    FOR J=1 TO J2 :PRINT #1,GAUB(I,J) :NEXT J
2520 NEXT I
2530 CLOSE #1
2540 RESUME 950
2545 '
2550 '----calculate the value of Np----------------------
2550 '
2560 P=(1+AP2+AP1+AP0)/(B1+B0)
2570 PRINT
2580 PRINT "Np =";P
2581 '
2582 ' the polynomial L(z)
2583 '
2590 L2=P
2600 L1=P*D1
2610 L0=P*D0
2620 OPEN "C.MAT" FOR INPUT AS #1
2630 INPUT #1,A,A
2640 INPUT #1,A0:INPUT #1,M0:INPUT #1,A1:INPUT #1,M1:INPUT
     #1,A2
2650 CLOSE #1
2660 IF A2><1 THEN BEEP:BEEP
2661 '
2662 '--realization  of C1(z), C2(z)
2663 '
2670 PRINT
2680 PRINT "A11 =";0
2690 PRINT "A12 =";-A0
2700 PRINT "A21 =";1
2710 PRINT "A22 =";-A1
2720 PRINT
2730 PRINT "B11 =";L0
2740 PRINT "B12 =";-M0
2750 PRINT "B21 =";L1
2760 PRINT "B22 =";-M1
2770 PRINT
2780 PRINT "C1  =";0
2790 PRINT "C2  =";1
```

APPENDIX B

Following is a listing of program for the digital controller used in Example 1 and Example 2.

```
*This program is written for plant G(s)=(s-1)/s(s+2)
*ADDRESS 04 FOR PORT A AS INPUT PORT
*ADDRESS 05 FOR PORT B AS OUTPUT PORT
*ADDRESS 06 FOR PORT C AS CLOCK INPUT PORT
*ADDRESS 07 FOR CONTROL PORT
*Sampling time=0.2
10 CLS
20 OUT 236,16
30 REM---ENABLE 8255A
35 OUT 7,153
40 A12=-0.7614    :A22=6.834608
60 B11=0.15632    :B12=-0.28649
70 B21=-0.190932 :B22=0.06313
100 INPUT"ENTER REFERENCE VALUE (-2.5 TO 2.5);R
110 NBITS=8:VOLTS=5
120 D2A=2^NBITS/VOLTS:A2D=VOLTS/2^NBITS
130 X1=0:X2=0:REM INITIAL COMPENSATOR STATE
135 INPUT"ENTER INITIAL VALUE OF CONTROL (-2.5 TO 2.5)
140 UA=UA+2.5:REM SCALE CONTROL SIGNAL FOR D/A
150 U%=D2A*UA
160 OUT 5,U%
170 U%=D2A*(X2+2.5)
180 IF U%>255 THEN U%=255:IF U%<0 THEN U%=0
190 INPUT"HIT A KEY TO START";S$
200 C%=INP(6)
210 IF C%=0 THEN GOTO 200
220 REM  THE LOOP STARTS HERE
230 OUT 5,U%
240 Y%=INP(4):Y=A2D*Y%-2.5:REM  SCALE OUTPUT FROM A/D
250 NX1=A12*X2+B11*R+B12*Y
260 NX2=X1+A22*X2+B21*R+B22*Y
270 X1=NX1:X2=NX2
280 U%=D2A*(X2+2.5)
290 PRINT U%
300 IF U%>255 THEN U%=255
310 IF U%<0 THEN U%=0
320 C%=INP(6)
330 IF C%=0 THEN PRINT "TOO SHORT"
340 GOTO 200
```

APPENDIX C

Following is a listing of program for the digital controller used in Example 3.

```
*This program is written for plant G(s)=1/(S^2+0.5S+1)
*ADDRESS 04 FOR PORT A AS INPUT PORT
*ADDRESS 05 FOR PORT B AS OUTPUT PORT
*ADDRESS 06 FOR PORT C AS CLOCK INPUT PORT
*ADDRESS 07 FOR CONTROL PORT
*Sampling time=0.2
10 CLS
20 OUT 236,16
30 REM---ENABLE 8255A
35 OUT 7,153
40 A13=0.589496  :B11=0        :B12=-28.5964
60 A23=0.15632   :B21=0        :B22=61.4775
70 A33=-0.190932 :B31=0.91531  :B32=-35.4379
100 INPUT"ENTER REFERENCE VALUE (-2.5 TO 2.5);R
110 NBITS=8:VOLTS=5
120 D2A=2^NBITS/VOLTS:A2D=VOLTS/2^NBITS
130 X1=0:X2=0:REM INITIAL COMPENSATOR STATE
135 INPUT"ENTER INITIAL VALUE OF CONTROL (-2.5 TO 2.5)
140 UA=UA+2.5:REM SCALE CONTROL SIGNAL FOR D/A
150 U%=D2A*UA
160 OUT 5,U%
170 U%=D2A*(X2+2.5)
180 IF U%>255 THEN U%=255:IF U%<0 THEN U%=0
190 INPUT"HIT A KEY TO START";S$
200 C%=INP(6)
210 IF C%=0 THEN GOTO 200
220 REM  THE LOOP STARTS HERE
230 OUT 5,U%
240 Y%=INP(4):Y=A2D*Y%-2.5:REM  SCALE OUTPUT FROM A/D
250 NX1=A13*X3+B11*R+B12*Y
260 NX2=X1+A23*X3+B21*R+B22*Y
265 NX3=X2+A33*X3+B31*R+B32*Y
270 X1=NX1:X2=NX2:X3=NX3
280 U%=D2A*(X3+2.5)
290 PRINT U%
300 IF U%>255 THEN U%=255
310 IF U%<0 THEN U%=0
320 C%=INP(6)
330 IF C%=0 THEN PRINT "TOO SHORT"
340 GOTO 200
```

REFERENCES

[1] Chi-Tsong Chen. Introduction to the Linear Algebraic Method for Control System Design. IEEE Control System Magazine. p.36-p.42, Oct. 1987

[2] Chi-Tsong Chen. Linear System Theory and Design. New York: Holt, Rinehart and Winston, 1984

[3] Gene H. Hostetter and Mohammed S. Santina. Rational Linear Algebraic Tracking Control System Design. IEEE Control System Magazine. August, 1988

[4] M. Vidyasagar. System Synthesis: A Factorization Approach. MIT Press, 1985

[5] G. H. Hostetter, Clement J. Savant, Jr. and Raymond T. Stefani. Design of Feedback Control System. New York: Holt, Rinehart, and Winston. 2nd ed,1988

[6] K. J. Astrom and B. Wittenmark. Computer Controlled Systems. Englewood Cliffs, N.J.: Prentice-Hall, 1984.

[7] Radio Shack. TRS-80 Model 4 Service Manual. Forth Worth, TX: Radio Shack, 1983.

[8] Texas Instruments. The TTL Data Book. Dallas, TX: Texas Instruments, 1981

[9] National. Data Conversion/Acquisition Databook. Santa Clara, CA: National, 1984.

[10] Katsuhiko Ogata. Discrete-Time Control Systems. Englewood Cliffs, N.J. : Prentice-Hall, 1987

[11] J. Gary Reid. Linear System Foundamentals. New York: McGraw-Hill Book Company, 1983.

[12] Gene F. Franklin and J. David Powell. Digital Control of Dynamic Systems. New York: Addison-Wesley Publishing Company., 1980.

[13] Tomomichi Hgiwara and Mituhiko Araki. Design of a Stable State Feedback Controller Based on the Multirate Sampling of the Plant Output. IEEE Transactions on Automatic Control. Sept., 1988

[14] Thompson Peter M. Program CC User's Guide. Version 3.0 Hawthorn, CA: System Technology, 1985.

[15] Bassil Ibrahim. Implementation of a Digital Control Algorithm with Multiplexed State Variable Feedback. Masters Thesis Youngstown State Univ., 1988.

[16] Nai-Chian Guo. **Comparison of Parameter Estimation Algorithms Implemented on a Personal Computer.** Masters Thesis Youngstown State Univ., 1988.

[17] Araki and Hagiwara. **Pole Assignment by Multirate Sampled-data output feedback.** Int. J. Control, vol.44 pp 1661-1673, 1986.