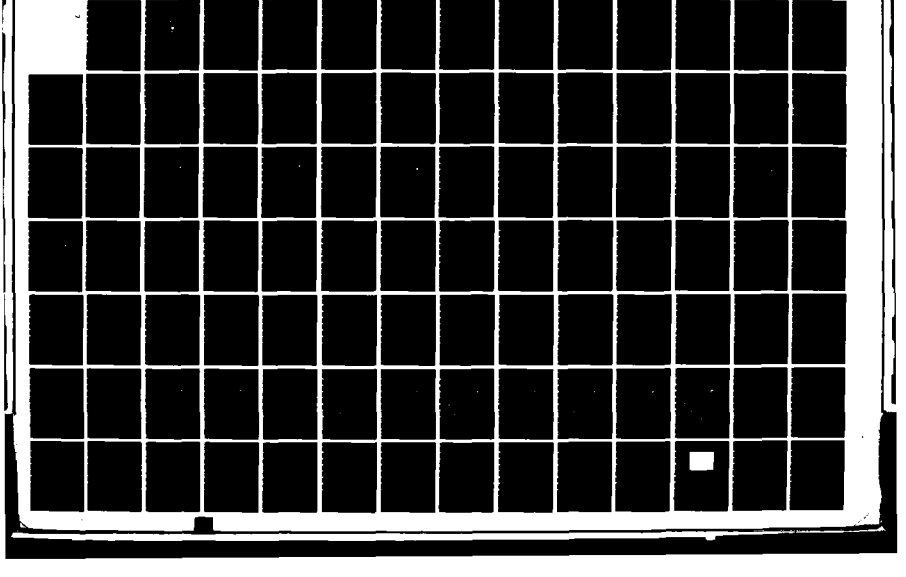
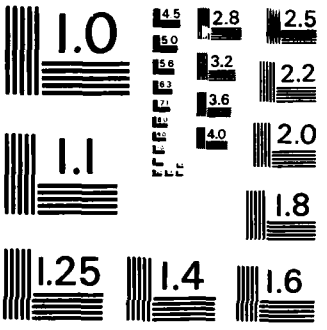
AD-A142 101 A DESIGN METHOD FOR A STATE FEEDBACK MICROCOMPUTER CONTROLLER OF A WIDE BANDWIDTH ANALOG PLANT(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA K C KIM DEC 83 1/2

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A DESIGN METHOD FOR A STATE FEEDBACK MICROCOMPUTER CONTROLLER OF A WIDE BANDWIDTH ANALOG PLANT

by

Ki Chul Kim

December 1983

Thesis Advisor: Alex Gerba Jr.

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A142101	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Design Method for a State Feedback Microcomputer Controller of a Wide Bandwidth Analog Plant.	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1983
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ki Chul Kim	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943	12. REPORT DATE December 1983
	13. NUMBER OF PAGES 111
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Microcomputer	Controller	Wide Bandwidth
State feedback	Transport lag	Compensation

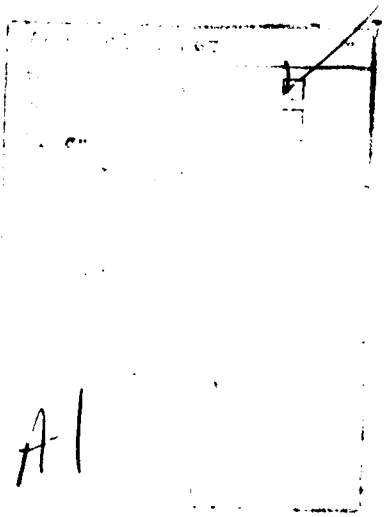
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In the design of a microcomputer regulator, continuous or discrete method can be applied. The objective of this thesis is to provide a continuous controller design method that can be used to compensate for the effect of the microcomputer transport lag. The compensation over the frequency range of interest yields a regulator response approximately equivalent to a direct analog feedback controller.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The technique uses state feedback method for development of the additionally compared to the discrete system design for the second order system numerical example. The method however is shown to be general enough to also apply to higher order system.



A-1

Approved for public release; distribution unlimited.

A Design Method for a State Feedback Microcomputer Controller of a Wide Bandwidth Analog Plant

by

Ki Chul Kim
Lieutenant Commander, Republic of Korea Navy
B.S., Republic of Korea Naval Academy, 1976
B.S.E.E., Korea University, 1980

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1983

Author: _____

Approved by: _____
Thesis Advisor

_____
Second Reader

_____
Chairman, Department of Electrical Engineering

_____
Dean of Science and Engineering

ABSTRACT

In the design of a microcomputer regulator, continuous or discrete method can be applied. The objective of this thesis is to provide a continuous controller design method that can be used to compensate for the effect of the microcomputer transport lag. The compensation over the frequency range of interest yields a regulator response approximately equivalent to a direct analog feedback controller. The technique uses state feedback method for development of the required phase compensation. The continuous system design is additionally compared to the discrete system design for the second order system numerical example. The method however is shown to be general enough to also apply to higher order system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. OVERVIEW

Most of the established design methods in control systems rely on the so-called fixed-configuration design in that the designer at the outset decides the basic composition of the overall system and the place where the controller is to be positioned relative to the components of the controlled process. The problem then involves the design of the elements of the controller. A majority of the design techniques in the modern control theory is based on the state feedback configuration. That is, instead of using controllers with fixed configurations in the forward or feedback path, control is achieved by feeding back the state variables through constant gains. In this case the state variables are fed back through constant gains to form the control. The problem with state feedback is that for high order systems the large number of state variables in practice would require a large number of transducers to sense for feedback. Thus, the actural implementation of the state feedback control scheme may be quite costly. A more serious problem is that not all the state variables in a given system may be directly accessible, and thus the need of an observer or estimator for the final implementation of state feedback [Ref. 1].

Usually, sampled-data systems refer to a more general class of systems whereas a digital control system refers to the use of a digital computer in the system. Since digital control systems have many advantages over continuous data systems, quite often, in practice, controllers that are designed in the analog domain are implemented digitally.

However, there are situations under which the controller of an existing system is analog, and the system already operates in a satisfactory fashion, but the availability and advantages of digital control suggest that the controller be implemented by digital elements [Ref. 2].

The flexibility is one of the advantages of a microprocessor. It is a key to increased productivity and energy efficiency in mechanical and process systems. Microcomputers provide the built-in intelligence which engineering systems must have in order to respond with maximum speed and efficiency to diverse and changing demands. The processing in microprocessor controllers is in real time and only fixed point two's complement arithmetic is used, consequently numbers are truncated. Furthermore, the major effort of the designer in practical implementation of digital controllers is concentrated around the subjects of finite word length, selection of sampling rate, programming of the algorithm, proper scaling of all variables and coefficients and selection of analog-to-digital and digital-to-analog converters [Ref. 3].

Digital control design can be accomplished via continuous design method or discrete design method. The continuous data control system design method is often more familier to control system designer than the discrete data control system design method. Because, throughout the years the analysis and design of continuous data control systems have been well developed, and pratically all these methods can be extended to digital control system.

The implementation of a microcomputer must take into account the important fact that there is a transport type lag from input to output of the microcomputer as well as other considerations such as word length, limit cycle, etc.. When the microprocessor is used as a controller for a high performance system where the time constants are very short

compared to the time of typical process control applications, the effect of the time to execute the control algorithm in a microprocessor on the control performance cannot be neglected. If one designs the control system by neglecting this processing time, the control system performance in this case never fails to become unsatisfactory. From the above mentioned point of view, the control algorithm may be complex and produce considerable delay in providing the control, since the control algorithm must treat the input and state variable constraints and the transportation lag simultaneously. The control algorithm which requires repeated computations or a long computation time, however, cannot be utilized because a resonable sampling period for the high performance system is too short to process a time-consuming algorithm during the sampling period [Ref. 2].

A fact that because very apparent early in the program development was the need for a cancellation factor to take into account the delays in the system due to calculations and conversions. It is possible to apply an attenuation in the controller output in the form of an exponential to correct for the effect of the transport lag, however, the bandwidth of the closed loop system becomes smaller [Ref. 4]. More improved compensators such as lead/lag network as well as state feedback technique are well known methods for providing the required phase shift cancellation. These methods usually require a high level of language for convenient implementation such as Basic and Fortran. The assumption to be used here is that only assembly language programming is available and state feedback will be applied to provide the required compensation.

Physical systems to be controlled often have large time constants and therefore the transport lag produced by the microcomputer controller becomes a problem only when the

control algorithm is lengthy such as in a nonlinear optimal estimation and control type problem. In this thesis, a wide bandwidth analog computer system is chosen as the plant so that the effect of transport lag can be an important consideration in the system design, that is, transport delay in the order of a few milliseconds can readily produce an unstable system.

B. THESIS OBJECTIVES

In the design of a microcomputer regulator, continuous or discrete method can be applied. The objective of this thesis is to provide a continuous controller design method that can be used to compensate for the effect of the microcomputer transport lag. The compensation over the frequency range of interest yields a regulator response approximately equivalent to a direct analog feedback controller. The technique uses state feedback method for development of the required phase compensation. The continuous system design is additionally compared to the discrete system design for the second order system numerical example. The method however is shown to be general enough to also apply to a higher order system.

II. THEORETICAL DEVELOPMENT

A. INTRODUCTION

In the development of the theory for microcomputer control, it is often not clear which procedure one should use to implement a compensator to negate the effect of the transport lag that results from the A/D, control algorithm computation and D/A delay times.

For state feedback applied to a regulator system, it is possible to provide the desired compensation within certain limitations using continuous system design technique.

In general, a linear plant can be represented by the state equations

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,u$$

Given that the order of the system is n, then the requirement is for a reduced state feedback of $k \leq (n-1)$ gains that will meet certain design specifications such as percent overshoot and time of first peak.

Now whenever a microcomputer is used to implement the above design, the additional factor of transport lag due to A/D, control algorithm computation and D/A delay times can, for high performance system, strongly affect the overall performance.

To take into account and reduce the effect of transport lag, it is possible to use (k+1) state feedback gains thereby providing a phase shift that will approximately cancel out the transport lag phase shift over the frequency range of interest.

A simple second order system is used to illustrate the method. Both real time application and computer simulation are used to verify the theory.

B. PRACTICAL DESIGN CONSIDERATIONS

The design considerations for a second order regulator system could be assumed as follow.

- ζ : 0.35 - 0.55 (percent overshoot : 15 - 35%)
- Bandwith : 50 - 70 Hz (300 - 440 rad/sec) for high performance control system.
- Peak overshoot time (tp) : 0.008 - 0.015 sec
- Natural frequency (ω_n) : 250 - 400 rad/sec
- Initial condition (or Disturbance) : below 5 V due to A/D and D/A conversion limitation in microcomputer

C. CONTINUOUS SYSTEM DESIGN METHOD

1. State Feedback Control System

A powerful design method in the state variable domain is the state feedback. In practice, not all the state variables are accessible, so an observer is used to estimate some or all the state variables. The state feedback control system is that, instead of using controllers with fixed configurations in the forward or feedback path, control is achieved by feeding back the state variables through constant gains. The typical second order system whose transfer function is represented by Equation 2.1 is considered in this section.

$$Gp(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)} \qquad \text{(eqn 2.1)}$$

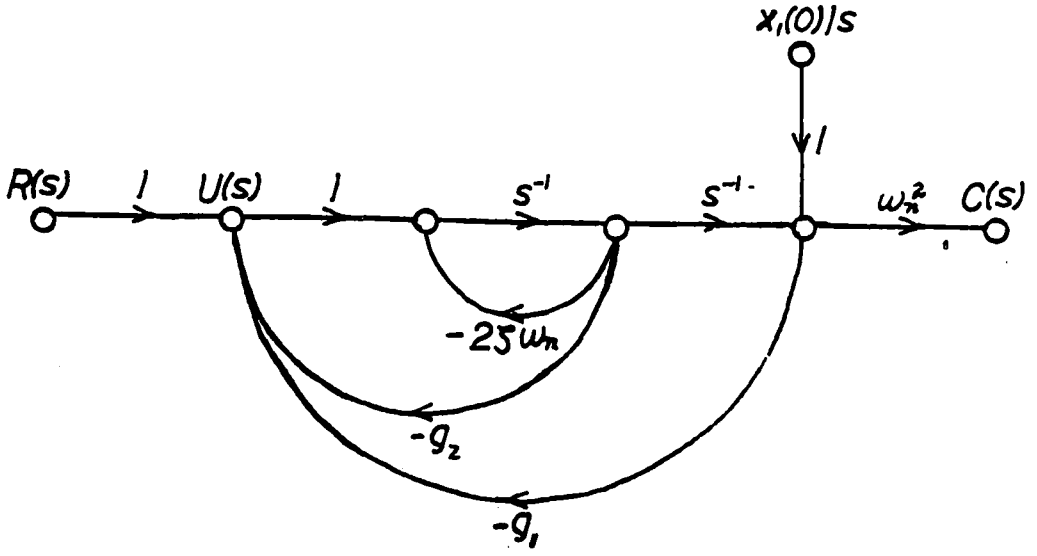


Figure 2.1 Signal Flow Diagram of State Feedback System.

If the states x_1 and x_2 are physically accessible, these variables may be fed back through constant gains g_1 and g_2, respectively, to form the control, as shown in Figure 2.1. The closed loop transfer function of the system is written

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + (2\zeta\omega_n + g_2)s + g_1} \qquad \text{(eqn 2.2)}$$

Here, it can be compared with a PD control system. For the system with PD control shown in Figure 2.2, the closed loop transfer function is

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + (2\zeta\omega_n + KaKd\omega_n^2)s + \omega_n^2 KaKp} \qquad \text{(eqn 2.3)}$$

Thus, the characteristic equations of the systems described by Equation 2.2 and Equation 2.3 would be identical if $g_1 = \omega_n^2 \cdot Ka \cdot Kp$ and $g_2 = \omega_n^2 \cdot Ka \cdot Kd$. If the reference

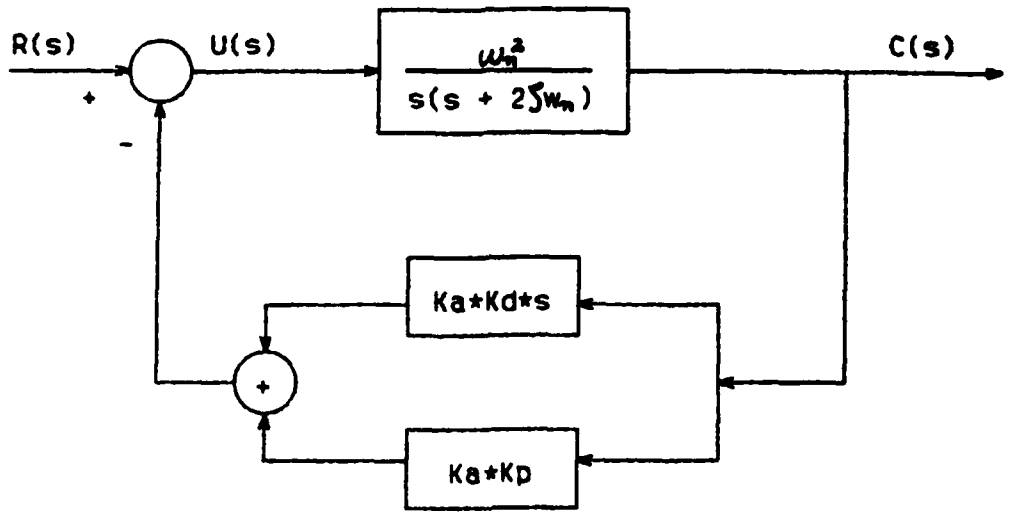


Figure 2.2 Feedback Control System with PD Control.

input r(t) is zero, the class of systems is commonly described as regulators.

Under this condition the control objective is to drive any arbitrary initial conditions of the system to zero as quickly as possible. Then the regulator system with the PD controller is the same as the state feedback controller.

2. Microcomputer Controller Design with Time Delay

The time delay is most important in the controller design for microcomputer controller, since the microcomputer requires time for processing of the control algorithm. In general, a closed loop system with time delay in the loop will be subject to more stability problems than systems

without time delays. In practice, pure time delays may be encountered in various type of systems. In these systems the output will not begin to respond to an input until after a given time interval. Since a pure time delay Td is modeled by the transfer function relationship e^{-TdS} the characteristics equation of the system will no longer have constant coefficients. The transfer function of control system with time delay is represented in Equation 2.4 .

$$G(s)H(s) = G_1(s)H_1(s) \cdot e^{-TdS} \qquad \text{(eqn 2.4)}$$

In order to apply the continuous system design method to design of microcomputer controller, the time delay must be considered in the system modeling. The procedure of the design of microcomputer controller using continuous control system design method will be described in the next section.

3. Design Procedure

The Equation 2.4 has a time delay factor as an exponential form. The effect of the exponential form is that it rotates the phasor $G_1(jw)H_1(jw)$ at each w by an angle of wTd radians in the clockwise direction. The amplitude of $G_1(jw)H_1(jw)$ is not affected by the time delay. Since, the magnitude of e^{-TdS} is unity for all frequencies. In principle, the stability of the closed loop system can be investigated by sketching the Nyquist locus of $G(jw)H(jw)$ and then observing its behavior with referance to the (-1, j0) point of the complex function plane. For the closed loop system to be stable, all the intersects of the $G(jw)H(jw)$ locus with the real axis must occur to the right of the (-1, j0) point.

The analysis and design problems involving pure time delays are more easily carried out graphically in the Bode diagram. Since the time delay term affects only the phase but not the magnitude of $G(jw)H(jw)$, the phase with time delay is obtained in the Bode plot by adding a negative angle of wTd to the phase curve of $G_1(jw)H_1(jw)$. The frequency at which the phase curve of $G(jw)H(jw)$ crosses the 180-degree axis is the place where the Nyquist locus intersects the negative real axis. Thus, in order to reduce the affect of transportation lag due to time delay, the phase lead type compensator should be chosen. The PD type controller is used as the phase lead type compensator. The state feedback controller and PD controller are used interchangablely in this work as described before. The PD controller transfer function is

$$H(s) = Ka(Kp + Kd \cdot s) \qquad \text{(eqn 2.5)}$$

where $Ka \cdot Kp$: Proportional gain

$Ka \cdot Kd$: derivative gain

Ka : Attenuation factor

In the microcomputer state feedback controller with single input, state x_2 is not measured and must be estimated. The estimation is performed using backward difference approximation method. If the sampling period becomes too large, then the difference of two adjacent state values also becomes too large. Because of this affect, the output of the controller is not available for the input of plant. This is the reason for applying the attenuation factor in the PD controller transfer function. The details will be described in Chapter IV.

The design procedure is general enough to apply to a high order system as demonstrated in Chapter III - Computer Simulation.

The general idea of this design technique is:

(1) The system is designed with less than full state feedback. A continuous system plant of order n is designed with $k \leq (n-1)$ state feedback gains to meet the desired system performance specification.

(2) The transportation delay from analog input to analog output is then estimated or computed to a good approximation. This delay Td is due to A/D and D/A as well as control algorithm computation time.

(3) The number of state feedback gain are selected equal to $m = (k+1)$ to provide a phase lead cancellation of the phase lag due to the transport delay within a reasonable approximation over the frequency range of interest.

A pure second order system is used for convenience, and the ζ and w_n are chosen for the numerical example as the following:

$\zeta = 0.5$

$w_n = 250$ rad/sec

These values are chosen to meet the desired design considerations.

Since the system is second order, n equals 2, then the number of less than full state feedback gain is 1. In this particular case, less than full state feedback control is the same as direct feedback control. We assumed that the

direct feedback control system performs to meet the desired performance. The transfer function of direct feedback control system without transport delay is written

$$G(s)H(s) = \frac{250}{s(1 + \frac{1}{250}s)} \qquad \text{(eqn 2.6)}$$

When the microcomputer is used for the controller, it must include the transport delay term. Then, Equation 2.6 is

$$G(s)H(s) = \frac{250}{s(1 + \frac{1}{250}s)} \cdot e^{-TdS} \qquad \text{(eqn 2.7)}$$

This transport type delay time can be calculated by using Equation 2.8 .

$$Td = T_{A/D} + T_{Ex} + T_{D/A} \qquad \text{(eqn 2.8)}$$

where Td : Transport delay time
$T_{A/D}$: A/D conversion time
T_{Ex} : Program execution time
$T_{D/A}$: D/A conversion time

For the cancellation of the phase lag due to the transport delay, the phase lead type compensation is needed, thus the number of state feedback gain m equals 2. Therefore, the transfer function of the compensator can be written

$$H(s) = Ka(Kp + Kd \cdot s) \qquad \text{(eqn 2.9)}$$

Finally, the closed loop system transfer function is

$$G(s)H(s) = \frac{250Ka(Kp + Kd \cdot s)}{s(1 + \frac{1}{250}s)} \cdot e^{-TdS} \qquad \text{(eqn 2.10)}$$

Using these transfer functions, the characteristics are analyzed by the Bode diagram and Nyquist plot. Figure 2.3 through Figure 2.18 show the graphical analysis which were obtained using standard IBM 3033 computer programs. Table I is a tabulation of the variation of gain margin and phase margin obtained from the computer output. Thus, the appropriate values of attenuation and other factors can be decided for a specific delay time using this table.

TABLE I

Summary of Analysis

Delay T (sec)	Ka	Kp	Kd	Gain Margin (dB)	Phase margin (Degree)
0.00045	1	1	0	16.73	46.73
	1	1	0	16.73	46.73
	1	1	T	19.13	46.73
	0.5	2	T	19.13	46.73
	0.25	4	T	19.13	46.73
	0.2	5	T	19.13	46.73
0.002	1	1	0	6.62	29.21
	1	1	T	6.62	29.21
	0.5	2	T	6.62	29.21
	0.25	4	T	6.62	29.21
	0.2	5	T	6.62	29.21
0.006	1	1	0	-5.63	-15.98
	1	1	T	-2.35	-15.98
	0.8	1	T	1.48	-2.79
	0.4	1	T	5.61	37.22
	0.2	1	T	11.63	61.98
	0.5	2	T	-2.35	-15.98
	0.25	4	T	-2.35	-15.98
	0.2	5	T	-2.35	-15.98
0.008	1	1	0	-4.63	-38.58
	1	1	T	-4.63	-38.58
	0.8	1	T	-2.69	-22.43
	0.4	1	T	3.33	26.48
	0.2	1	T	9.35	56.35
	0.5	2	T	-4.63	-38.58
	0.25	4	T	-4.63	-38.58
	0.2	5	T	-4.63	-38.58

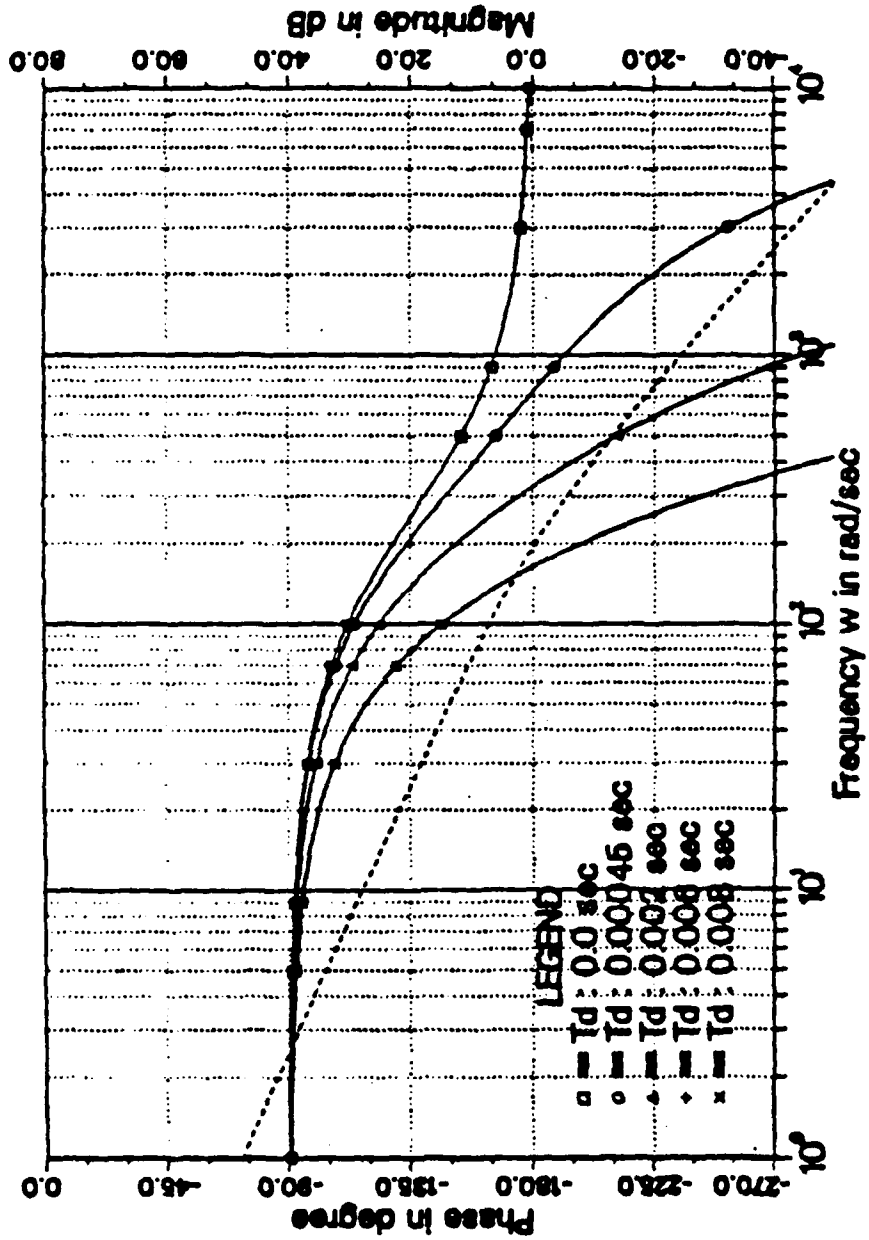


Figure 2.3 Bode Diagram of Eqn. 2.7.

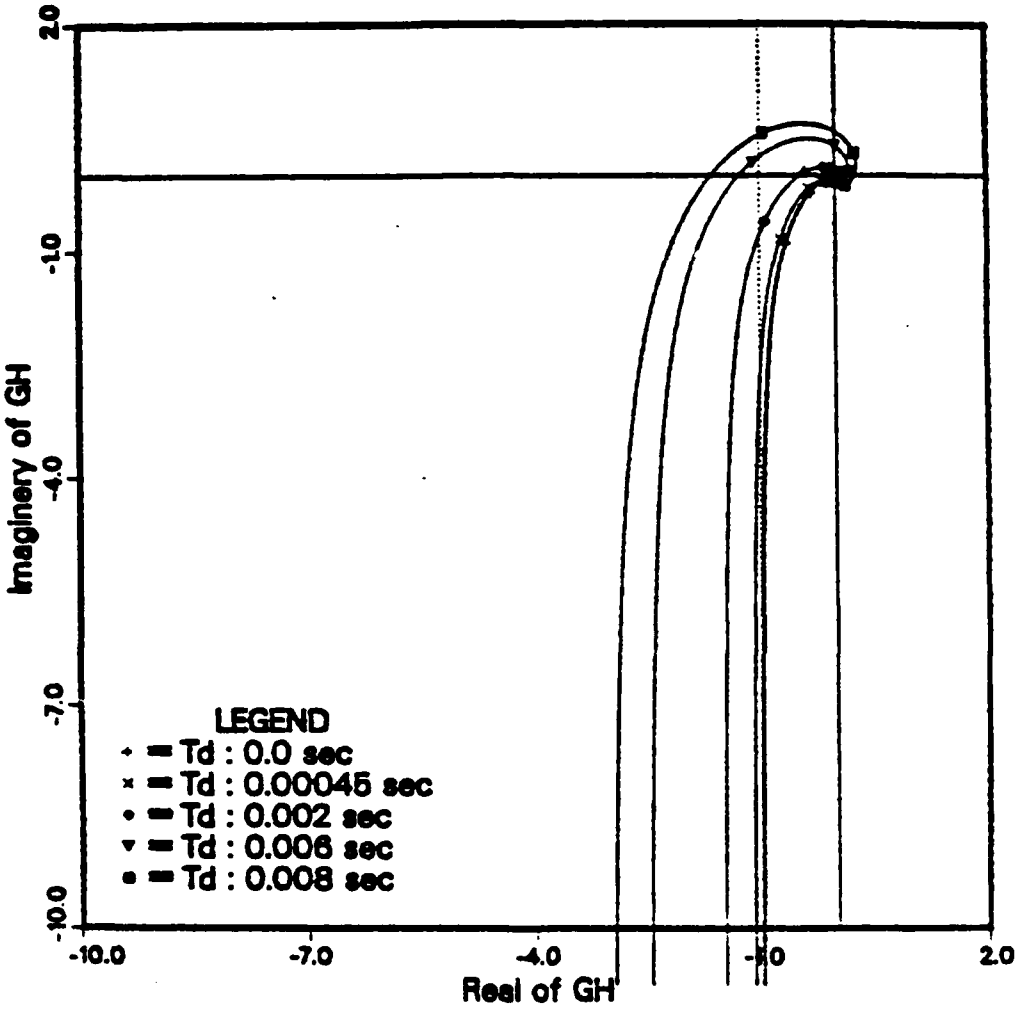


Figure 2.4 Nyquist Plot of Eqn. 2.7.

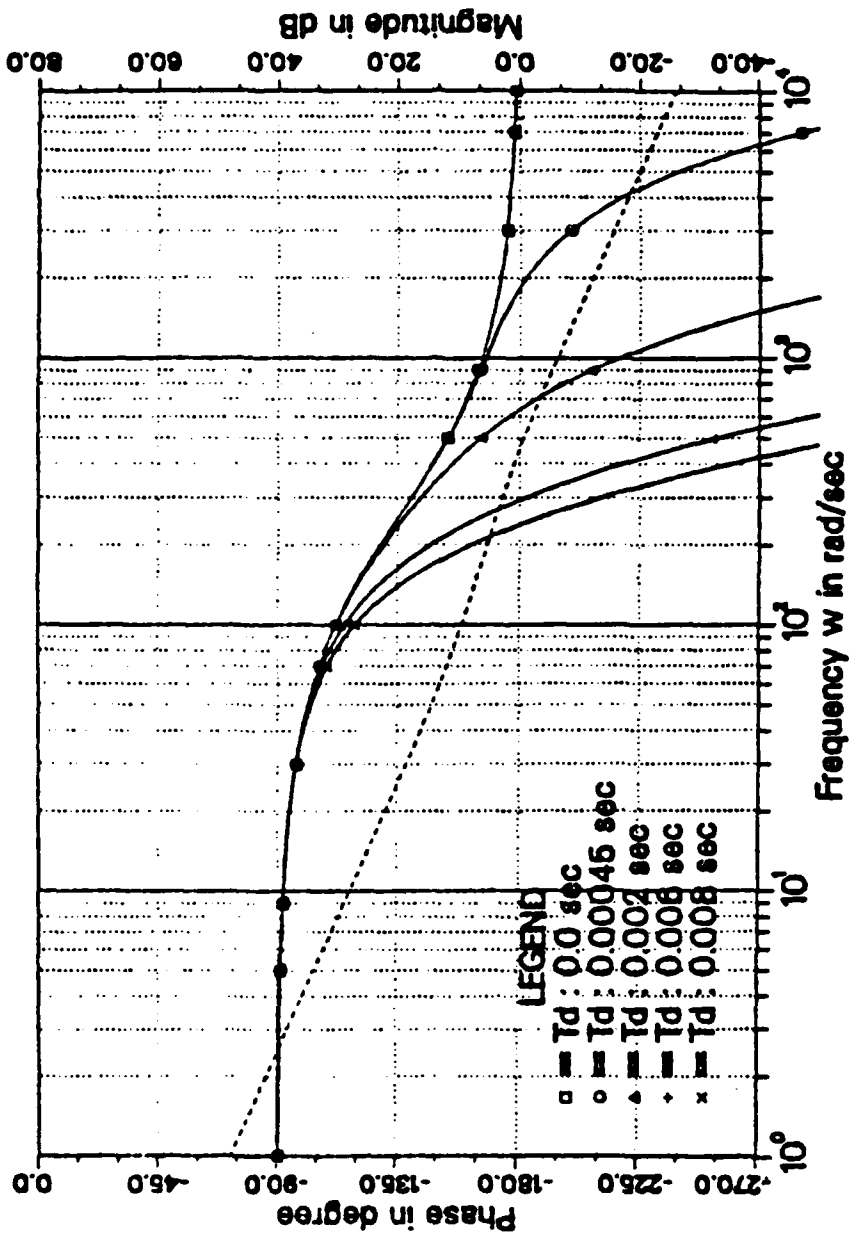


Figure 2.5 Bode Diagram of Eqn. 2.10 (Kp=1, Ka=1).

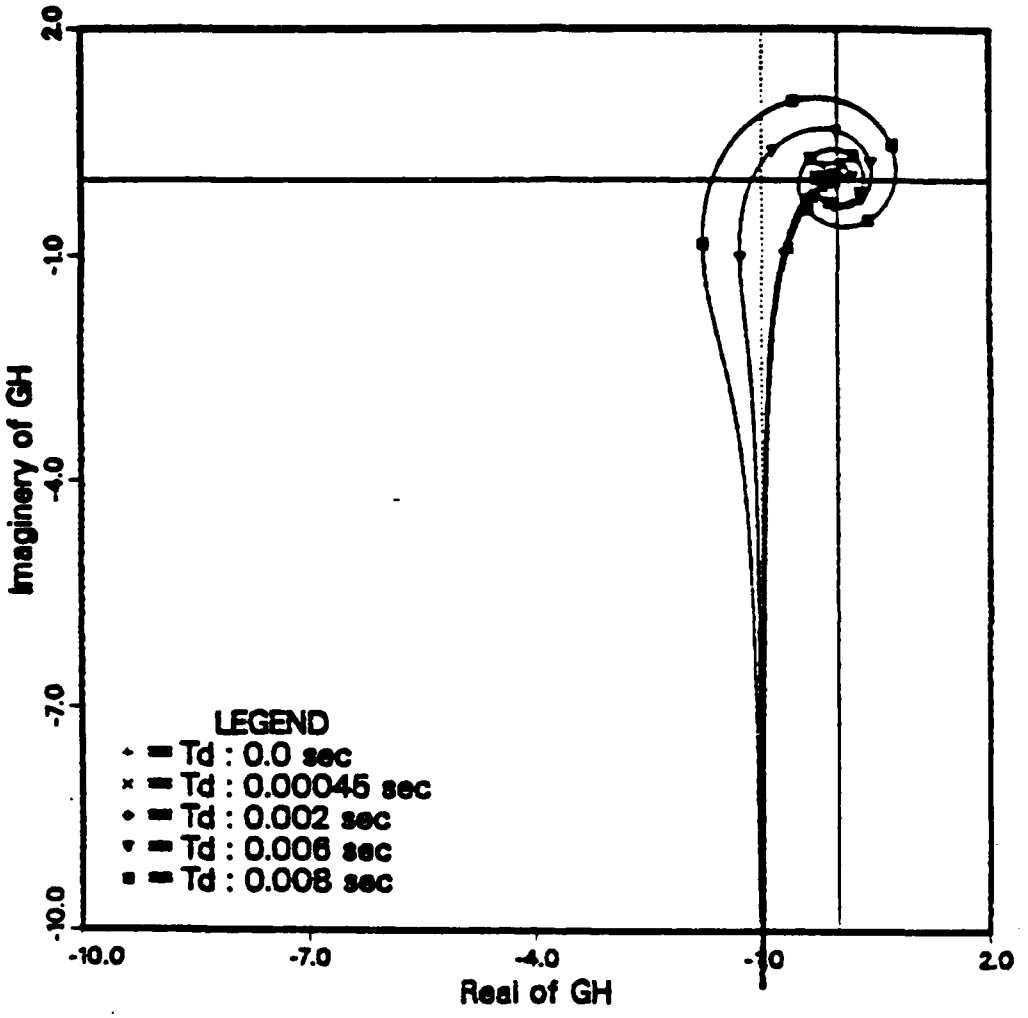


Figure 2.6 Nyquist Plot of Eqn. 2.10 (Kp=1, Ka=1).

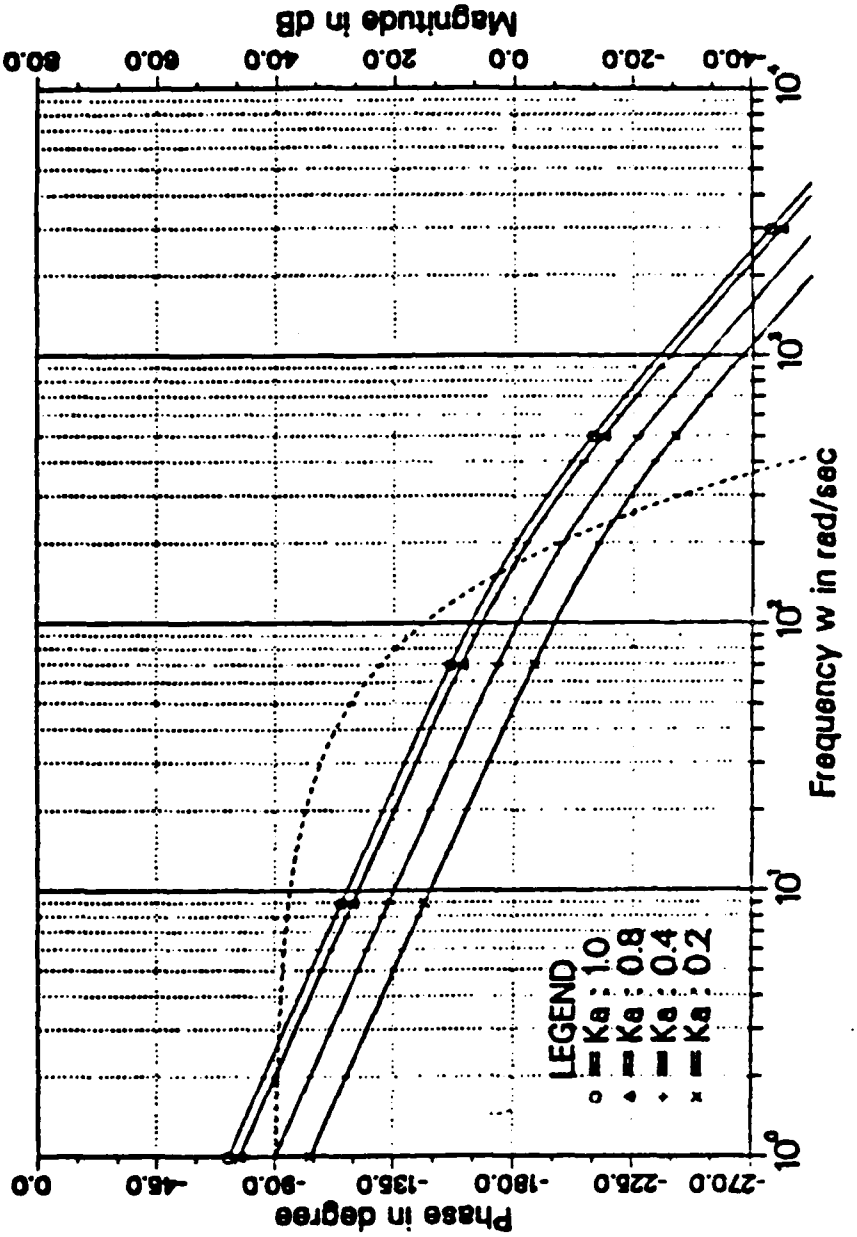


Figure 2.7 Bode Diagram (Td = 0.006 sec, Kp=1).

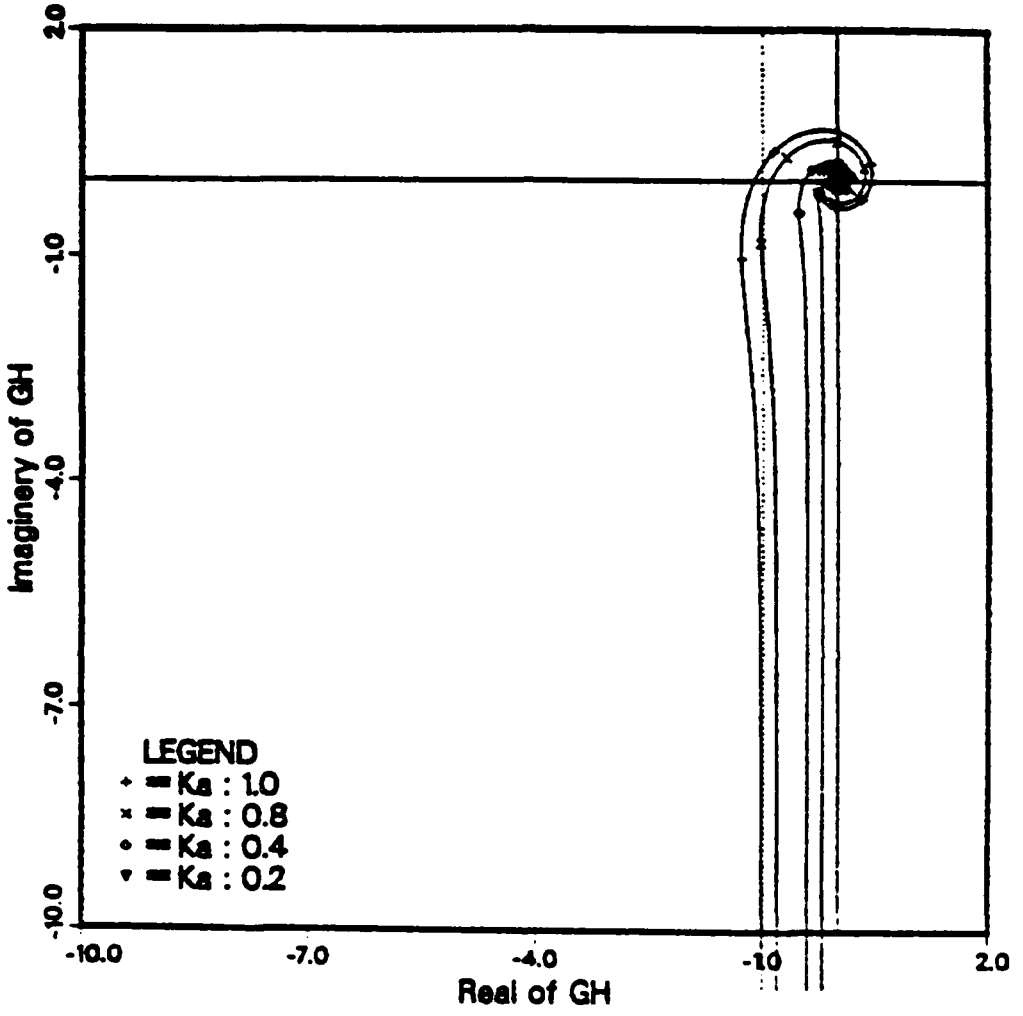


Figure 2.8 Nyquist Plot (Td = 0.006 sec, Kp=1).

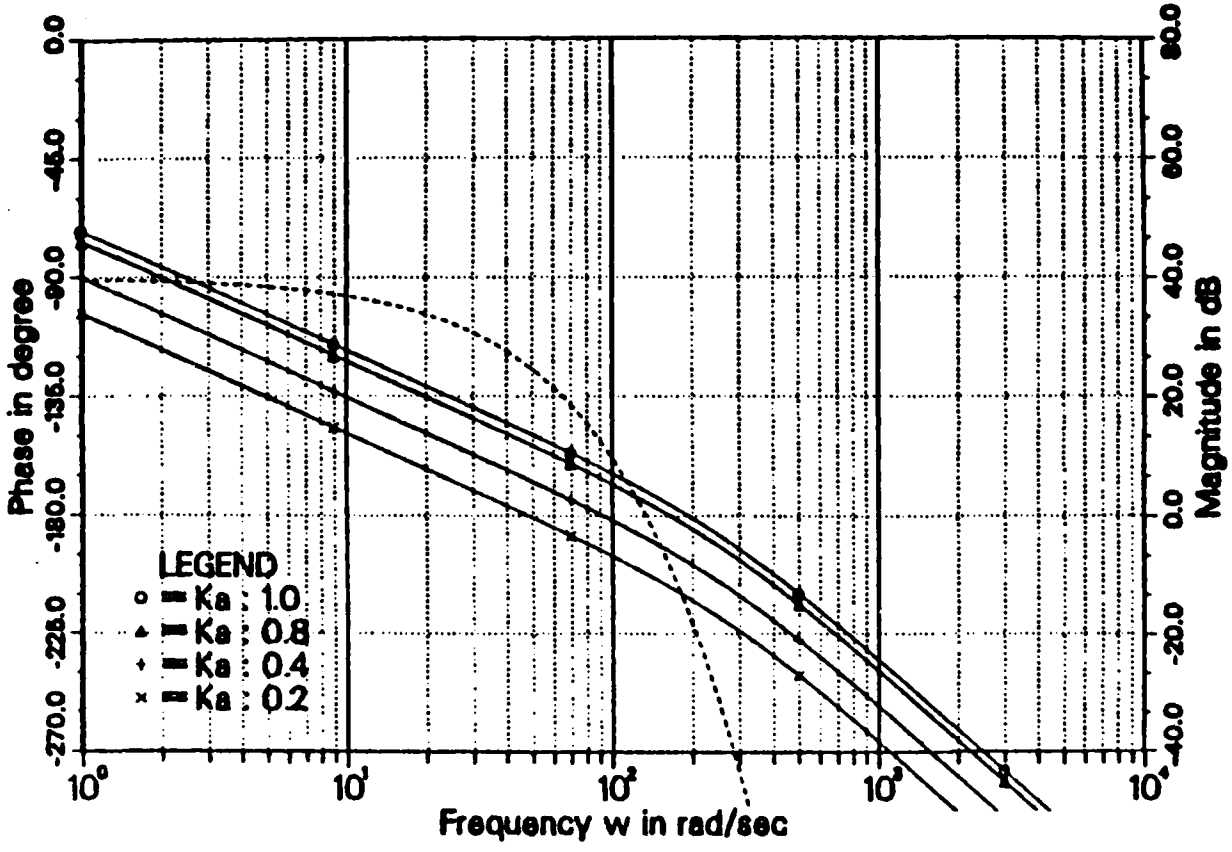


Figure 2.9 Bode Diagram (Td = 0.008 sec, Kp=1).

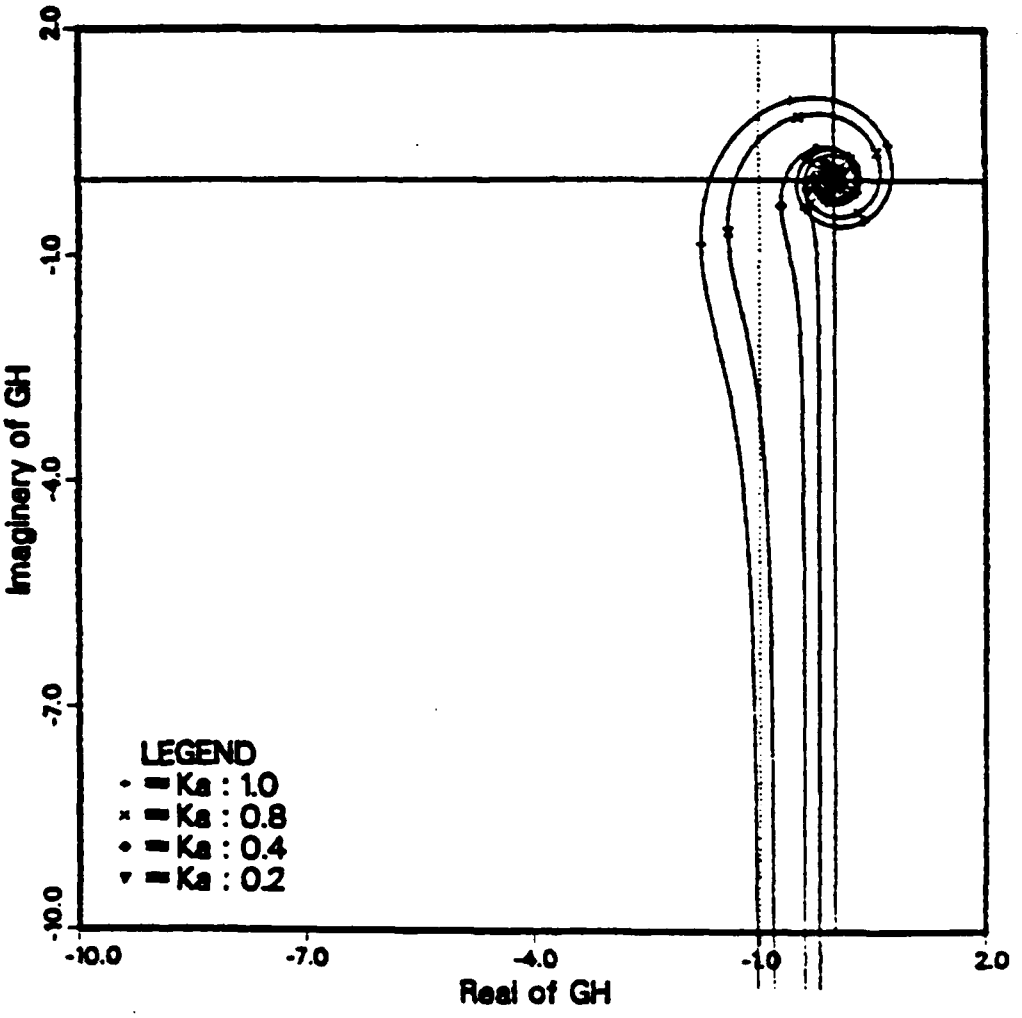


Figure 2.10 Nyquist Plot (Td = 0.008 sec, Kp=1).

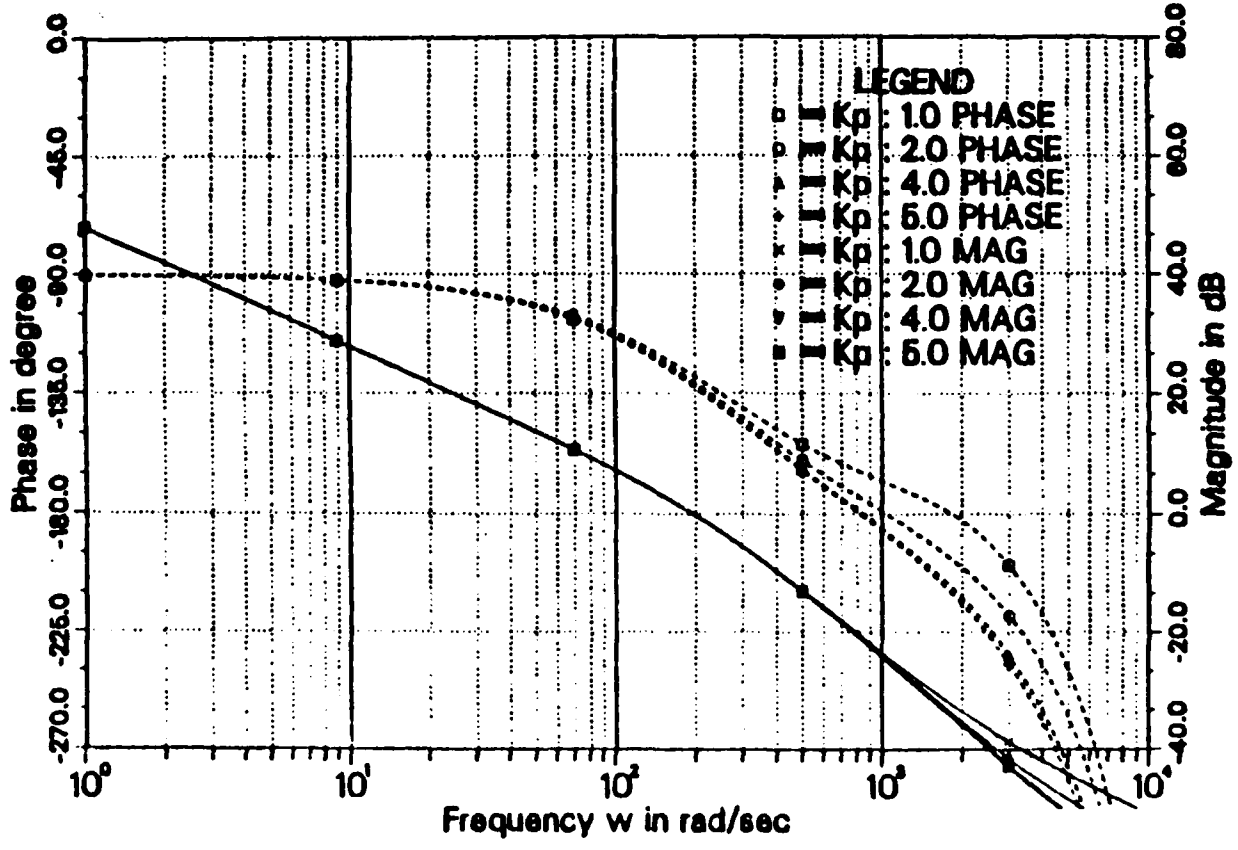


Figure 2.11 Bode Diagram (Td = 0.00045 sec, Ka•Kp=1).

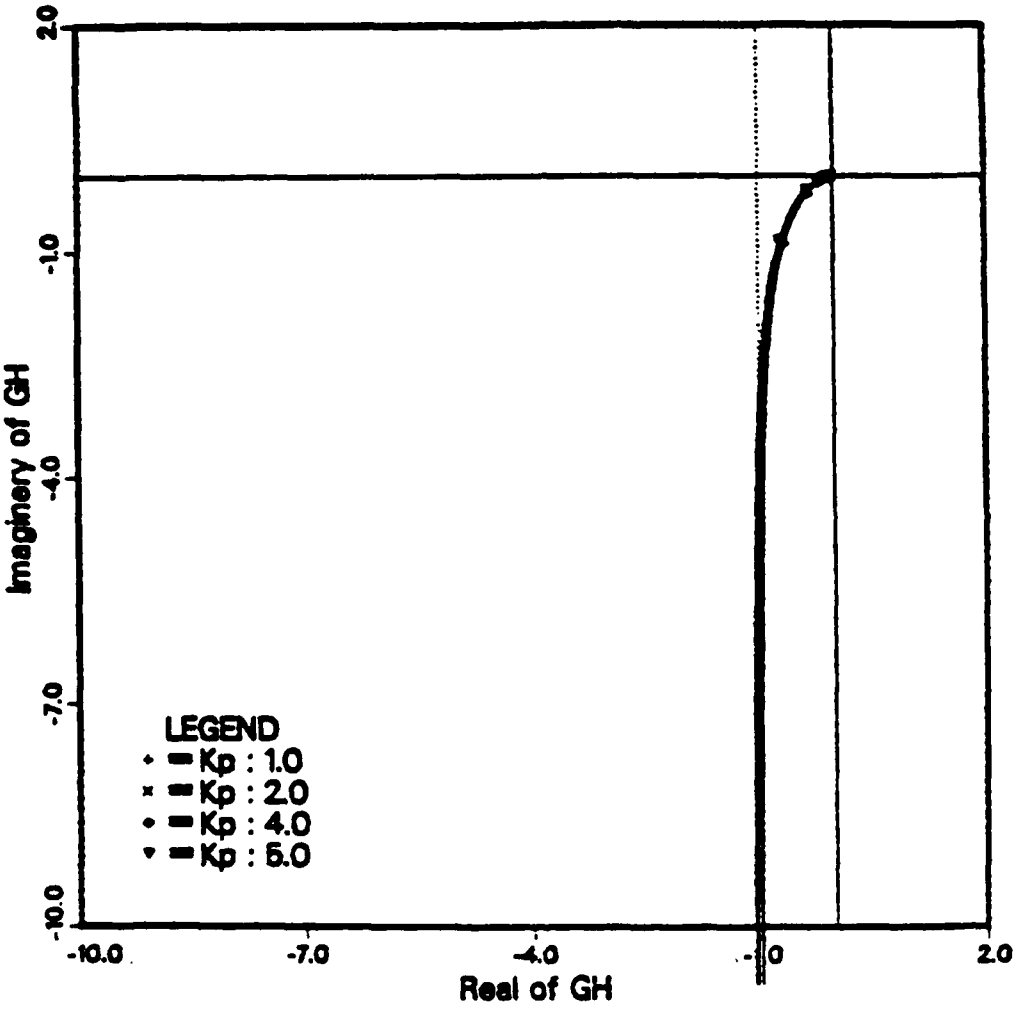


Figure 2.12 Nyquist Plot (Td = 0.00045 sec, Ka•Kp=1).

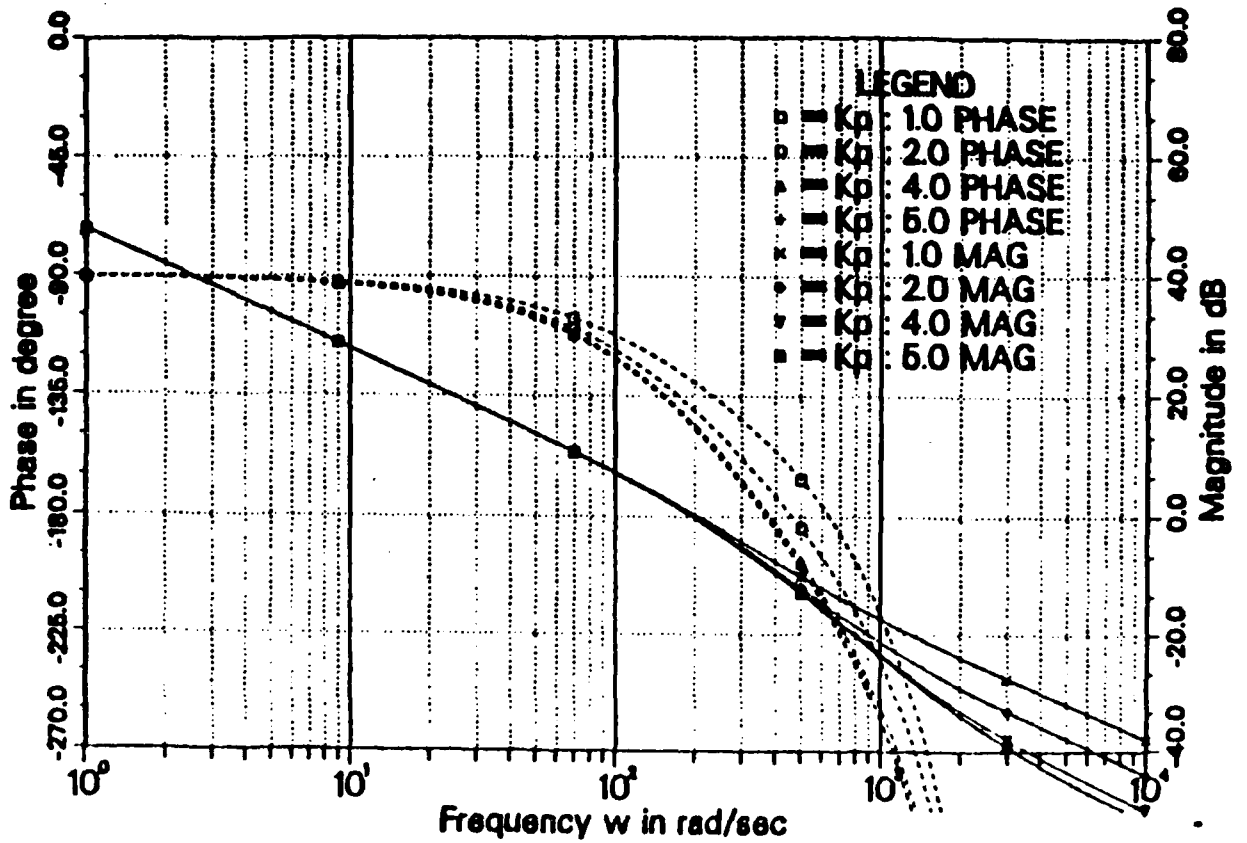


Figure 2.13 Bode Diagram (Td = 0.002 sec, Ka•Kp=1).

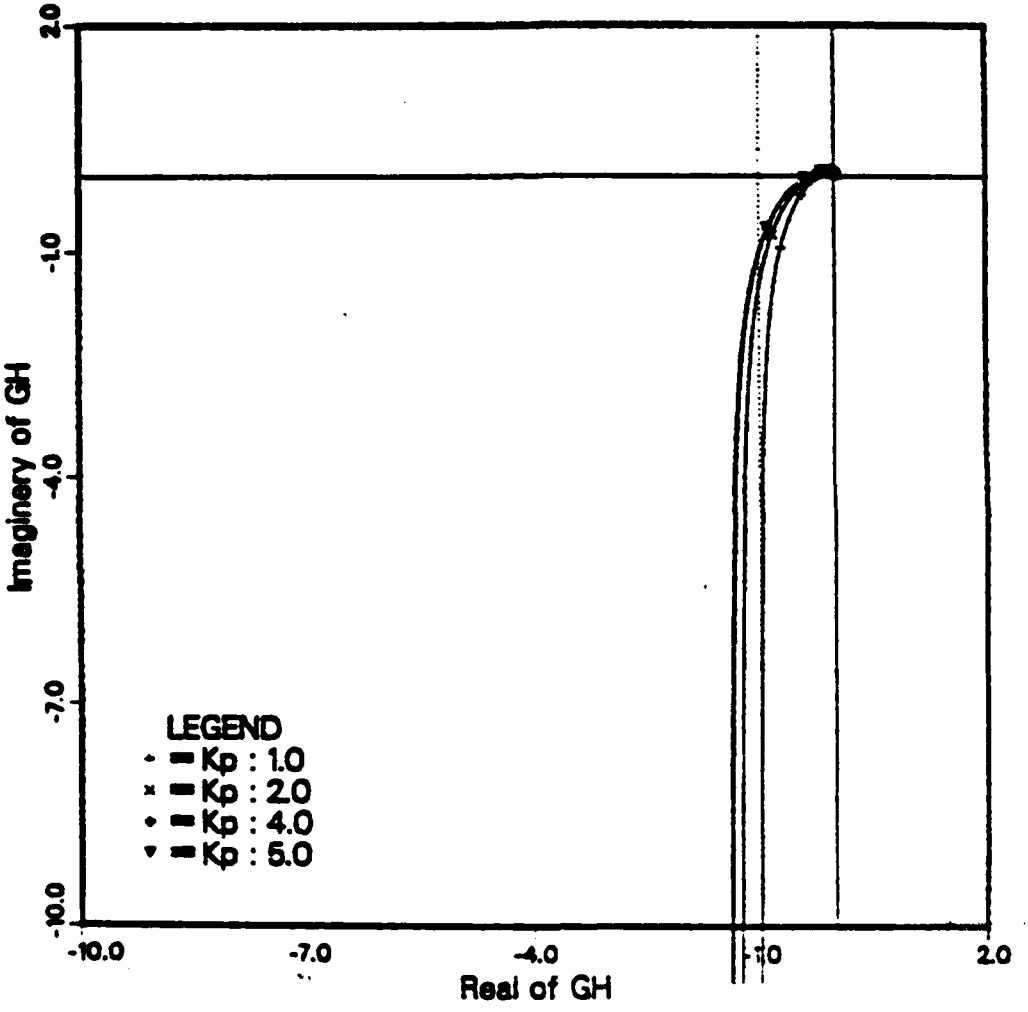


Figure 2.14 Nyquist Plot (Td = 0.002 sec, Ka•Kp=1).

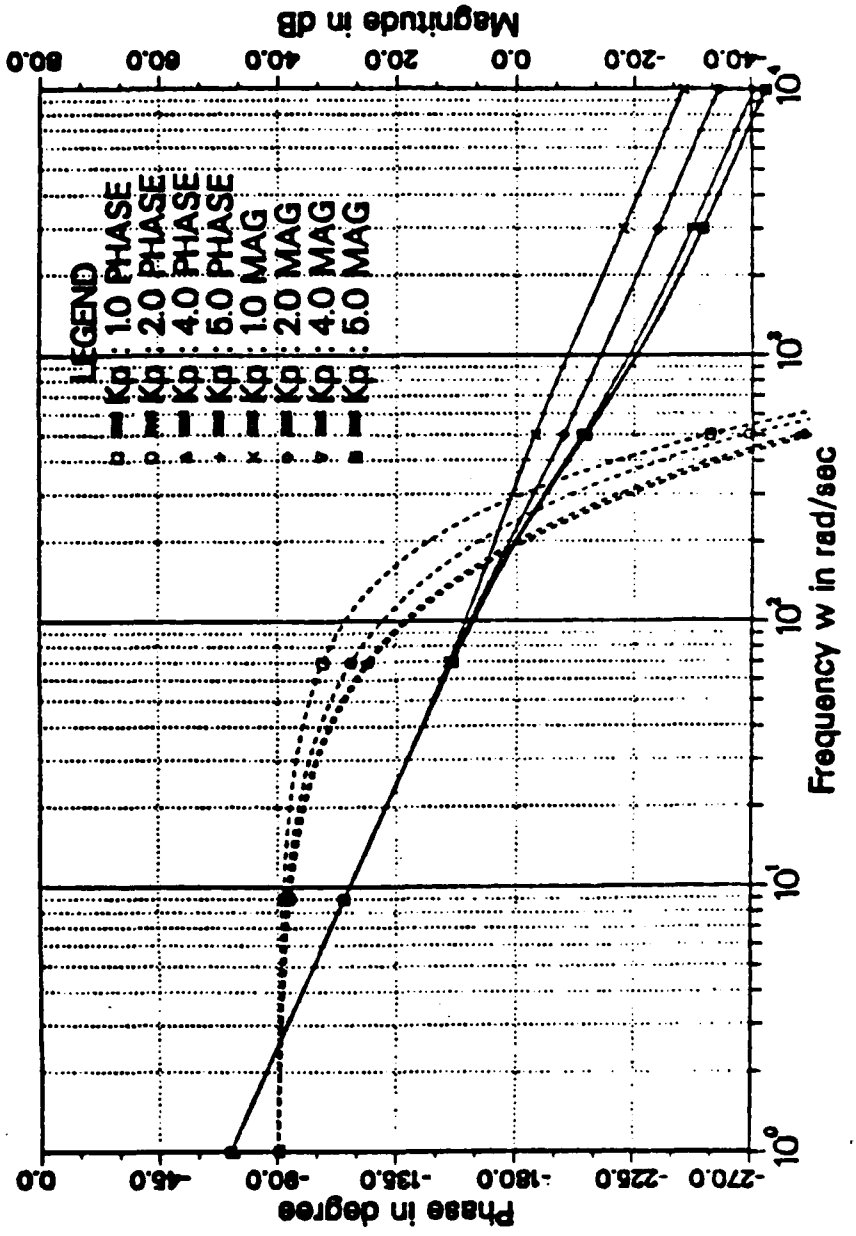


Figure 2.15 Bode Diagram (Td = 0.006 sec, Ka•Kp=1).

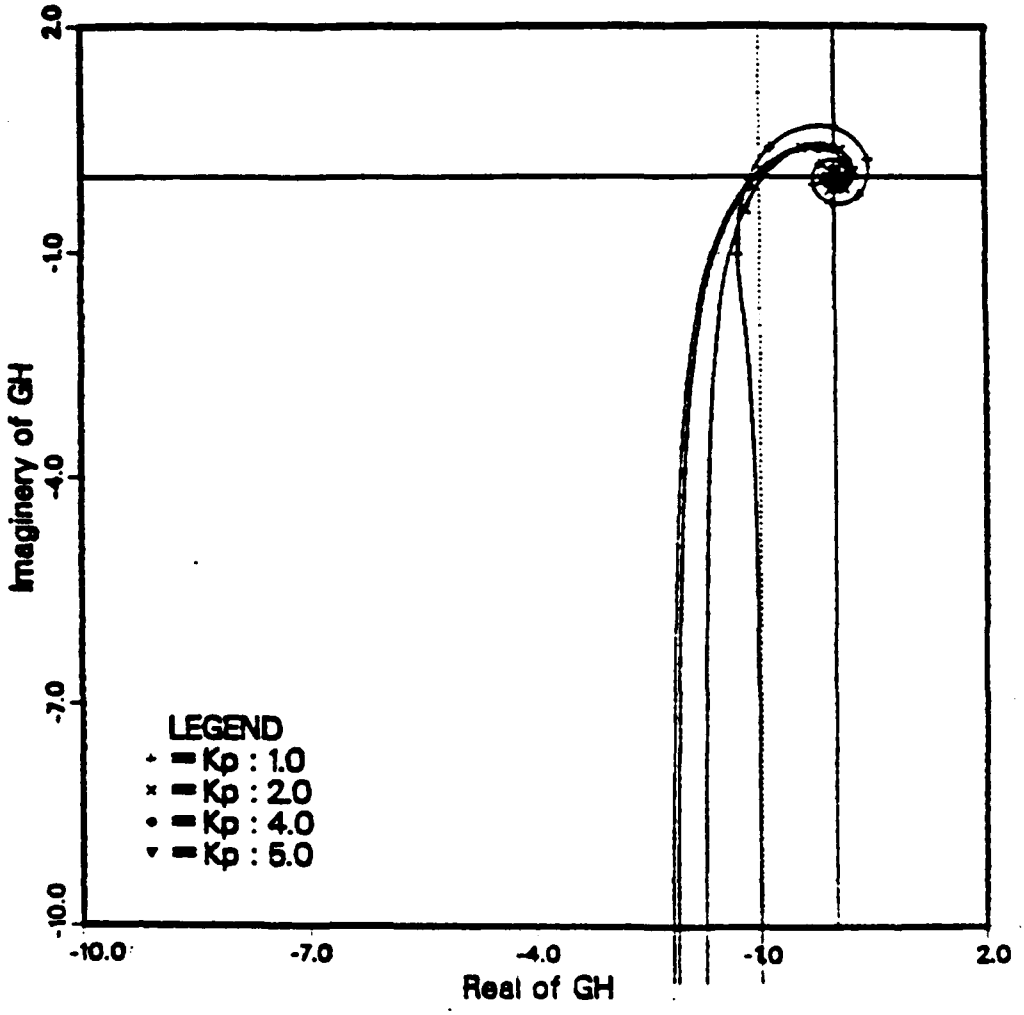


Figure 2.16 Nyquist Plot (Td = 0.006 sec, Ka•Kp=1).

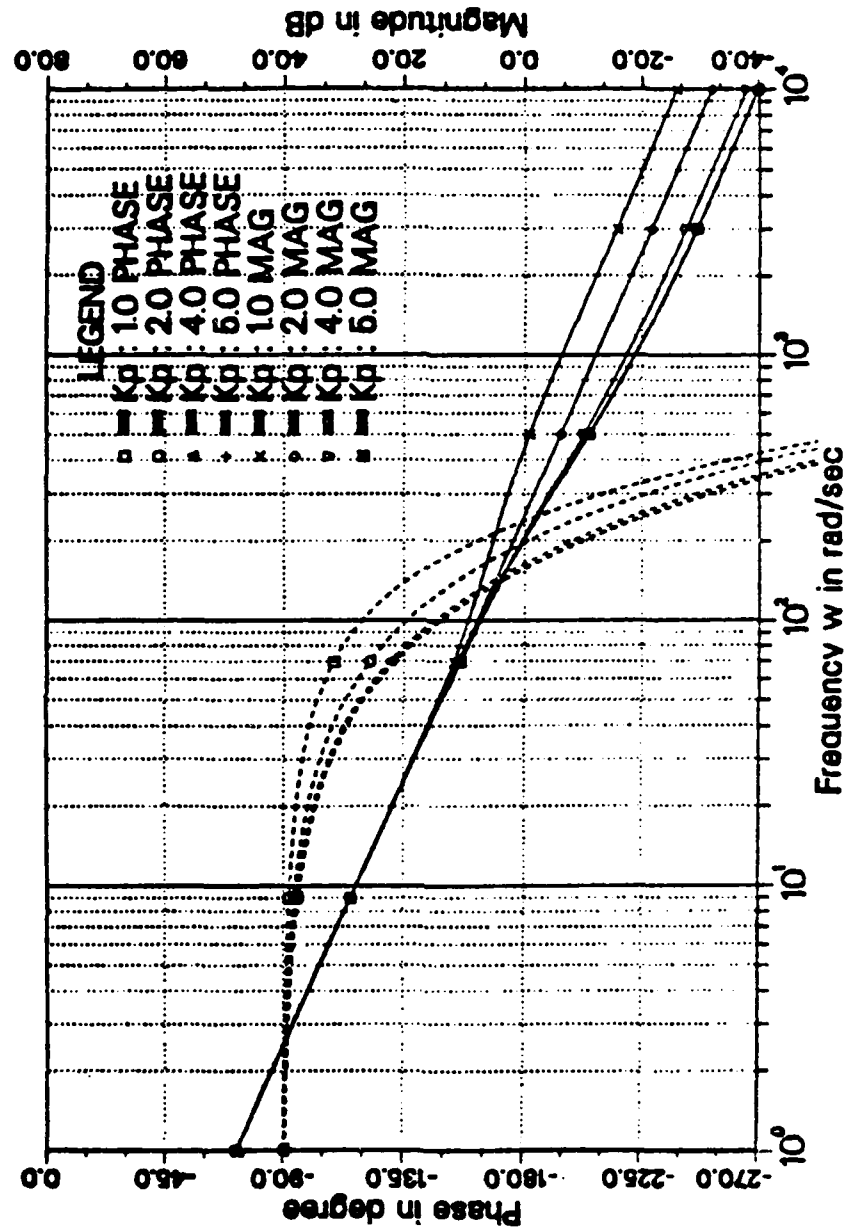


Figure 2.17 Bode Diagram (Td = 0.008 sec, Ka•Kp=1).

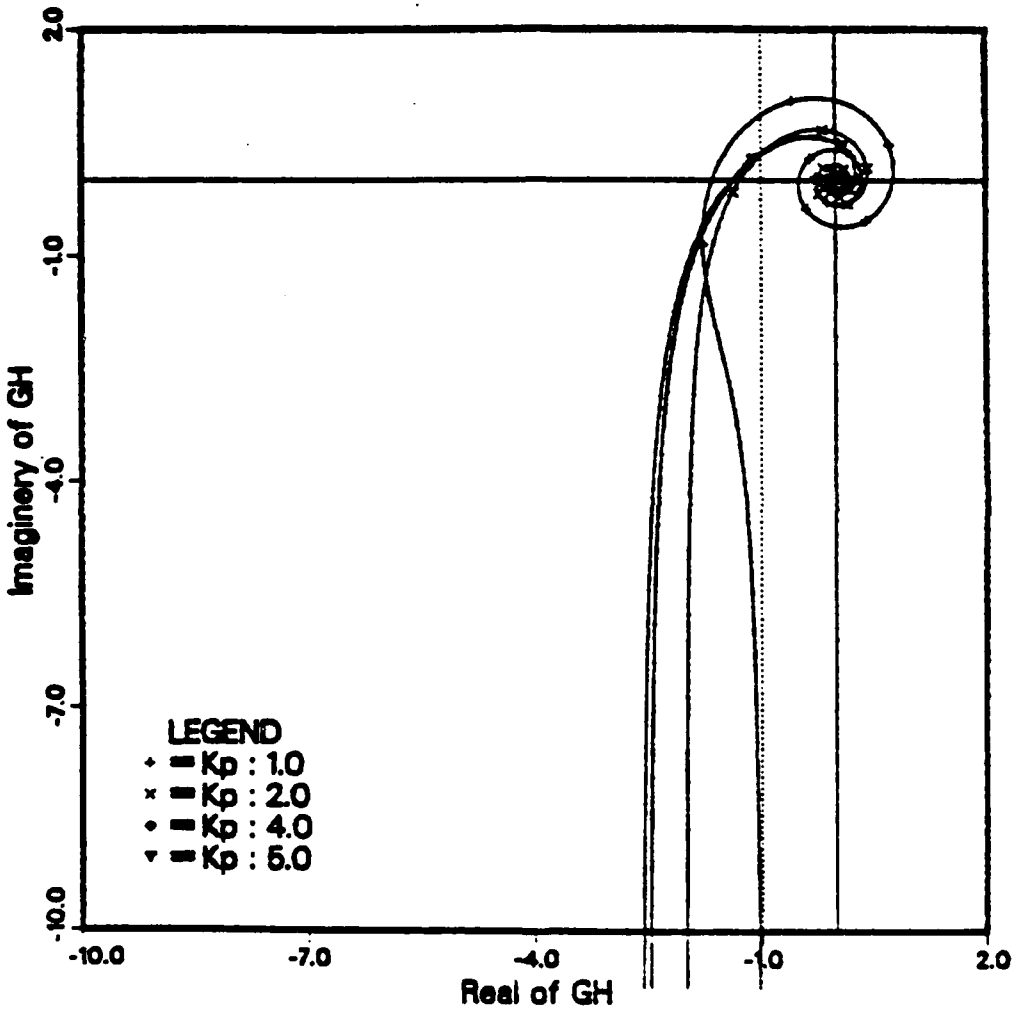


Figure 2.18 Nyquist Plot (Td = 0.008 sec, Ka•Kp=1).

III. SIMULATION

In this section, both a second order system which is the main example and a third order system which is used as an example of application for higher order system are simulated by using the Continuous System Modeling Program (CSMP) control simulation language [Ref. 13]. According to the result of simulation, the transport delay effect cancellation method will be verified.

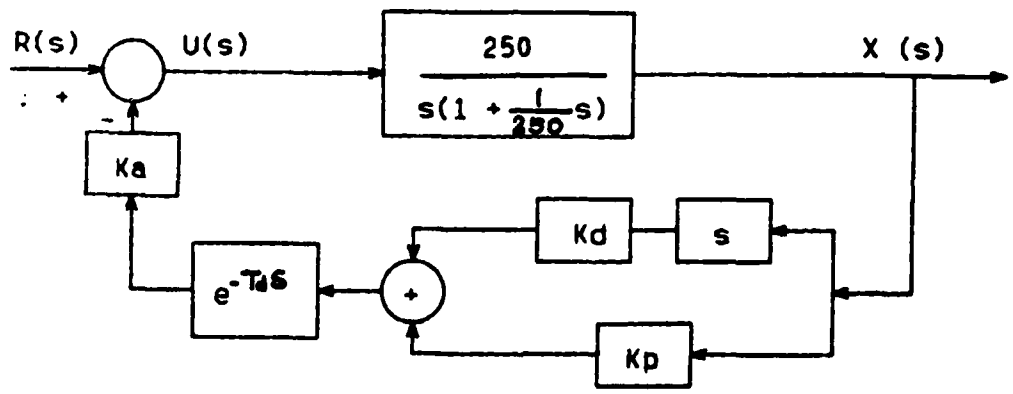


Figure 3.1 Second Order System Block Diagram.

A. SECOND ORDER SYSTEM

First of all, the second order regulator system with the block diagram as shown in Figure 3.1 is treated to verify the adaptability of the design method for compensation of transport type lag. There are four parameters which are Ka,

Kp, Kd and Td in the block diagram. Since the derivative gain(Kd) is assumed the same as transport delay(Td) for convenience in programming of the microcomputer, practically, there are three parameters. Various reasonable values are used in this simulation work in order to investigate the performance of compensation system to be designed by theory.

Figure 3.2 through Figure 3.11 show the results of simulation. Figure 3.2 and Figure 3.3 show the microcomputer regulator response with and without the state feedback compensation. As shown in the Figure 3.2, the system becomes unstable at Td = 6 milliseconds but when compensation is added the response is improved although still unstable. In this case, the attenuation factor Ka = 1. If Ka is reduced in value system stability improves as shown in Figure 3.6 , for example, but there is then a reduction in the system bandwidth which can be observed by the increase in the time of first peak overshoot value. This effect can clearly be seen in Figure 3.7 where the delay is Td = 8 milliseconds.

It should be kept in mind that this simulation result does not take into account such factors as the finite word length of the microcomputer. For this reason the upper limit of transport delay time for microcomputer controller implementation can be taken as 6 milliseconds and is verified in the experimental results which is discussed in the next chapter.

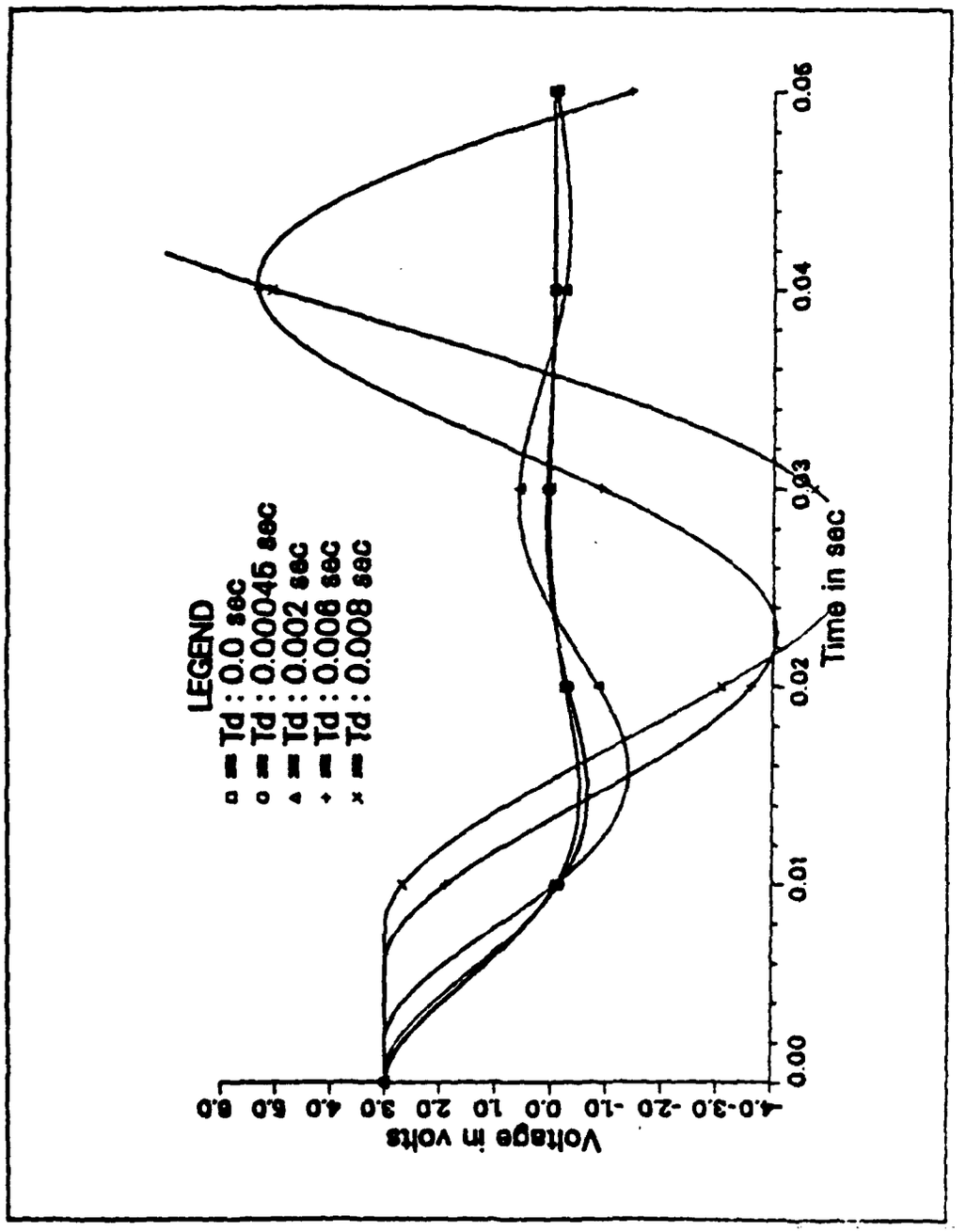


Figure 3.2 Response of Delayed System without Compensation.

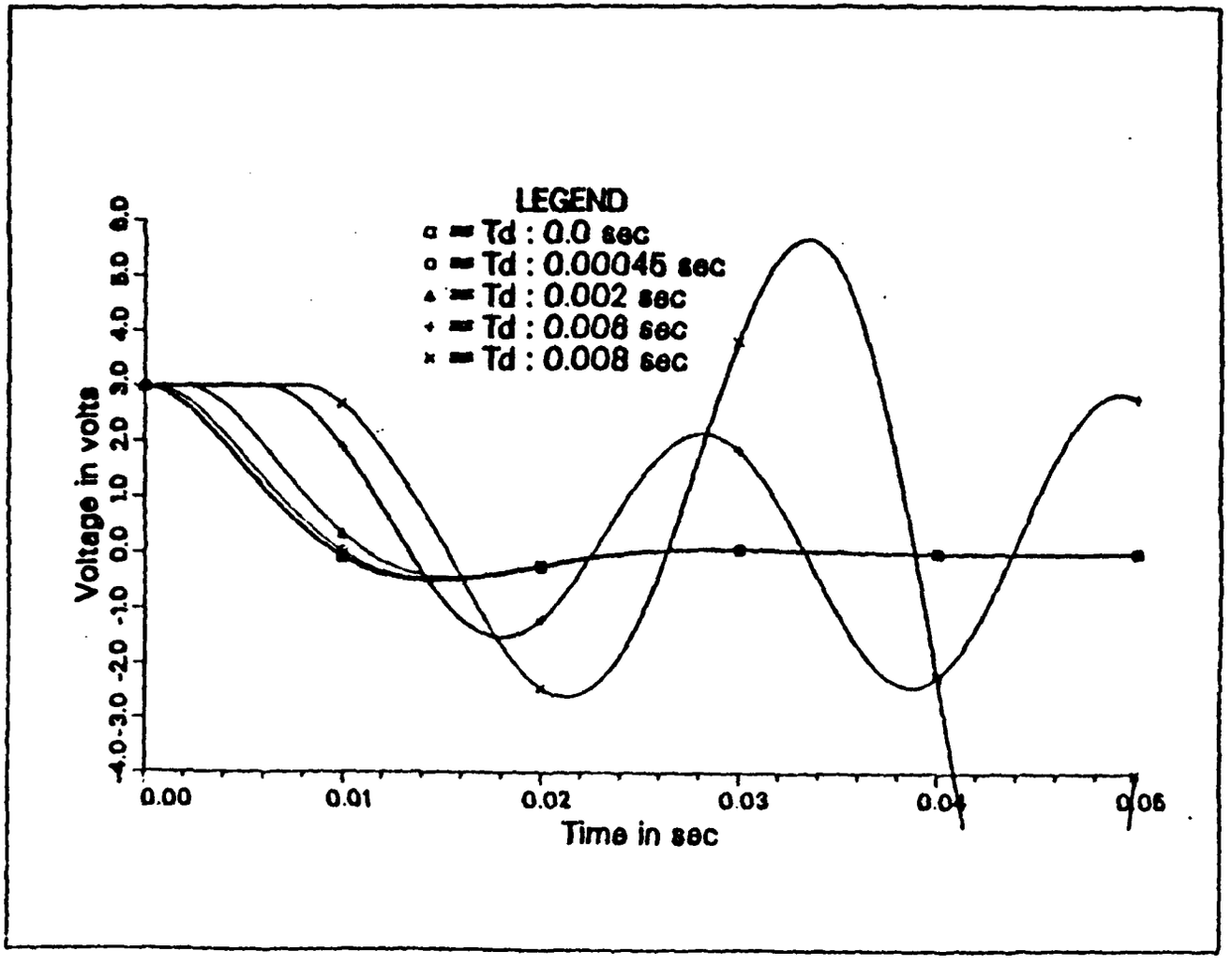


Figure 3.3 Response of Delayed System with Compensation (Kp=1, Ka=1).

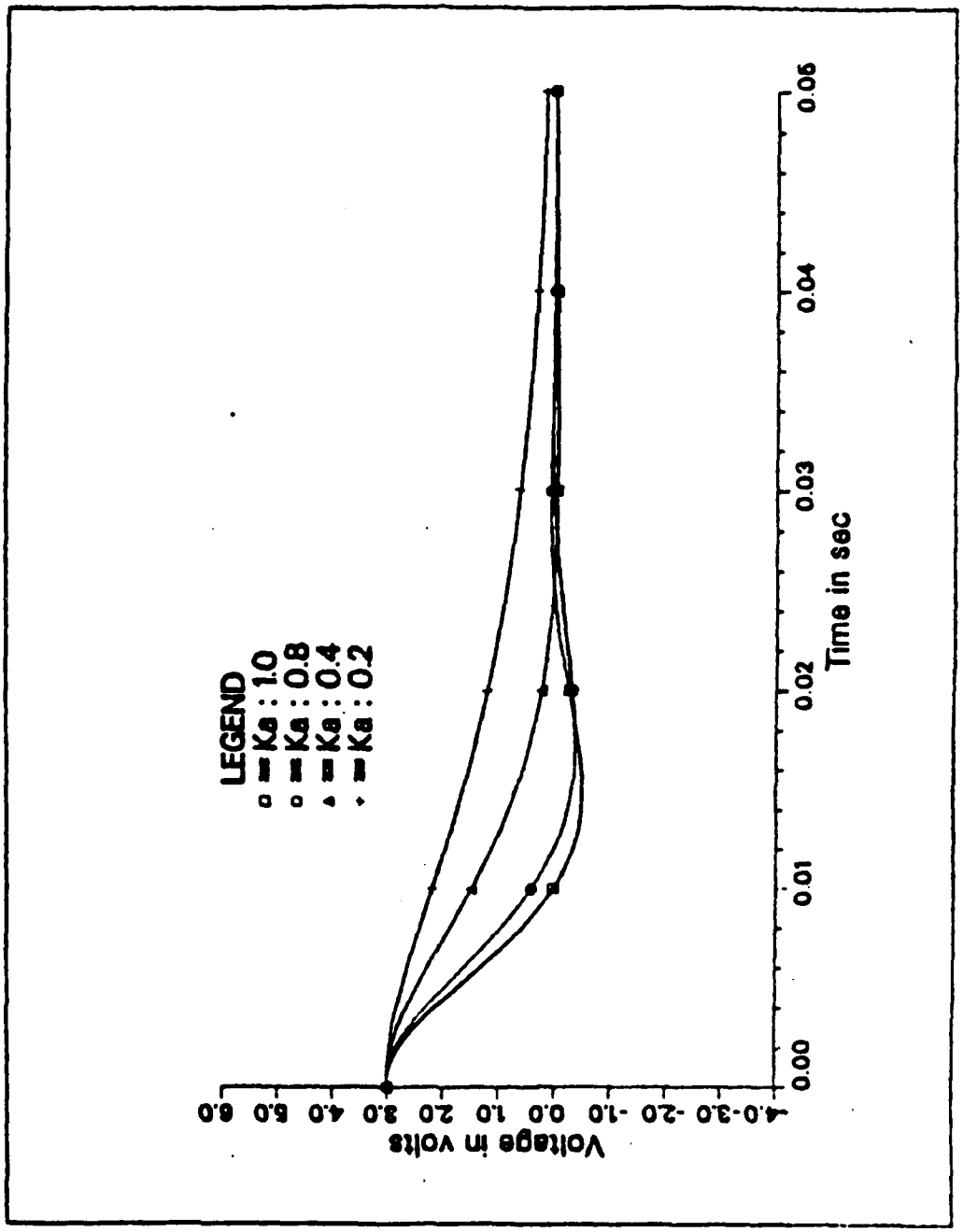


Figure 3.4 Response of Delayed System(Td = 0.00045 sec, Kp=1).

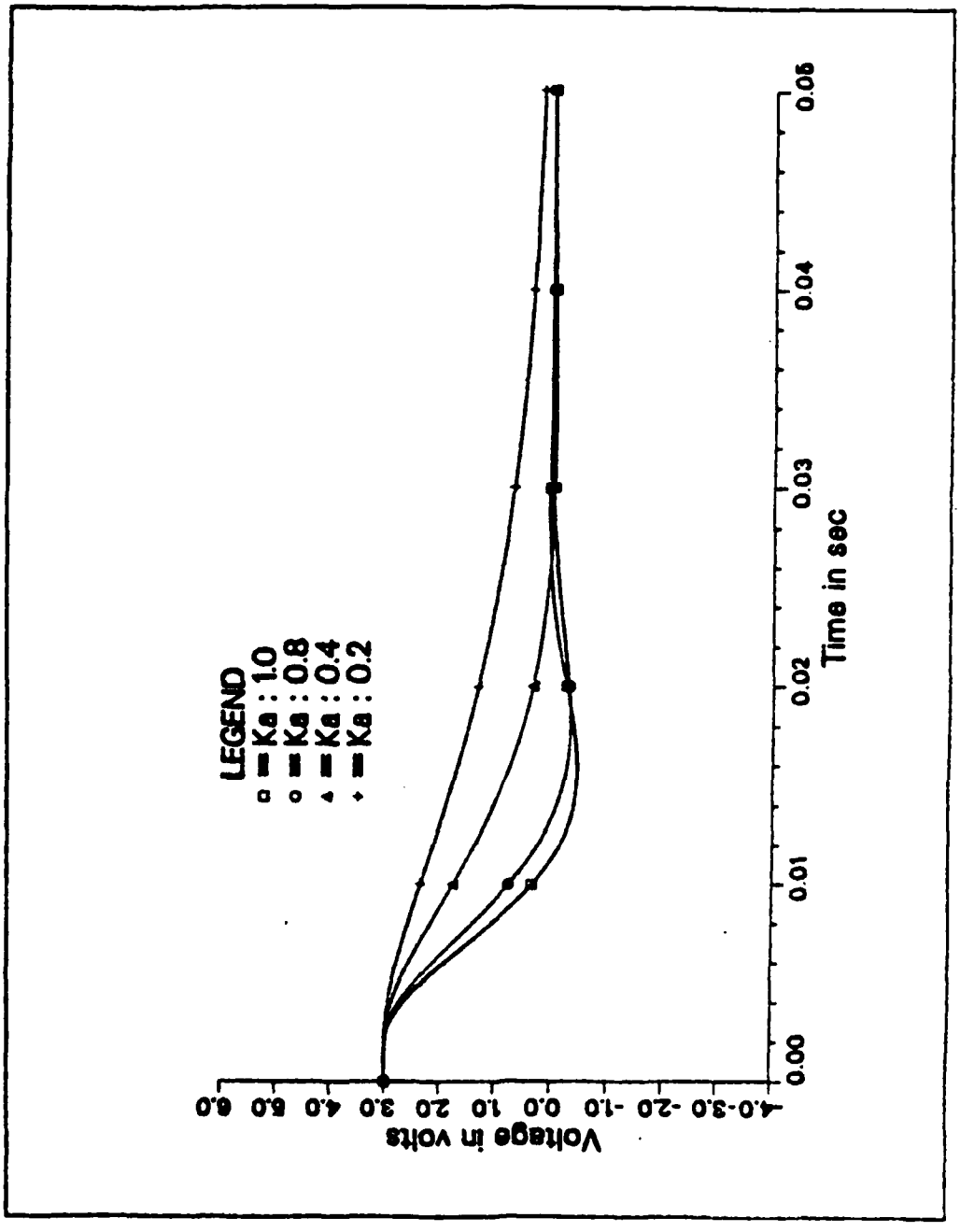


Figure 3.5 Response of Delayed System (Td = 0.002 sec, Kp=1).

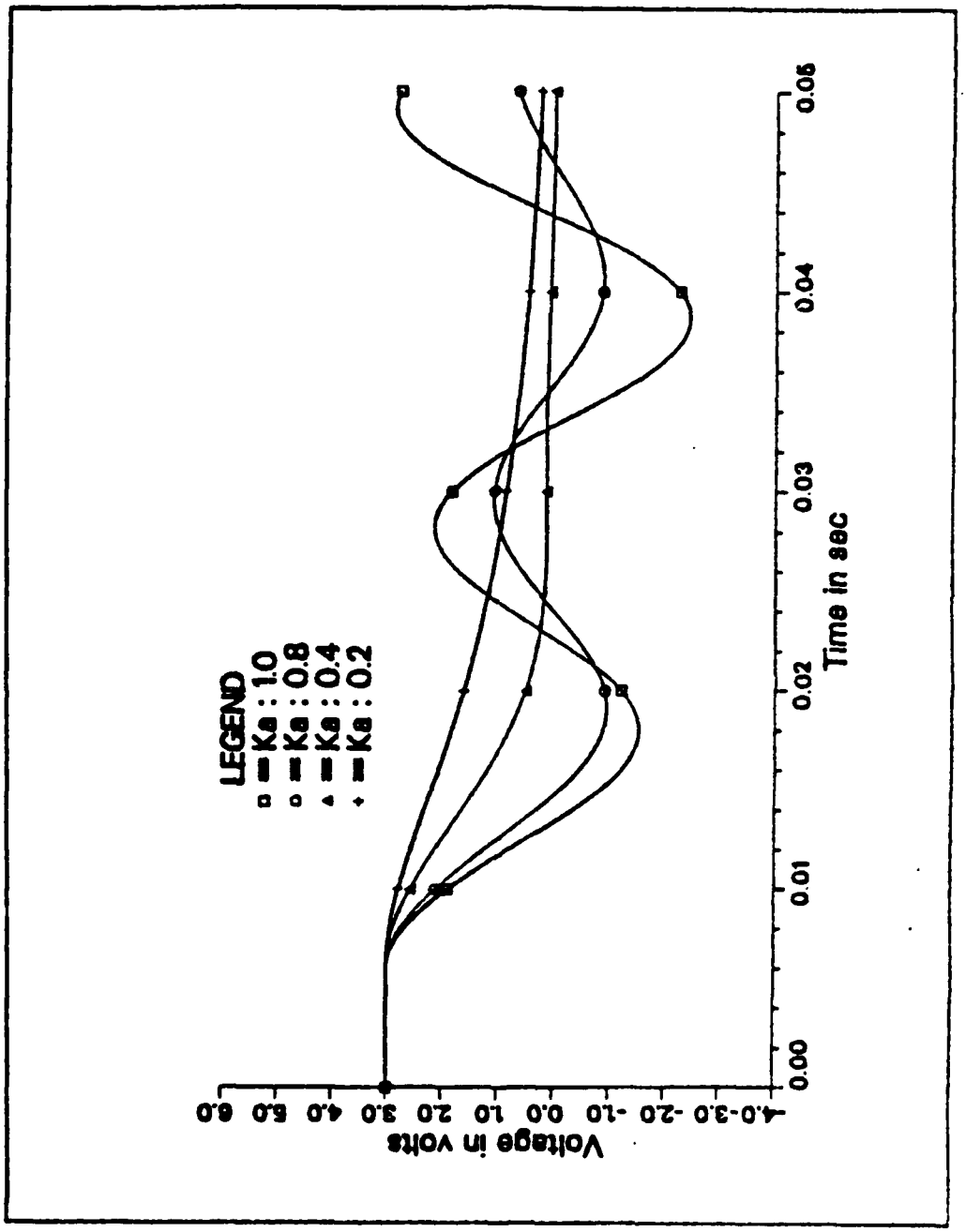


Figure 3.6 Response of Delayed System(Td = 0.006 sec, Kp=1).

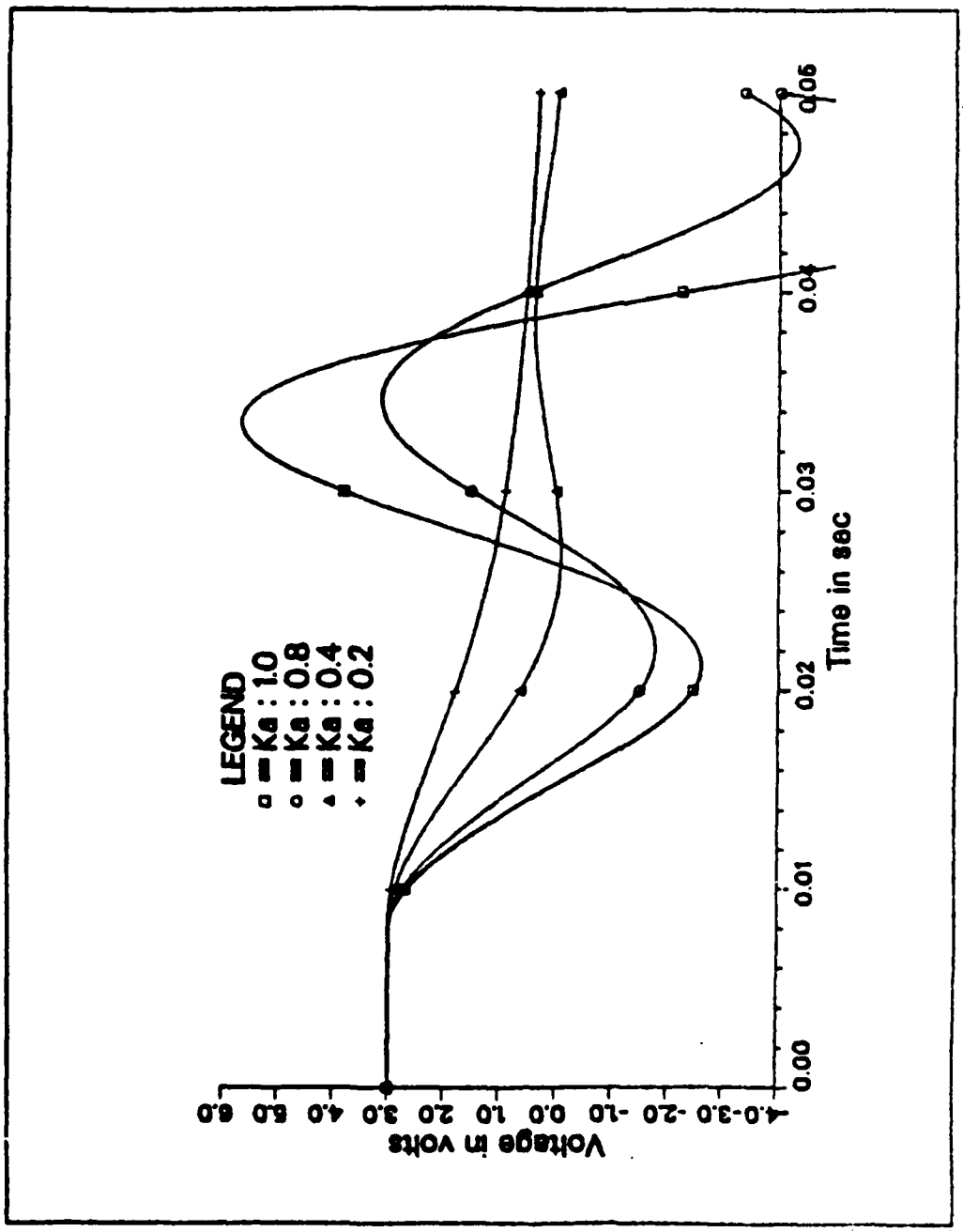


Figure 3.7 Response of Delayed System (Td = 0.008 sec, Kp=1).

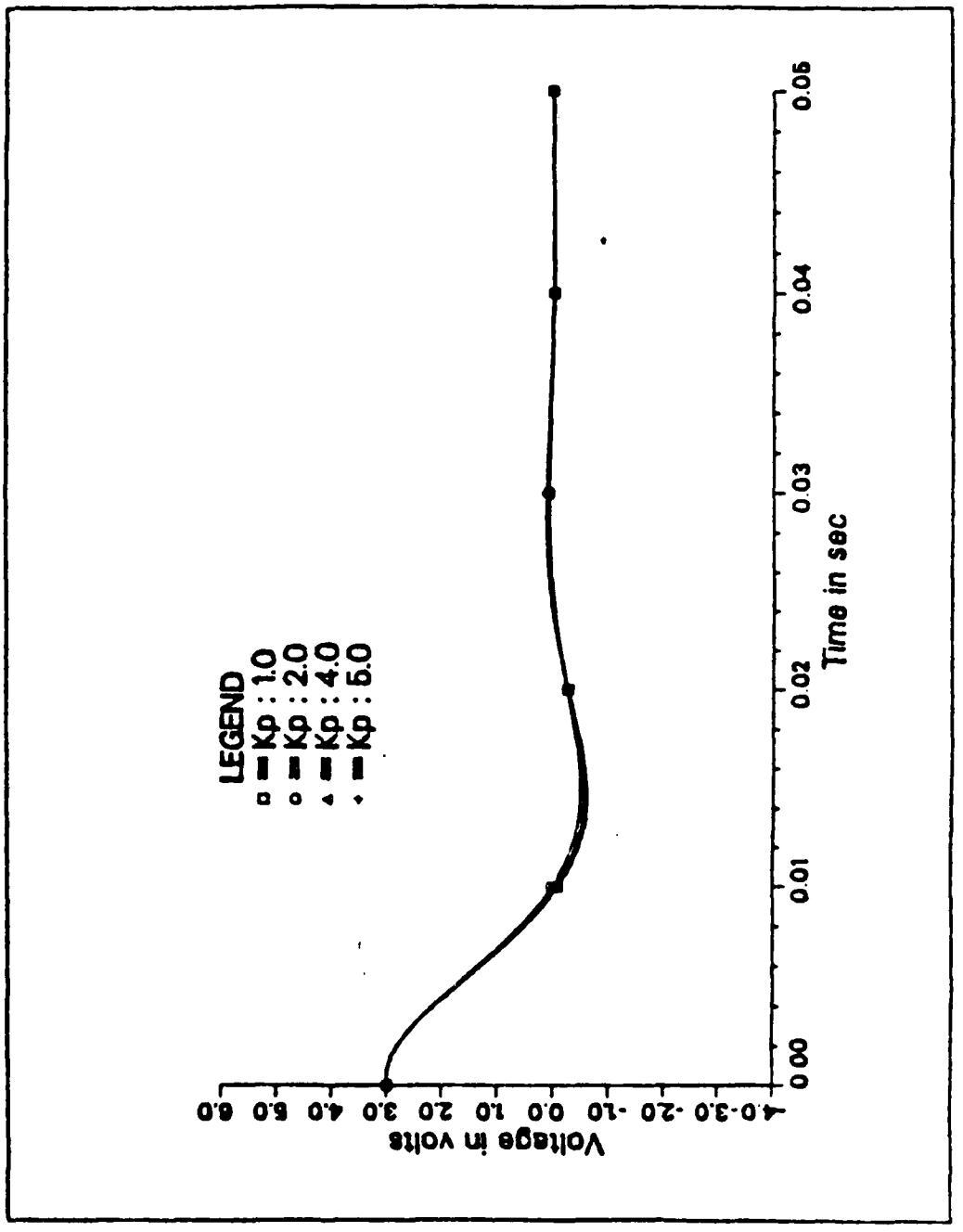


Figure 3.8 Response of Delayed System(Td = 0.00045 sec, Ka•Kp=1).

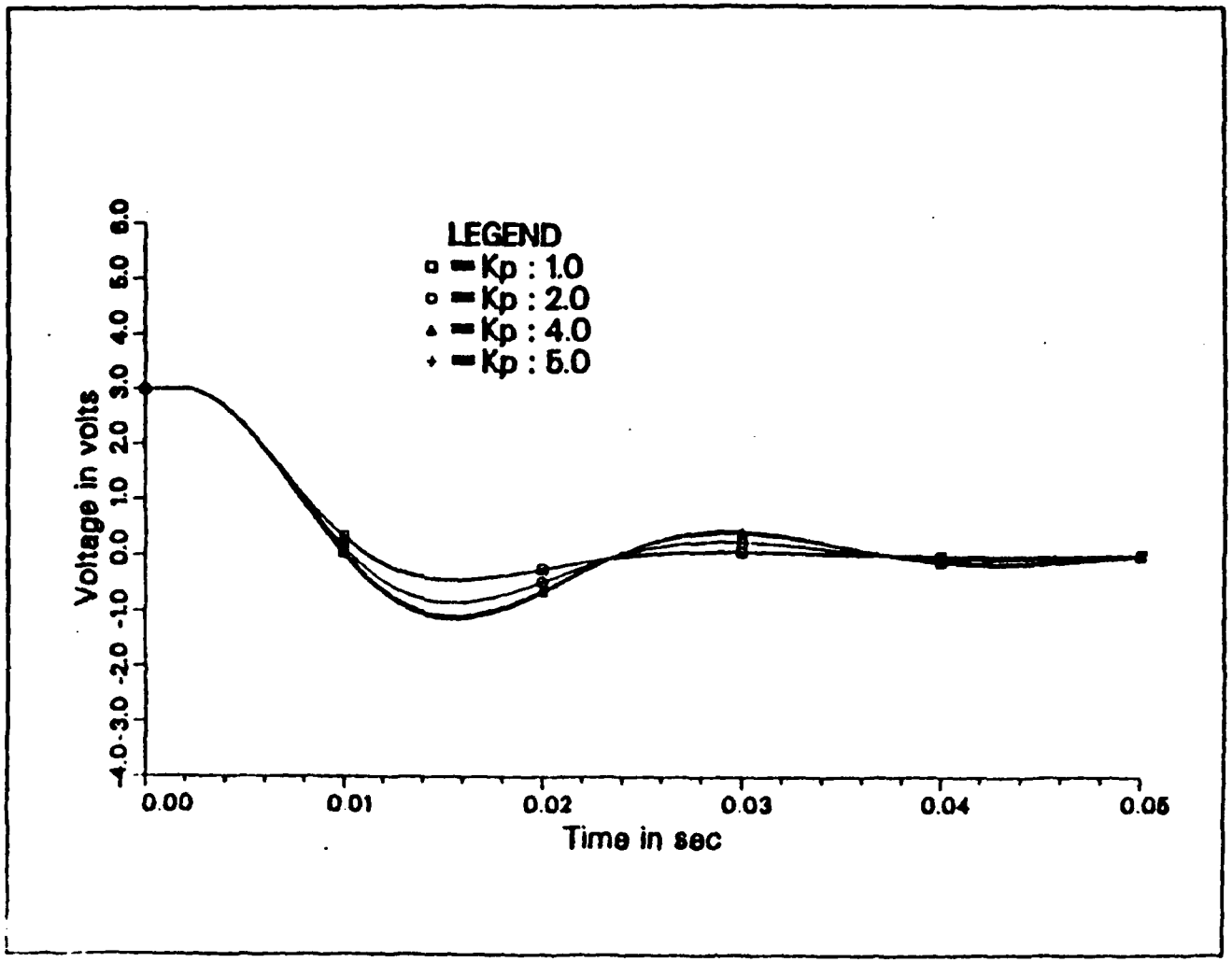


Figure 3.9 Response of Delayed System (Td = 0.002 sec, Ka•Kp=1).

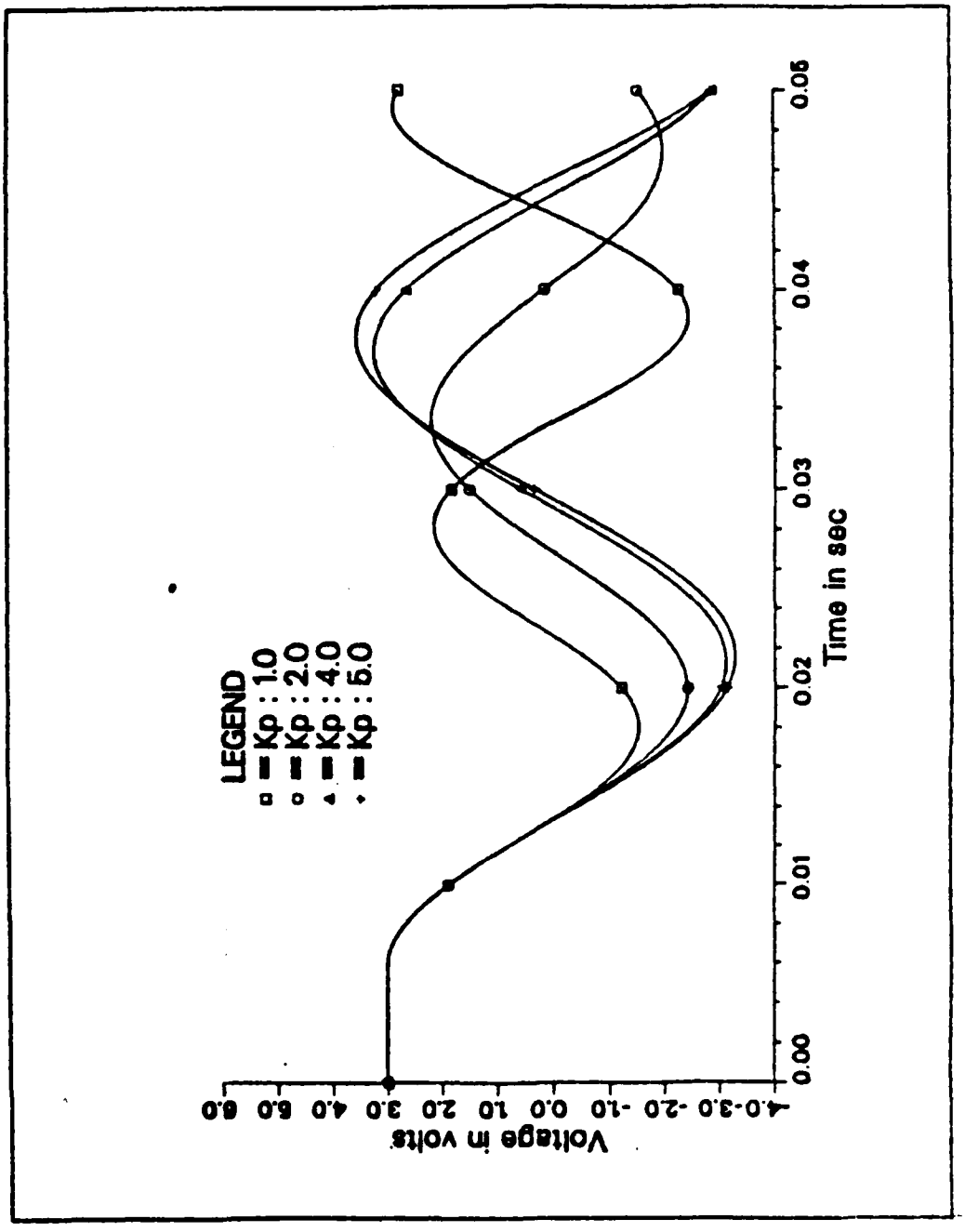


Figure 3.10 Response of Delayed System(Td = 0.006 sec, Ka•Kp=1).

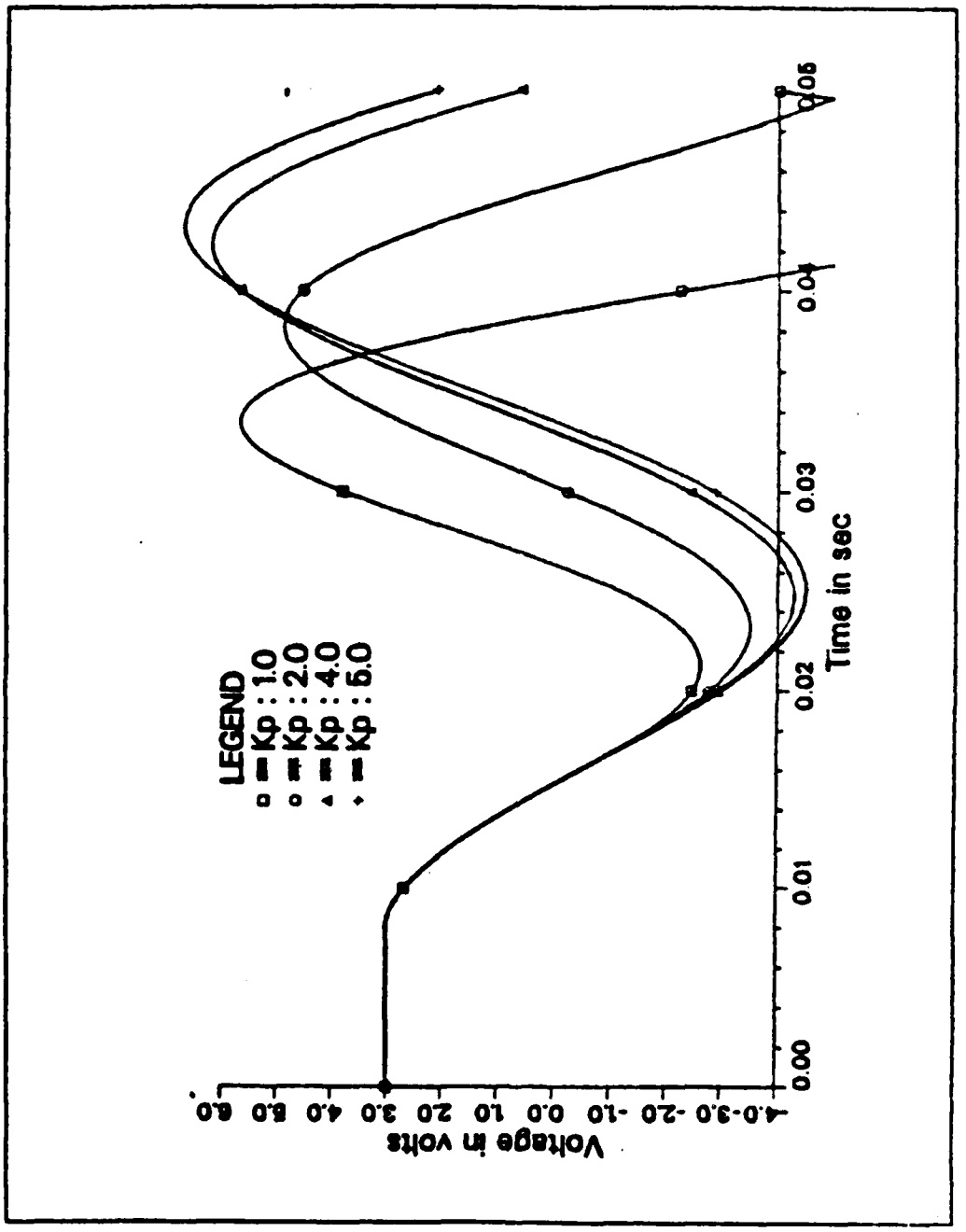


Figure 3.11 Response of Delayed System(Td = 0.008 sec, Ka•Kp=1).

B. HIGHER ORDER SYSTEM

To investigate the method applied to a higher order system, a third order example is used.

Generally, the transfer function of the third order system is written

$$G(s) = \frac{\alpha \omega_n^2}{(s+\alpha)(s^2 + 2\zeta\omega_n s + \omega_n^2)} \qquad \text{(eqn 3.1)}$$

$$\beta = \frac{\alpha}{\zeta\omega_n} \qquad \text{(eqn 3.2)}$$

and For the numerical example, the state equation is given

$$\dot{\underline{x}} = \underline{A}\,\underline{x} + \underline{B}\,u \qquad \text{(eqn 3.3)}$$

where

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -10 \end{bmatrix} \qquad \text{and} \qquad \underline{B} = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix}$$

with control $u = G\,x$ where

$$\underline{G} = [\ 1 \quad 0.5 \quad 0\ ]$$

Note that by design g_3 is not used here since it is required to compensate for phase lag due to Td.

Then the characteristics equation is

$$|\ s\,I - \underline{A} + \underline{B}\,\underline{G}\ | = 0$$

Thus, the equation becomes

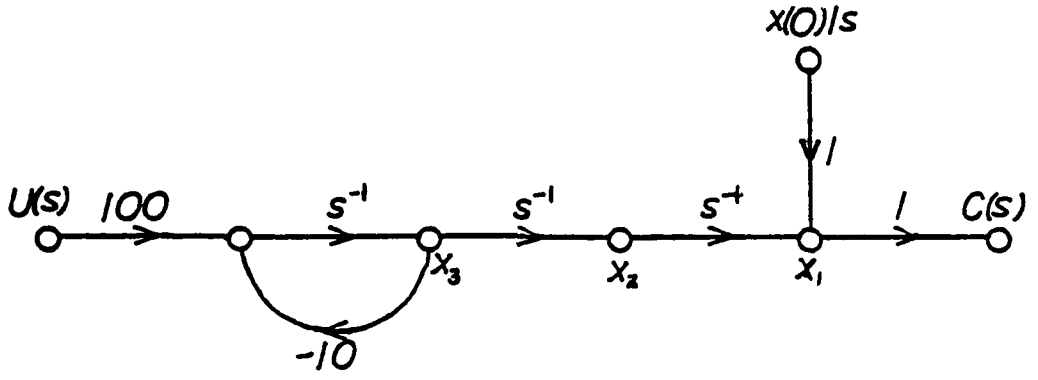


Figure 3.12 Signal Flow Graph of 3rd Order Plant.

$$s^3 + 10 \cdot s^2 + 50 \cdot s + 100 = 0 \qquad \text{(eqn 3.4)}$$

The parameters, β and ζ are chosen arbitrarily.

$$\beta = 2$$
$$\zeta = 0.5$$

Using these two parameters, the percent overshoot is determined to be about 8 % from using the design chart of Ref. 10.

The third order plant Signal Flow Graph is shown in Figure 3.12 .

From Mason's Gain Rule, the transfer function is

$$G(s) = \frac{X_1}{U} = \frac{100}{s^2(s + 10)} \qquad \text{(eqn 3.5)}$$

and, the less than full state feedback is

$$u = -x_1 - 0.5 \cdot x_2 \qquad \text{(eqn 3.6)}$$

Then

$$H(s) = -\frac{U}{X_1} = 1. + 0.5 \cdot s \qquad \text{(eqn 3.7)}$$

The closed loop transfer function is

$$GH(s) = \frac{10(1 + 0.5S)}{S^2(1 + 0.1S)} \qquad \text{(eqn 3.8)}$$

Now, when designing for the microcomputer controller, full state feedback will be used. Since n = 3, k = 2, and m = 3 as stated in the theory, the additional state feedback yields

$$H(s) = -\frac{U}{X_1} = Ka(1. + 0.5 \cdot s + g \cdot s^2) \qquad \text{(eqn 3.9)}$$

This is a feedback transfer function for the third order system. Thus, the complete closed loop transfer function is

$$GH(s) = \frac{10 \cdot Ka(1 + 0.5S + g_3 S^2)}{S^2(1 + 0.1S)} \cdot e^{-\tau_d S} \qquad \text{(eqn 3.10)}$$

The Ka and g are selected so that the phase shift due to transport delay(Td) is effectively cancelled. Figure 3.13 shows the block diagram of the third order system with

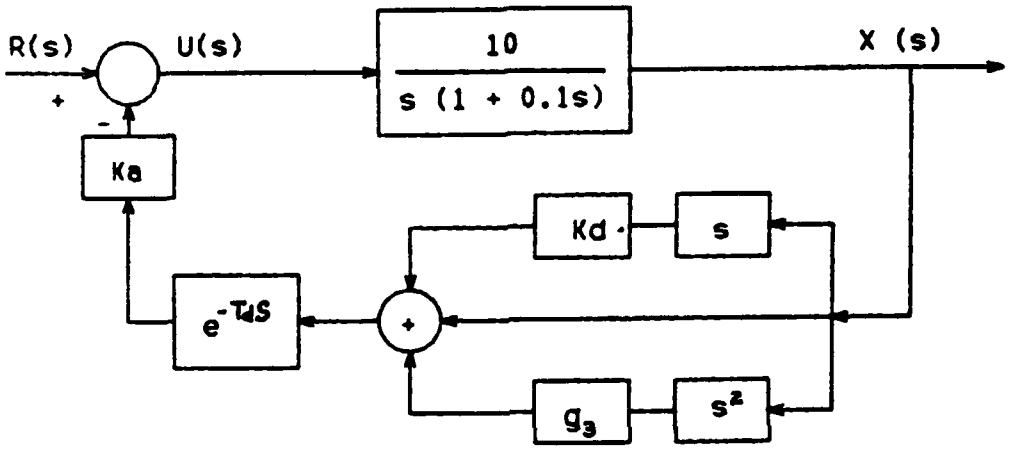


Figure 3.13 Third Order System Block Diagram.

compensation. In this simulation of third order regulator system, it can be expected that the compensation theory using microcomputer should be available for high order system.

In Figure 3.14, simulation results are shown for direct analog feedback as well as microcomputer controller design with and without compensation for various values of transport delay. Figure 3.15 and Figure 3.16 show their results seperately. Figure 3.17 shows the effect of state feedback

gain g on the response for transport lag of 50 milliseconds. Additional time response results for the parameters study are shown in Figure 3.18 to Figure 3.25 . The Nyquist plots for each of the above cases are presented in Figure 3.26 to Figure 3.35 .

The results of this simulation study indicate that the method can be applied successfully in a microcomputer controller dasign for higher order system. In order to provide the desired degree of compensation, however, more freedom would be necessary in the use of the parameters of the controller. This might require that the direct feedback design use $k \leq (n-2)$ rather than $k \leq (n-1)$ state feedback gains which would allow additional phase compensation to be applied in the required frequency range for suitable phase shift cancellation.

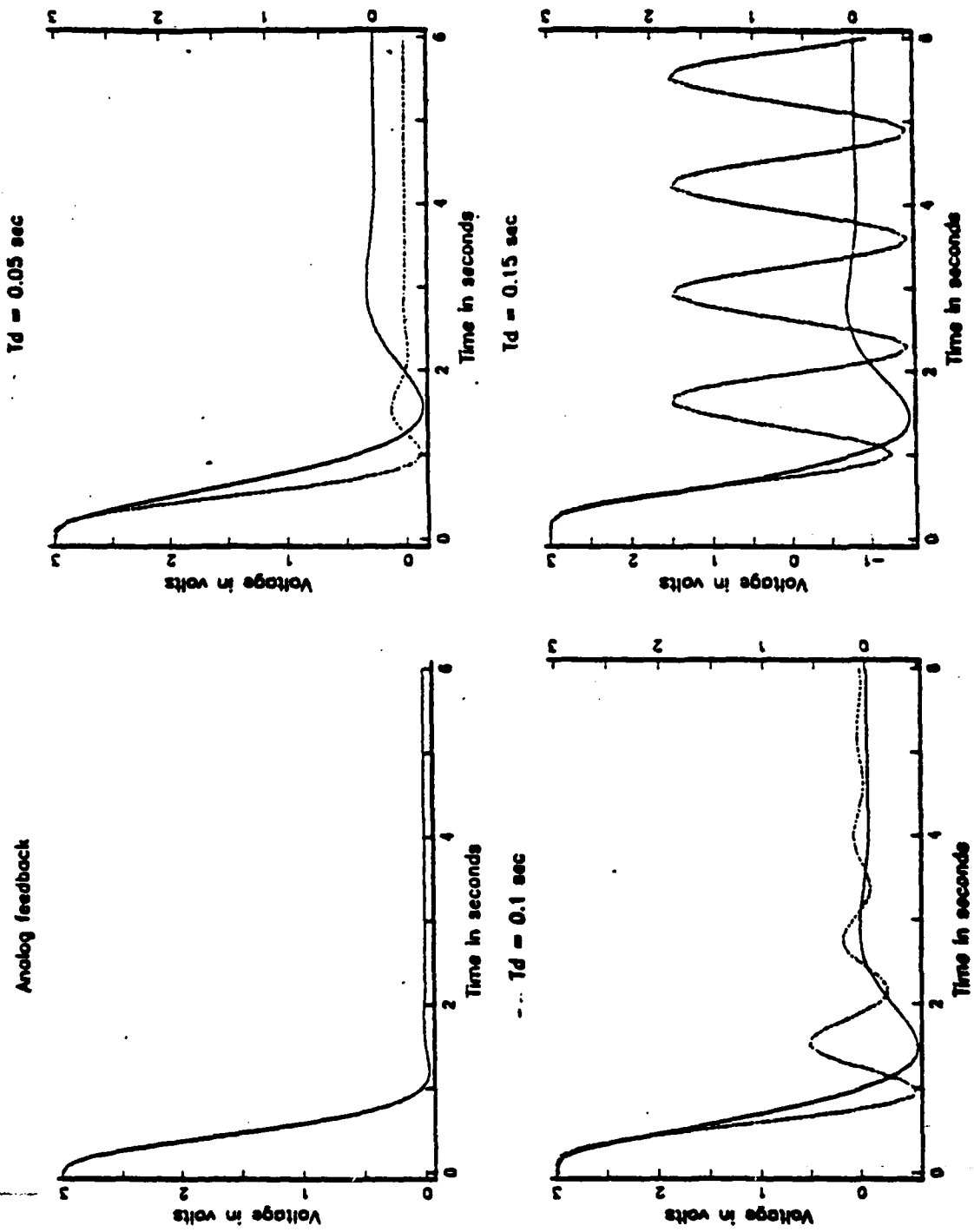


Figure 3.14 Simulation Result Summary.

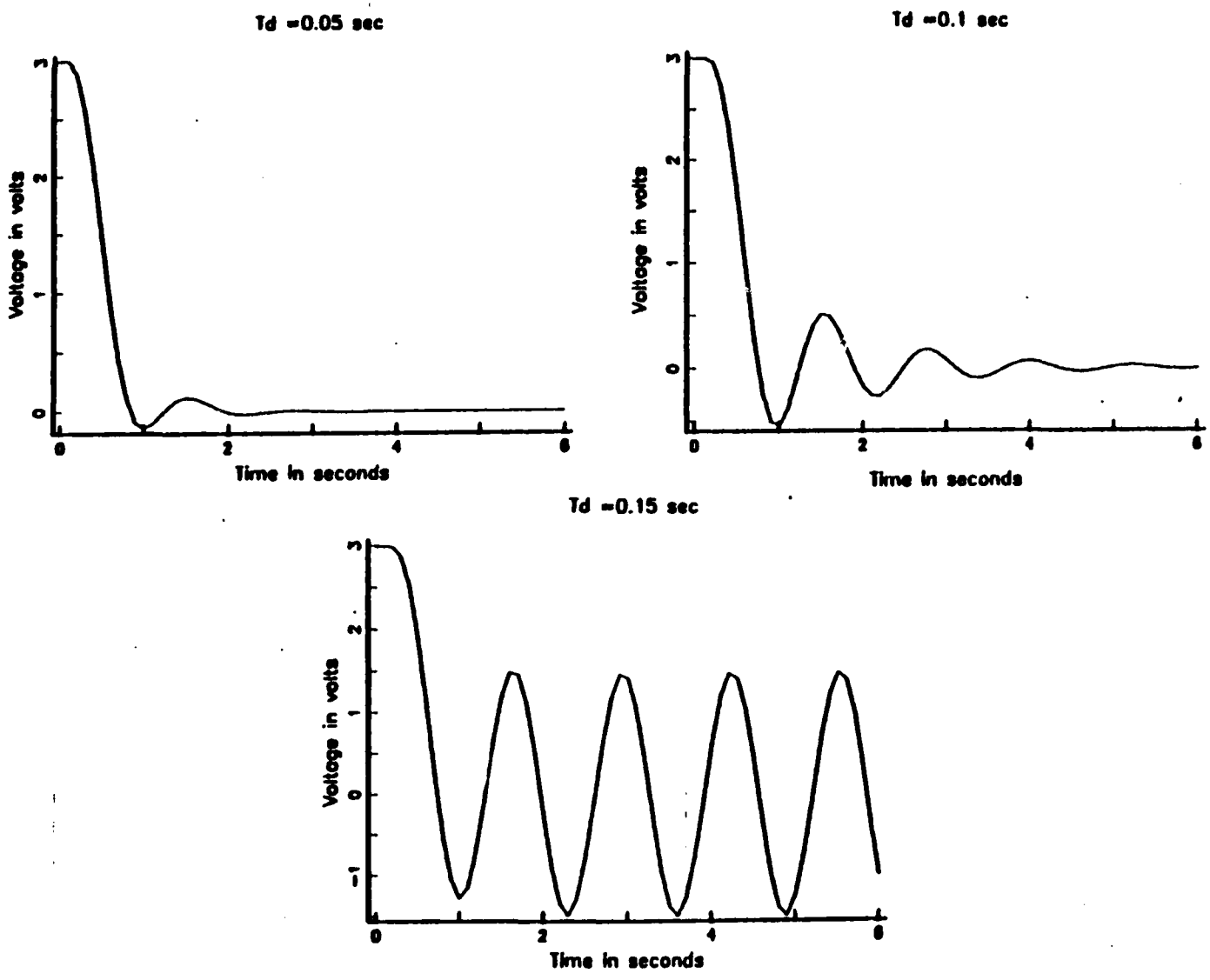


Figure 3.15 Simulation Result without Compensation.

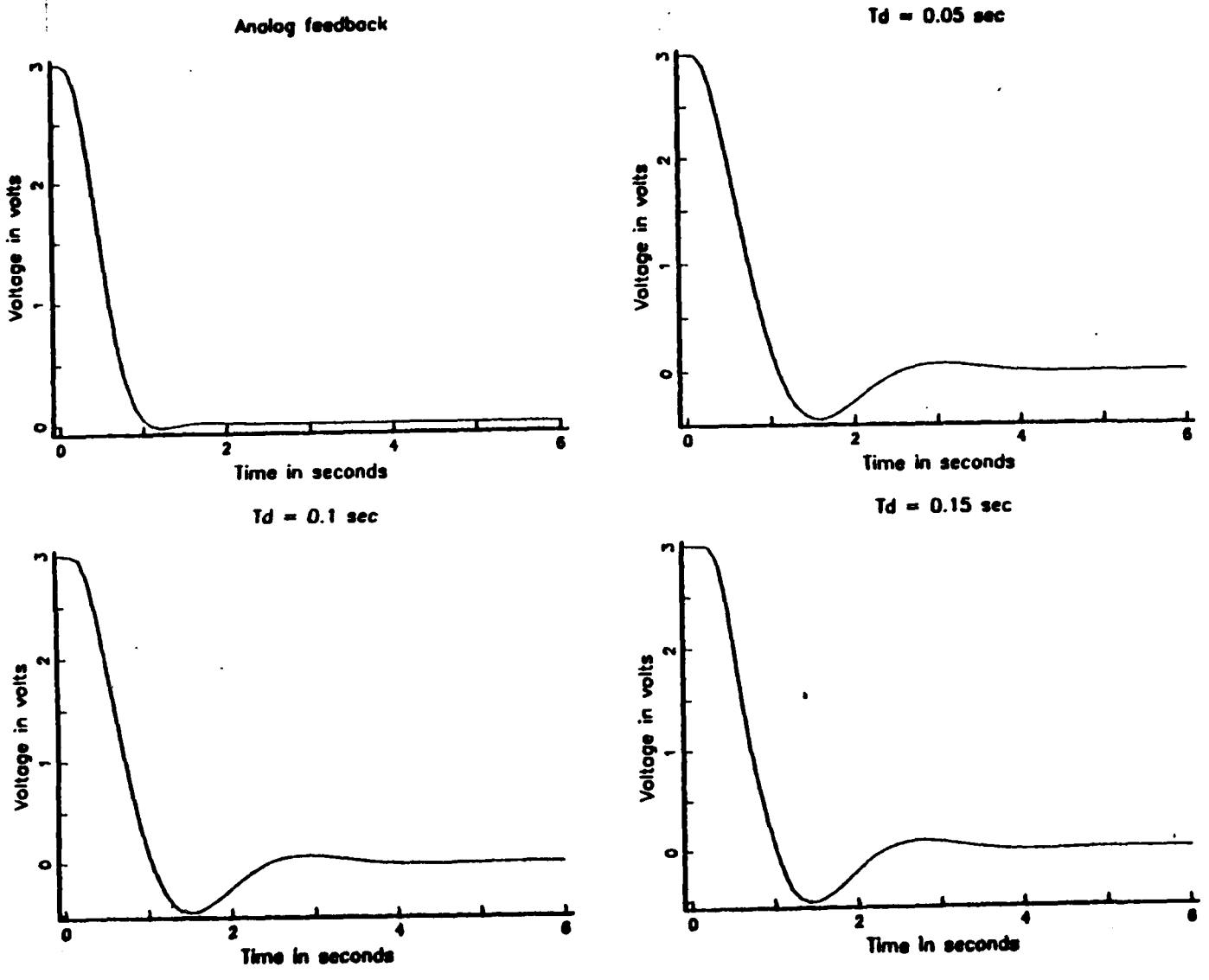


Figure 3.16 Simulation Result with Compensation (g = 0.1).

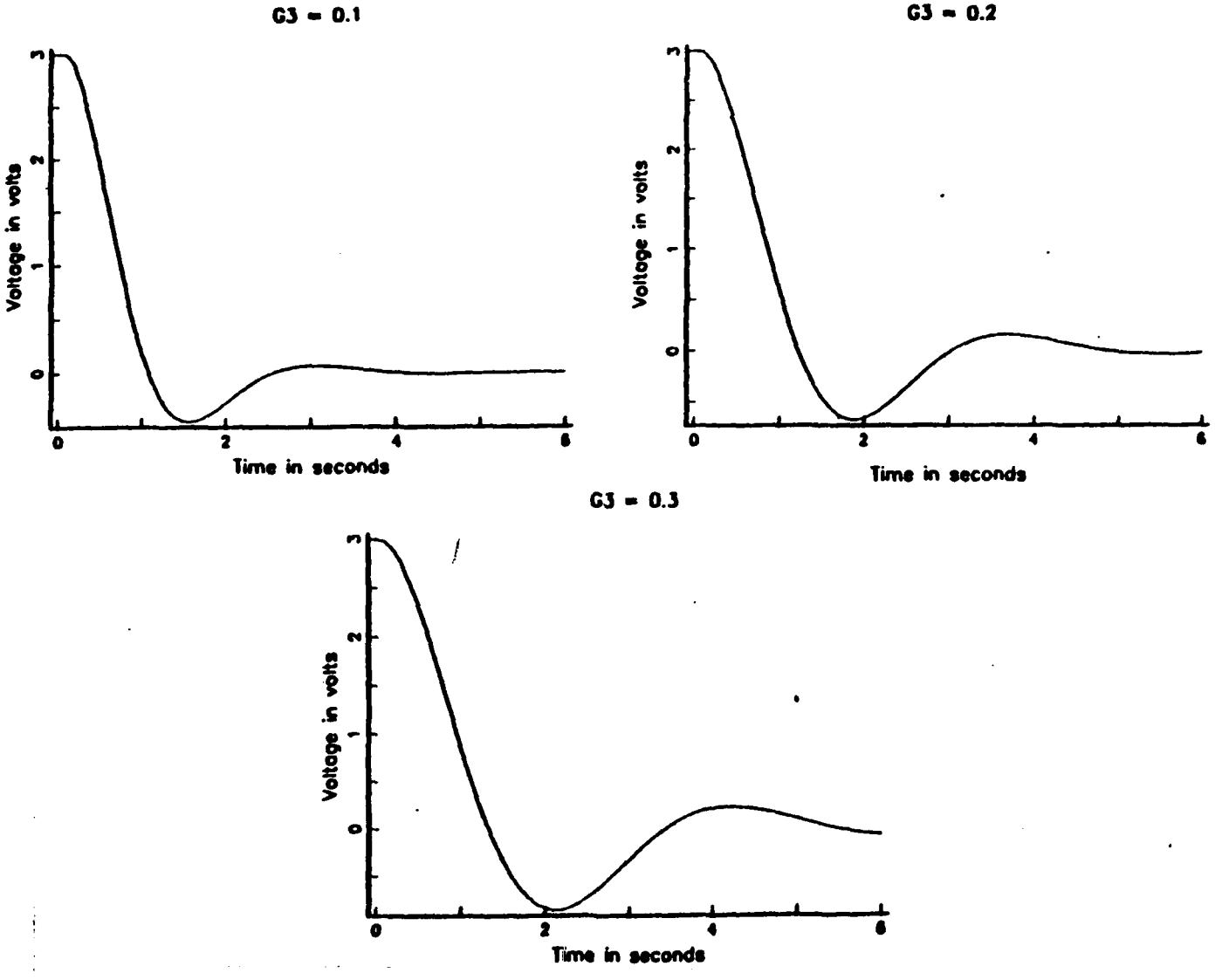


Figure 3.17 Simulation Result with Compensation (Td = 0.05 sec, Ka=1).

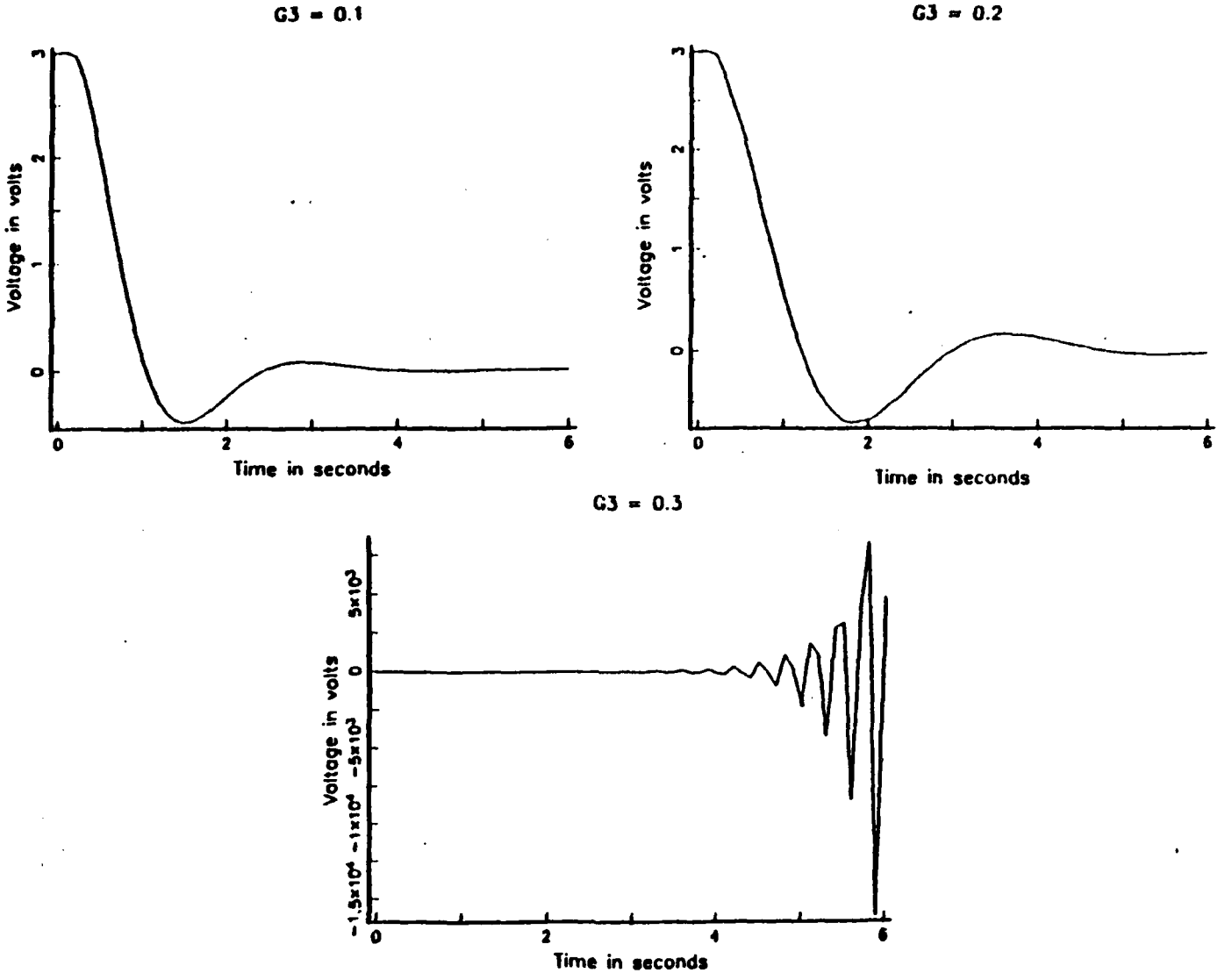


Figure 3.18 Simulation Result with Compensation (Td = 0.1 sec, Ka=1).

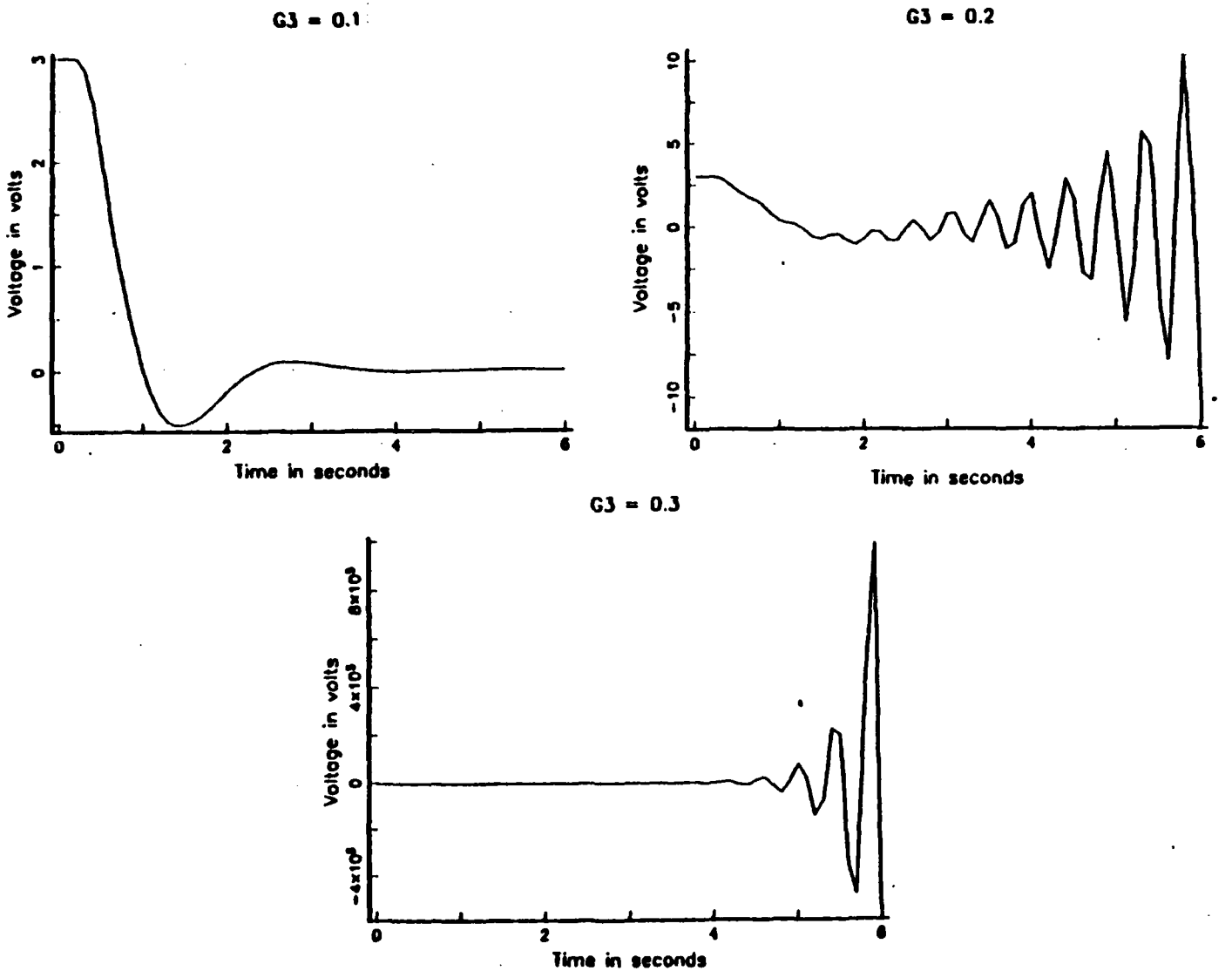


Figure 3.19 Simulation Result with Compensation (Td = 0.15 sec, Ka=1).

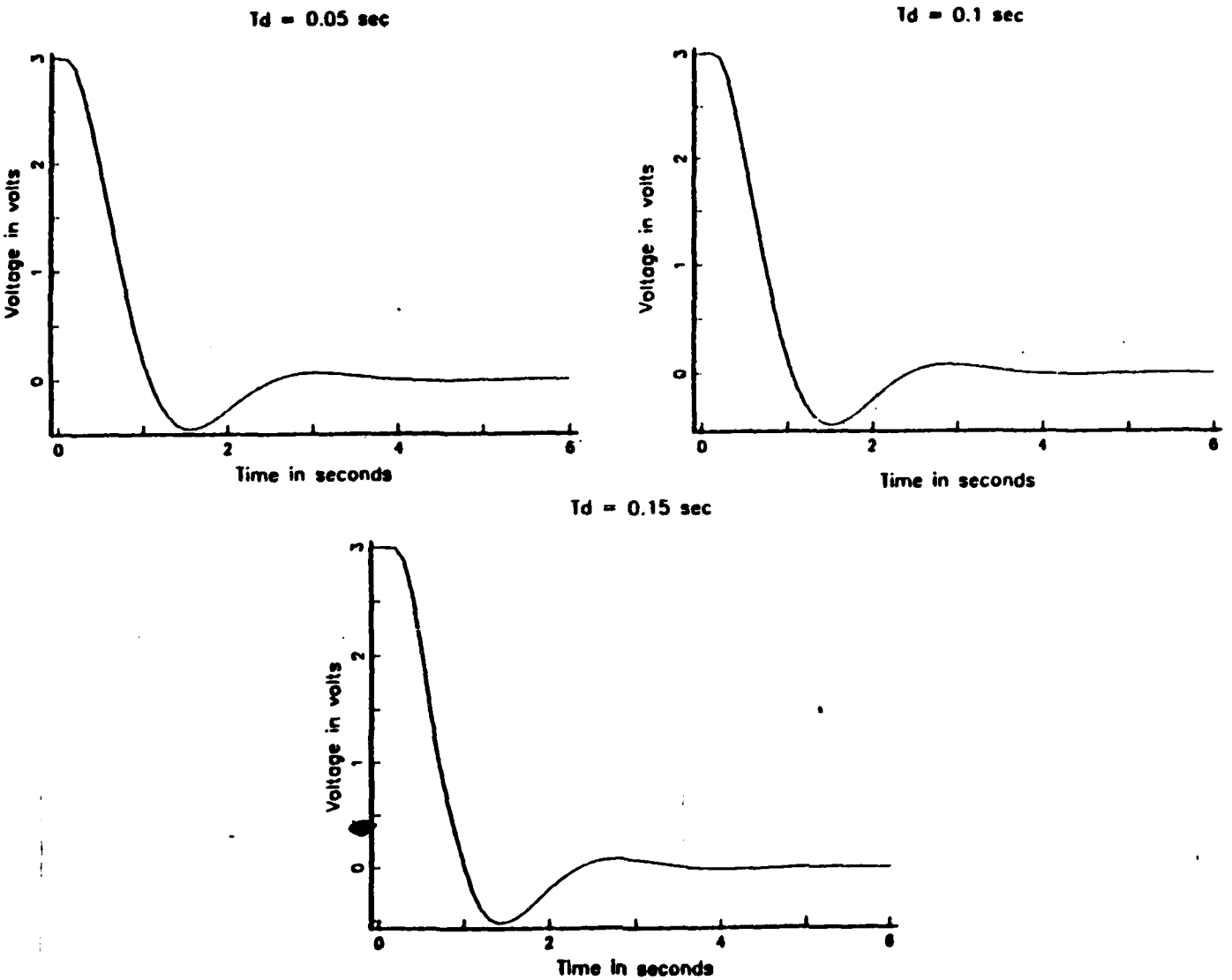


Figure 3.20 Simulation Result with Compensation (g = 0.1, Ka = 1).

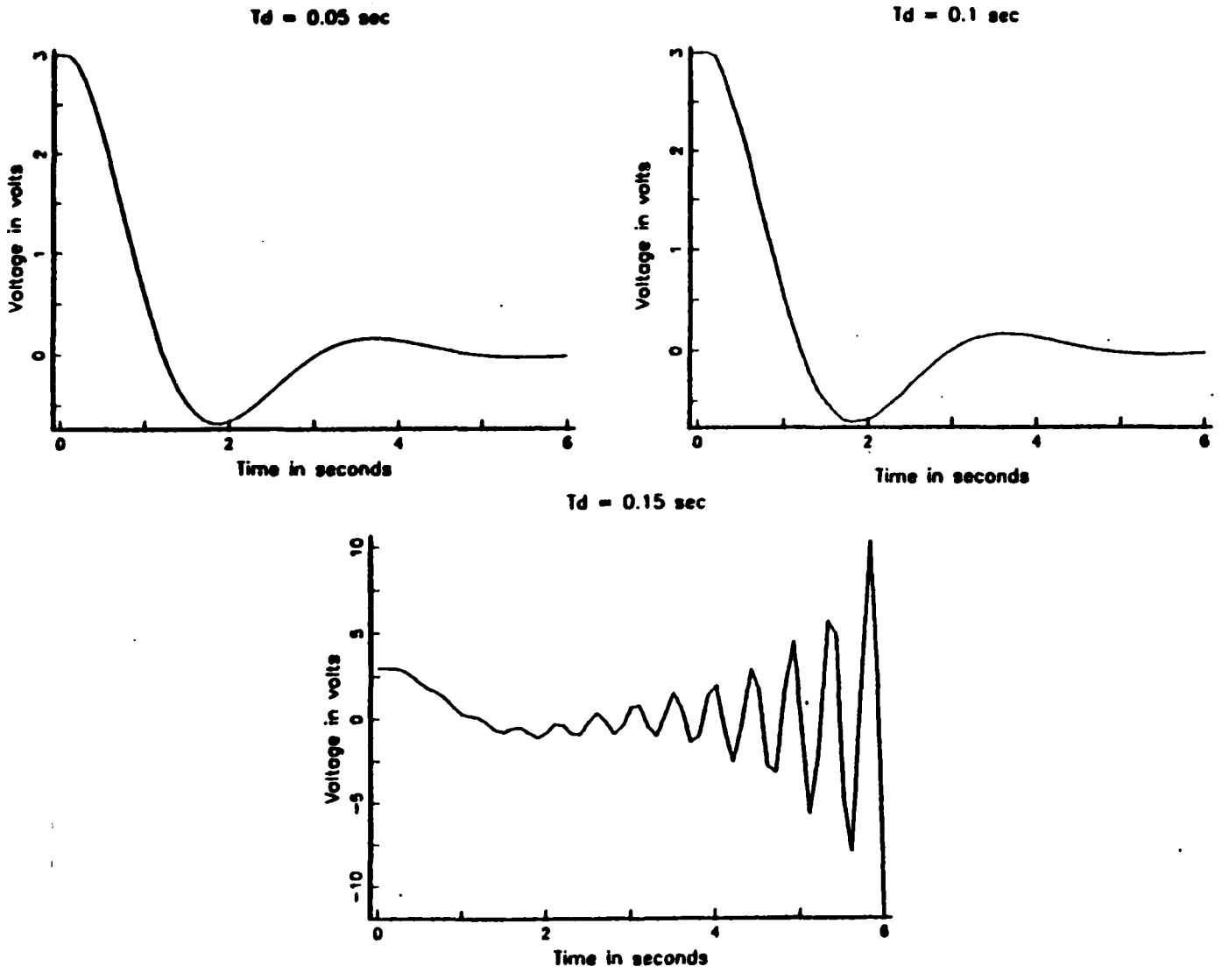


Figure 3.21 Simulation Result with Compensation (g = 0.2, Ka = 1).

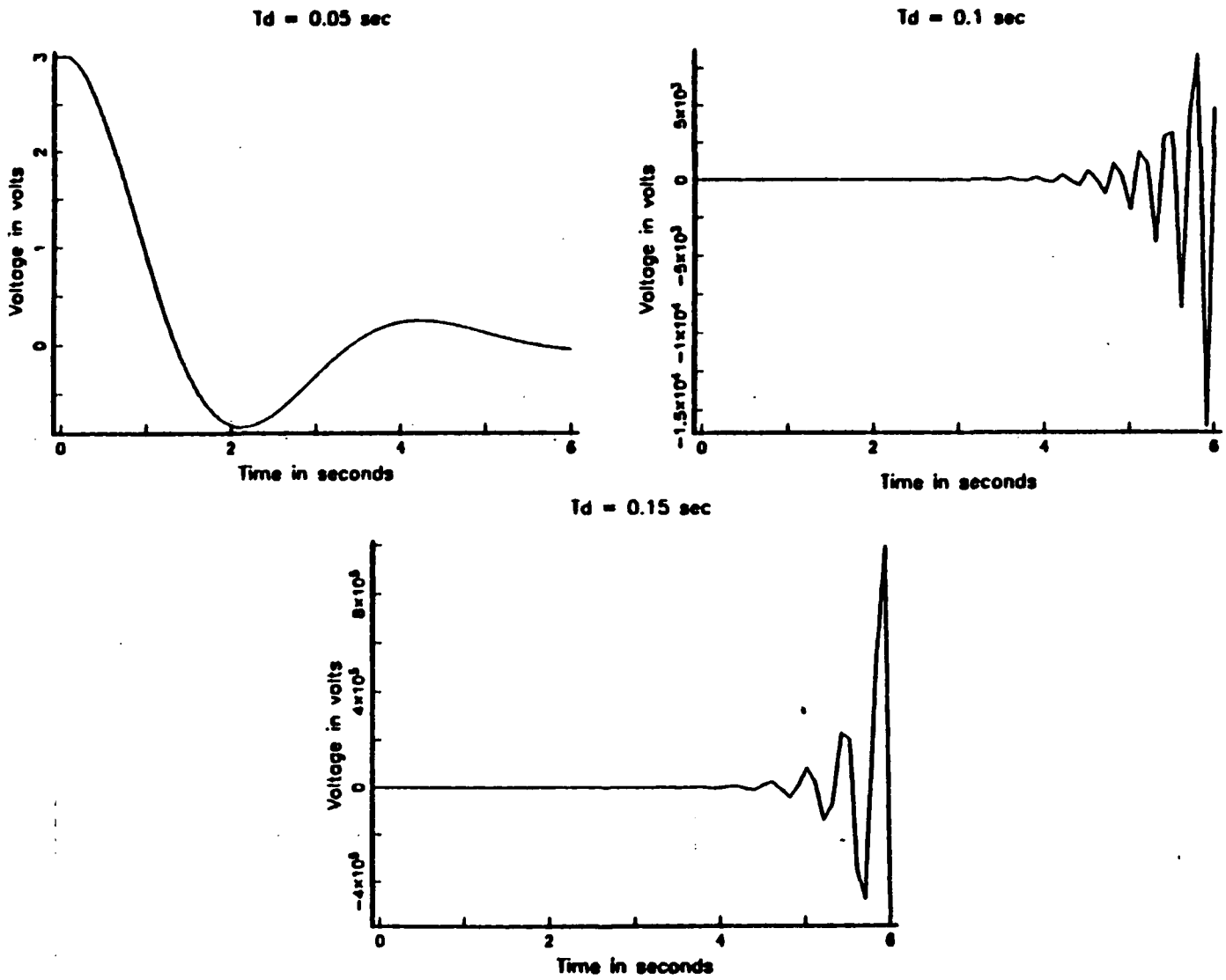


Figure 3.22 Simulation Result with Compensation (g = 0.3, Ka = 1).

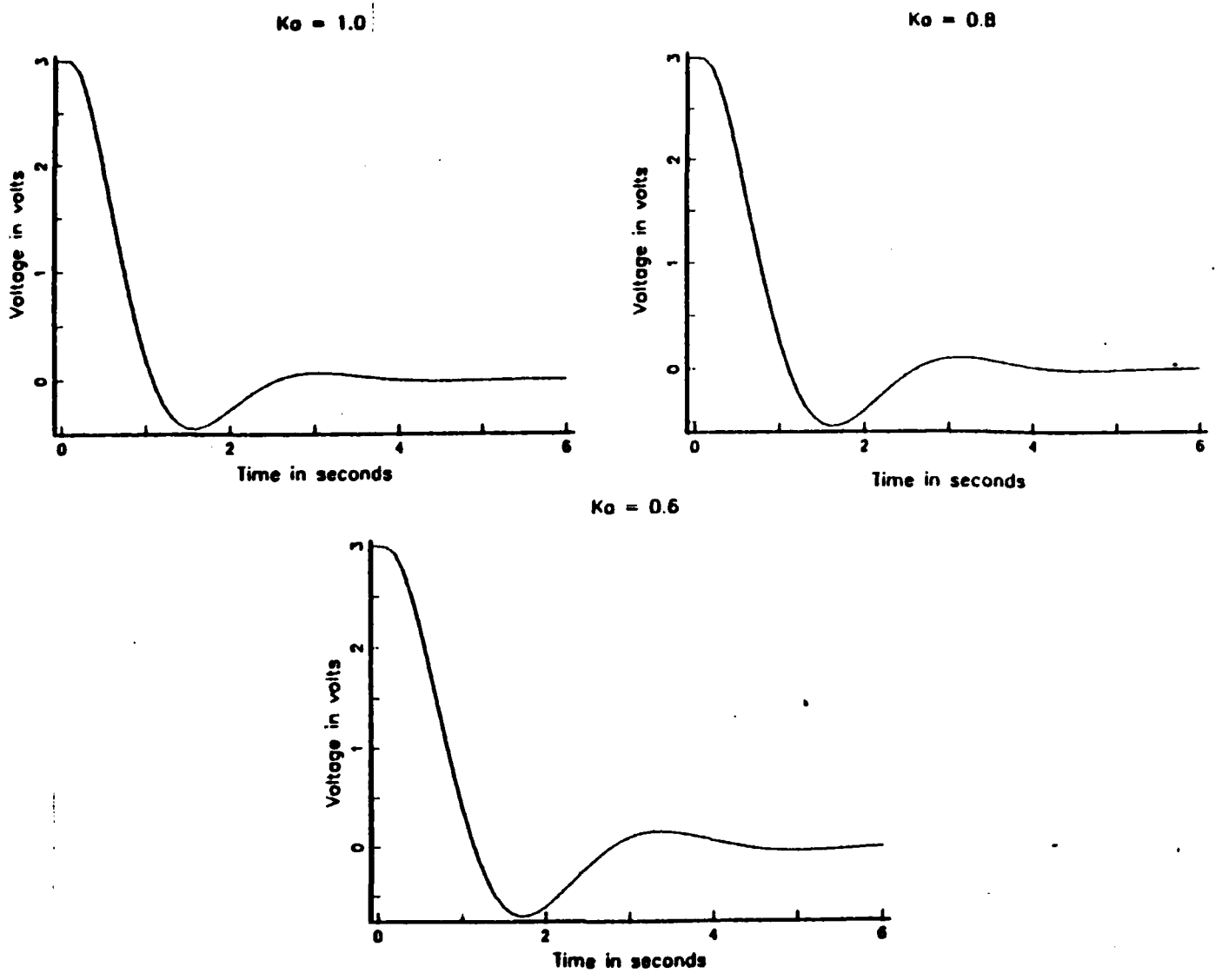


Figure 3.23 Simulation Result with Attenuation (Td = 0.05 sec, g =0.1).

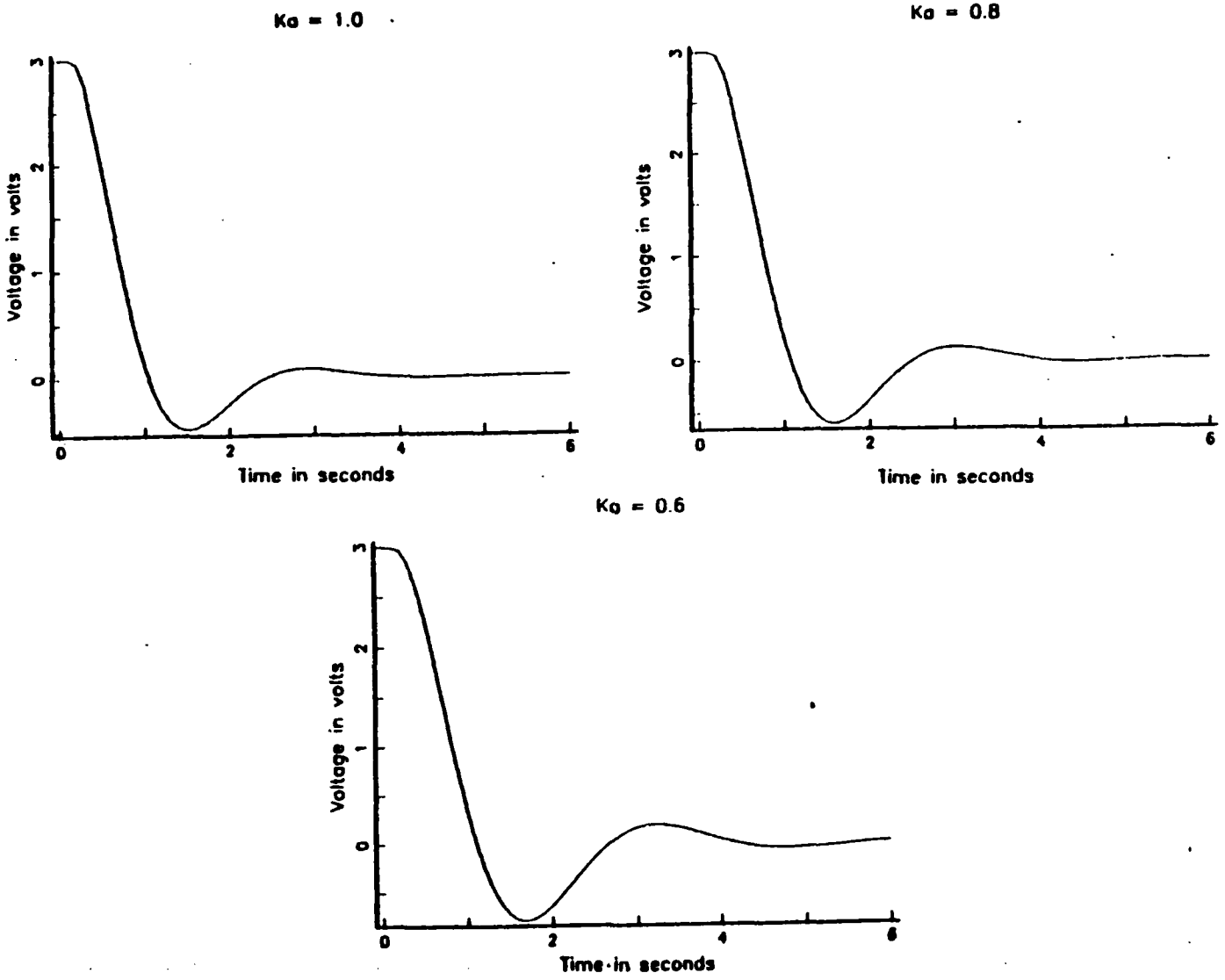


Figure 3.24 Simulation Result with Attenuation (Td = 0.1 sec, g =0.1).

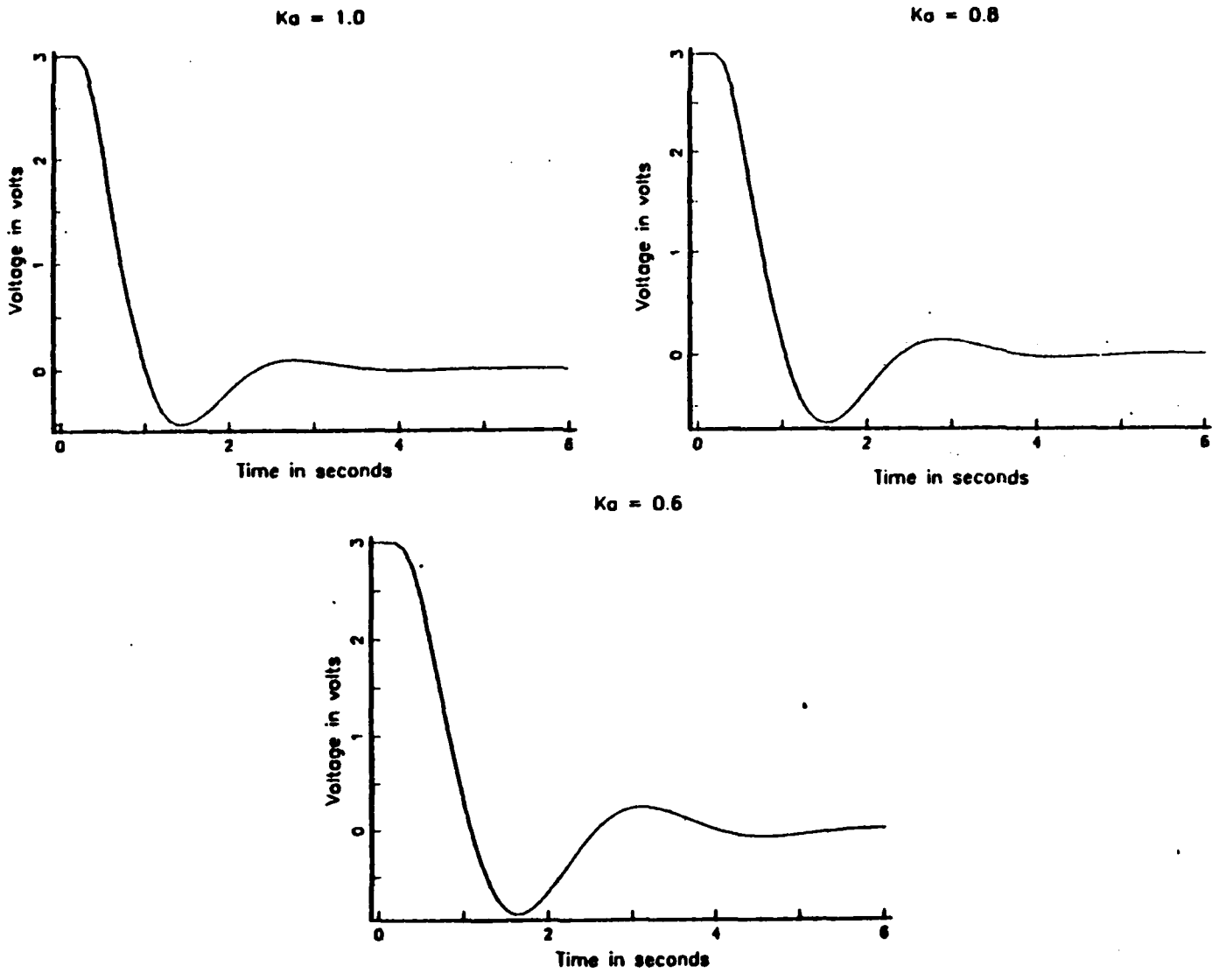


Figure 3.25 Simulation Result with Attenuation (Td = 0.15 sec, g =0.1).

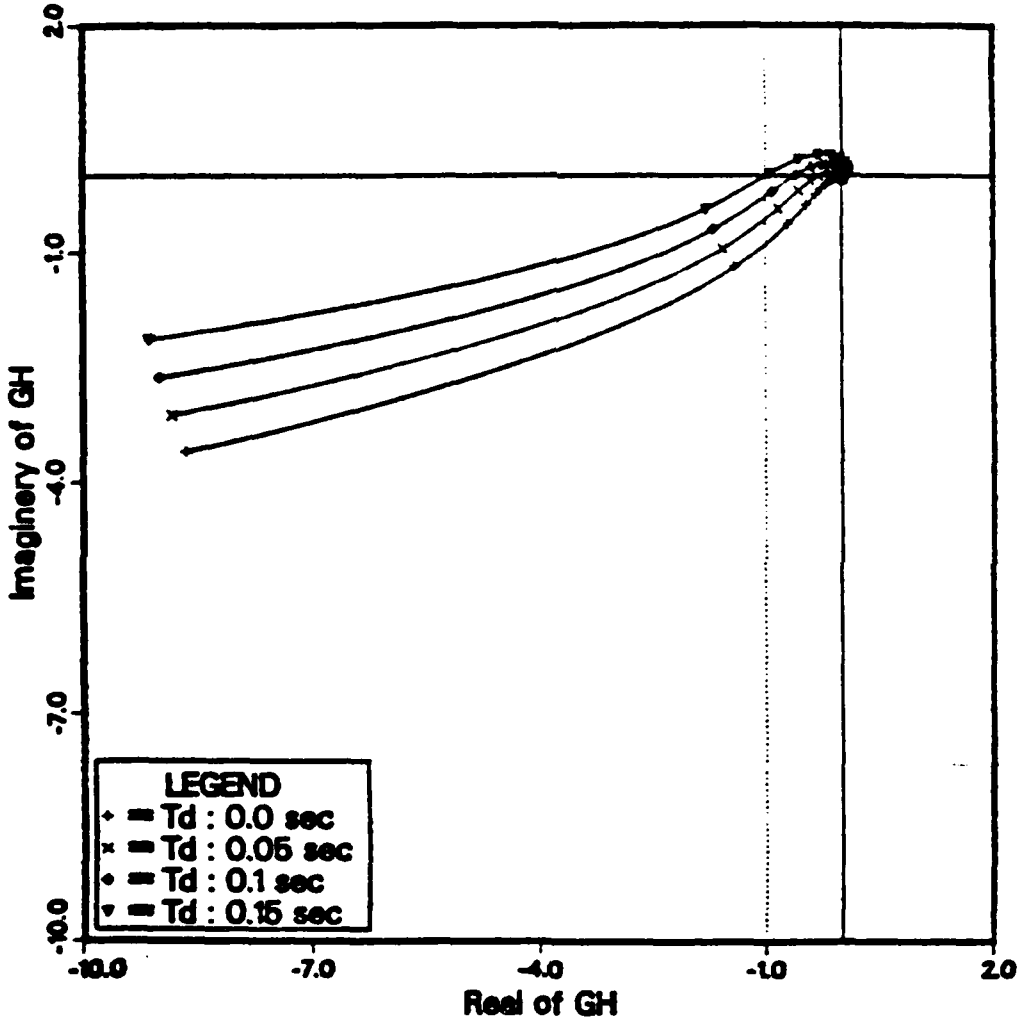


Figure 3.26 Nyquist Plot of Delayed System.

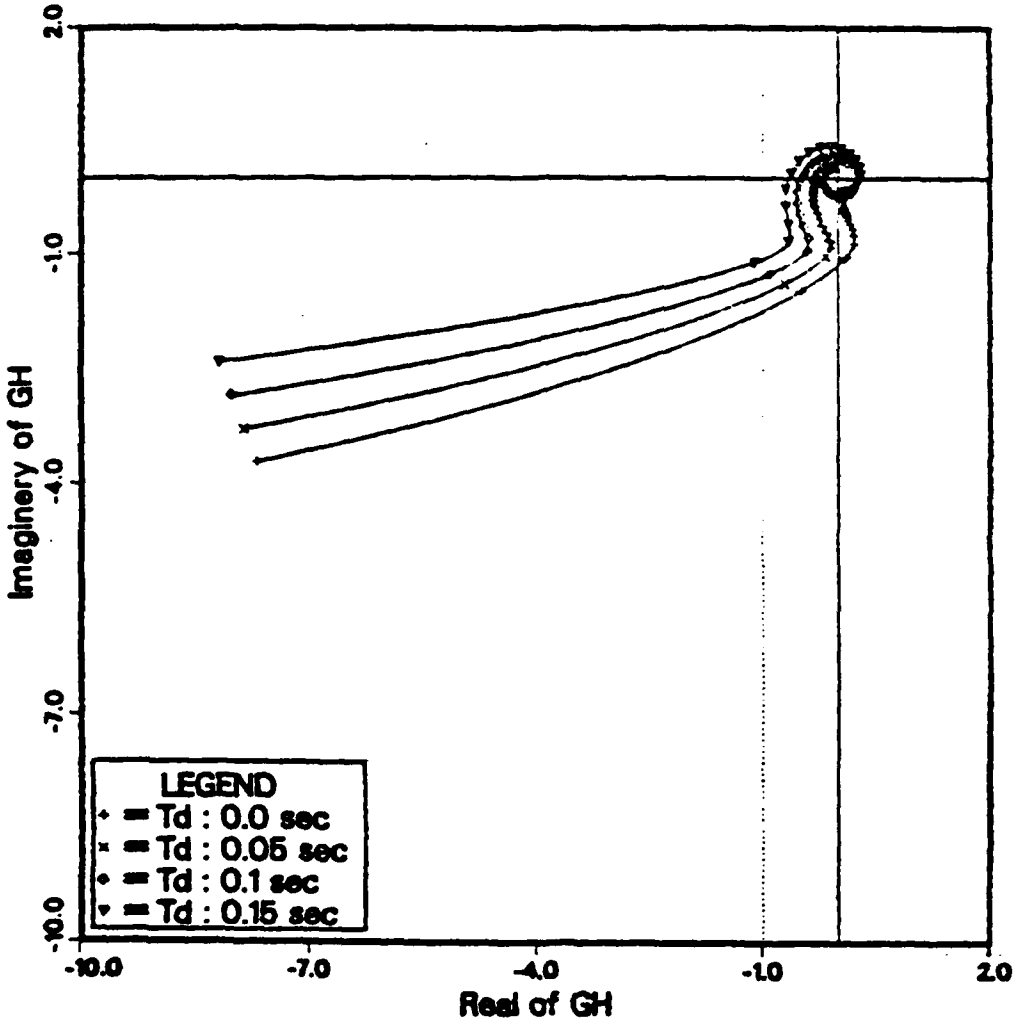


Figure 3.27 Nyquist Plot(g =0.1, Ka=1).

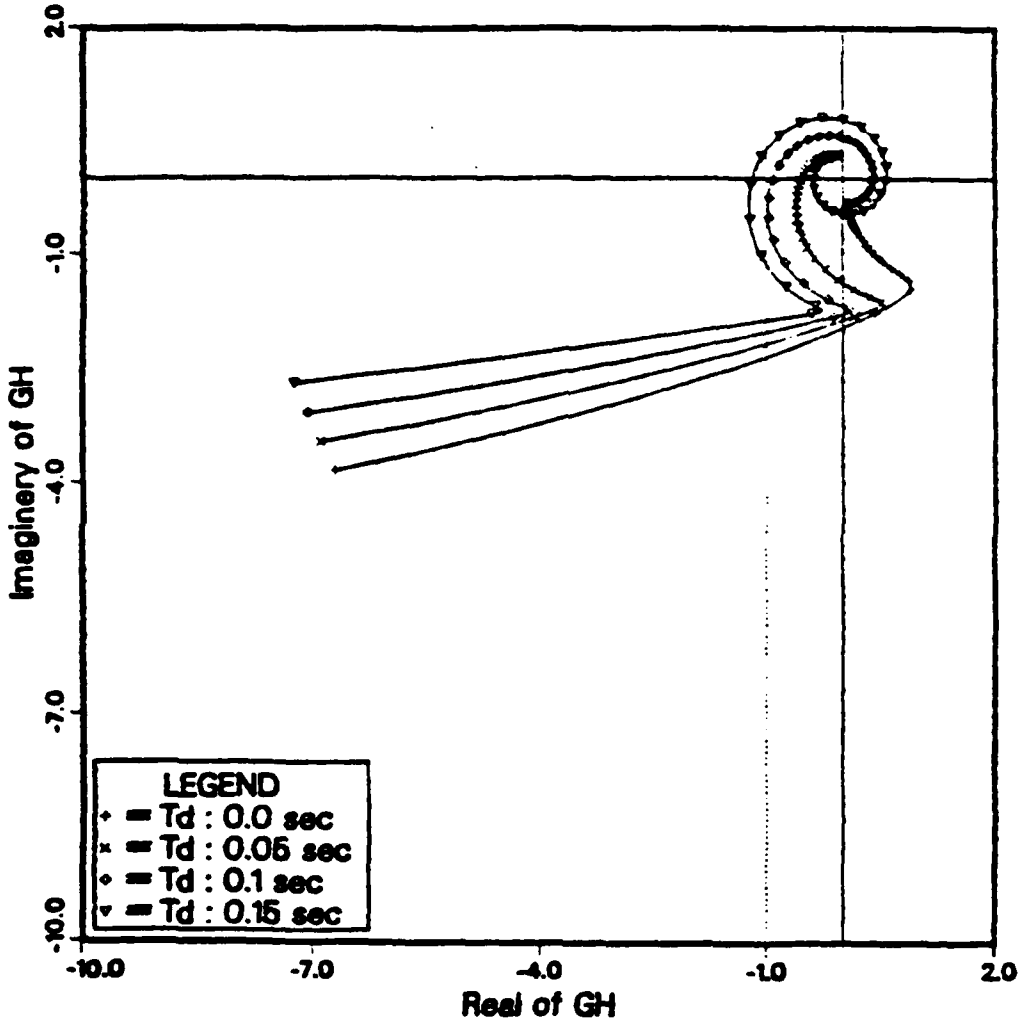


Figure 3.28 Nyquist Plot(g =0.2, Ka=1).

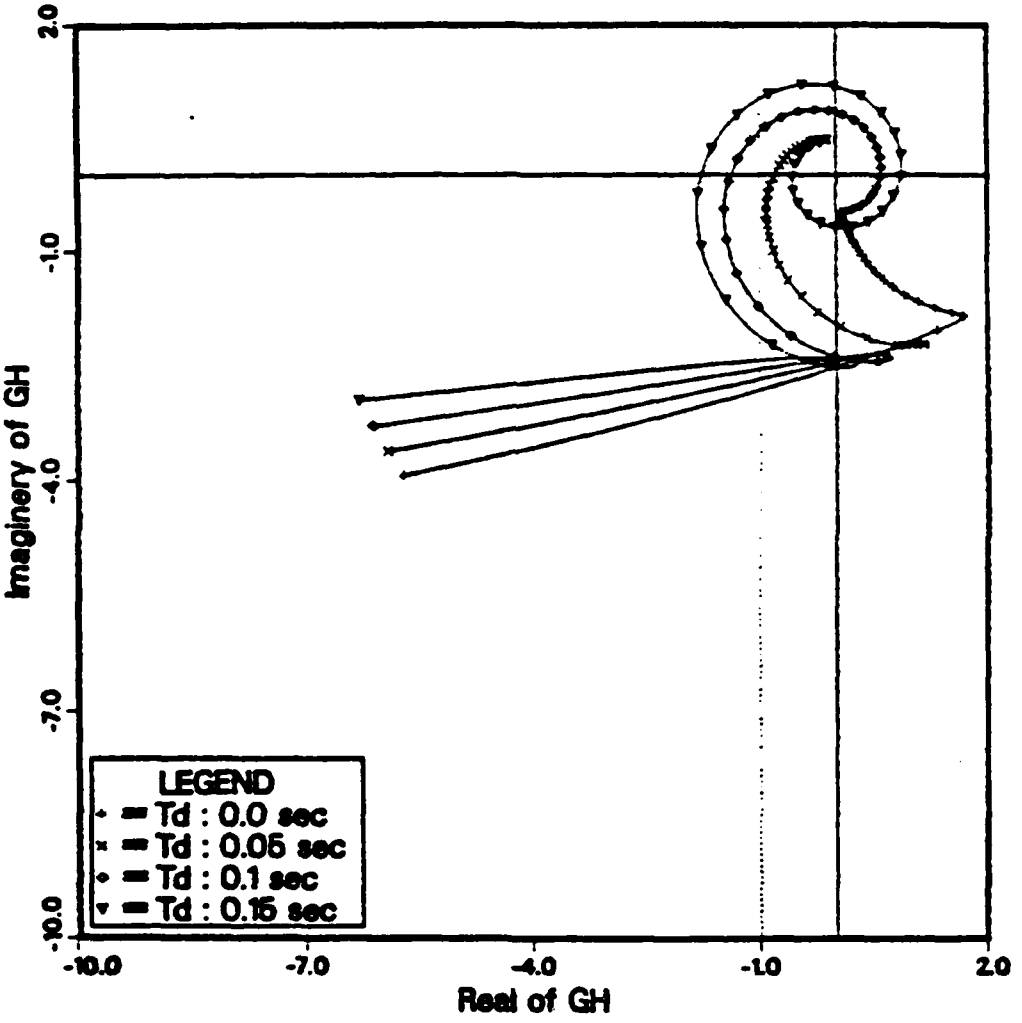


Figure 3.29 Nyquist Plot(ζ = 0.3, Ka=1).

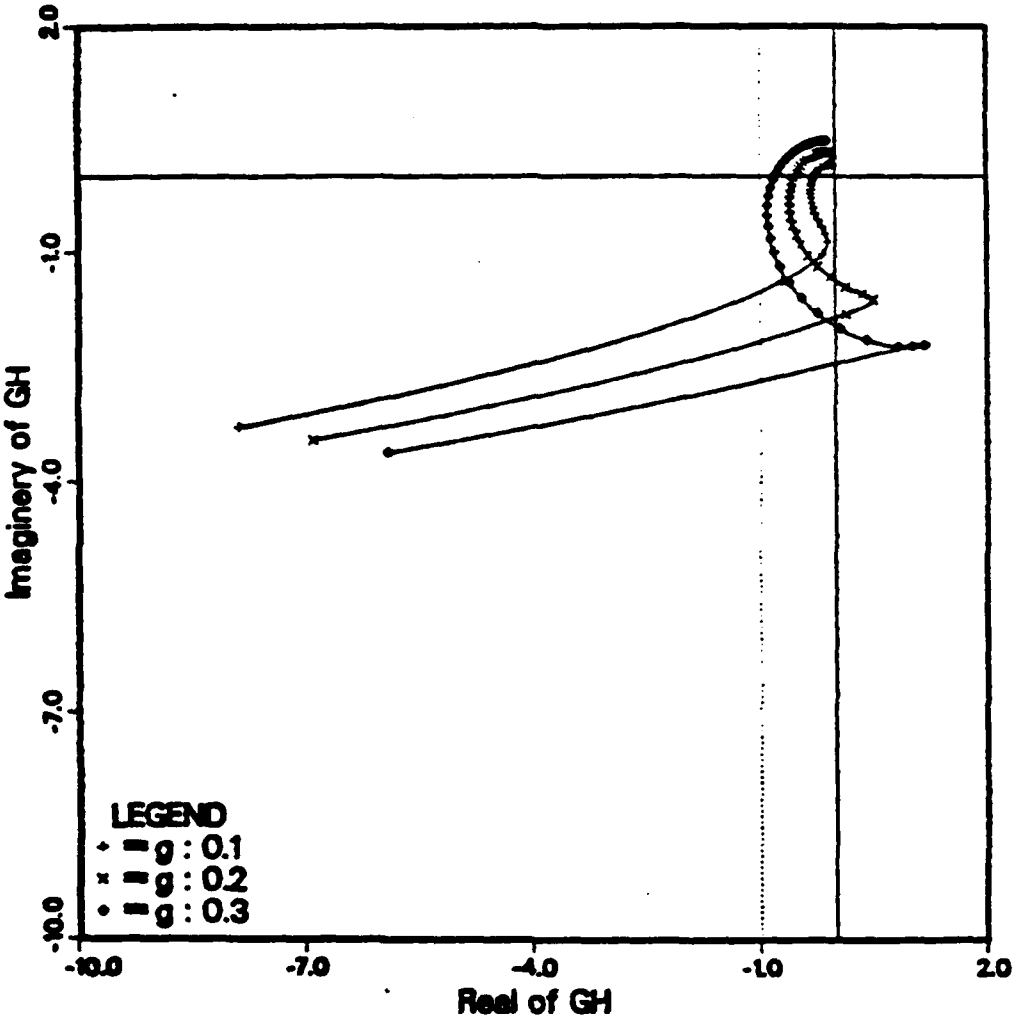


Figure 3.30 Nyquist Plot(Td = 0.05 sec, Ka=1).

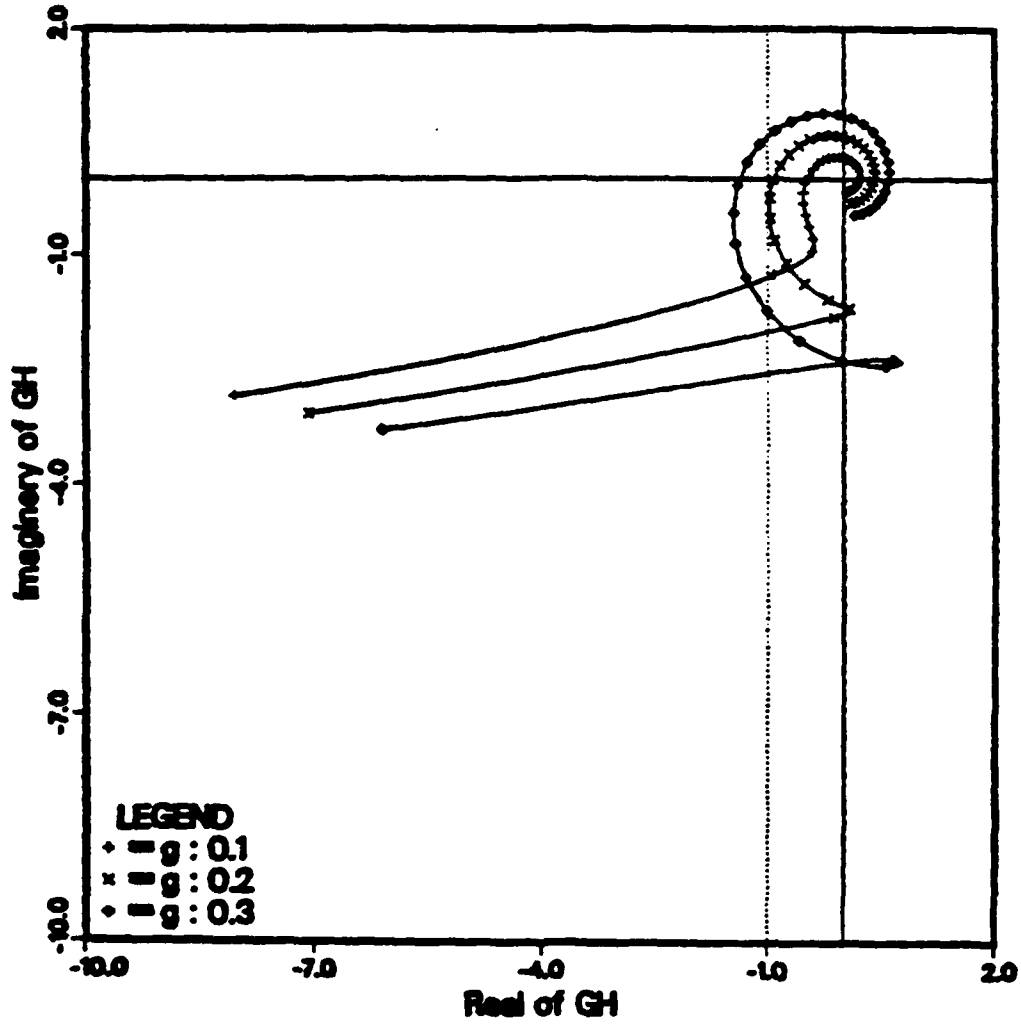


Figure 3.31 Nyquist Plot (Td = 0.1 sec, Ka=1).

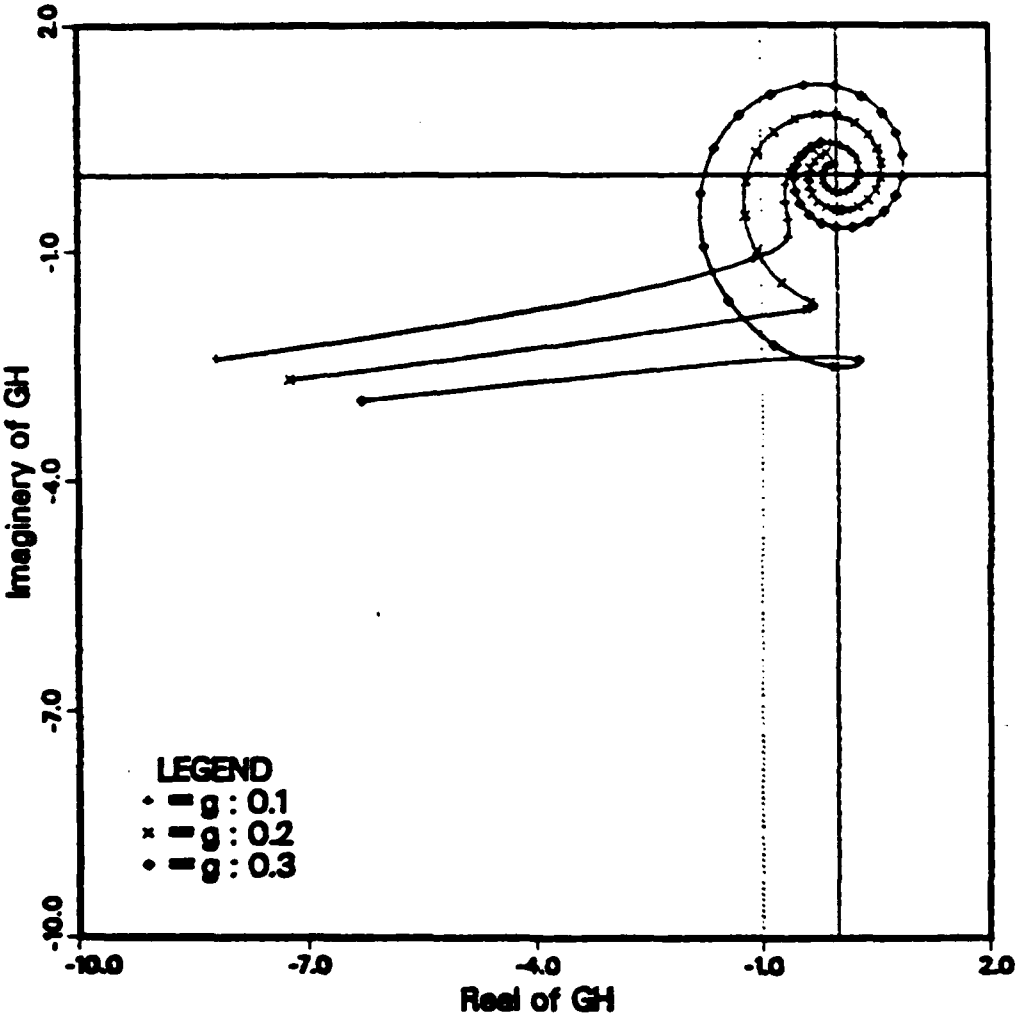


Figure 3.32 Nyquist Plot(Td = 0.15 sec, Ka=1).

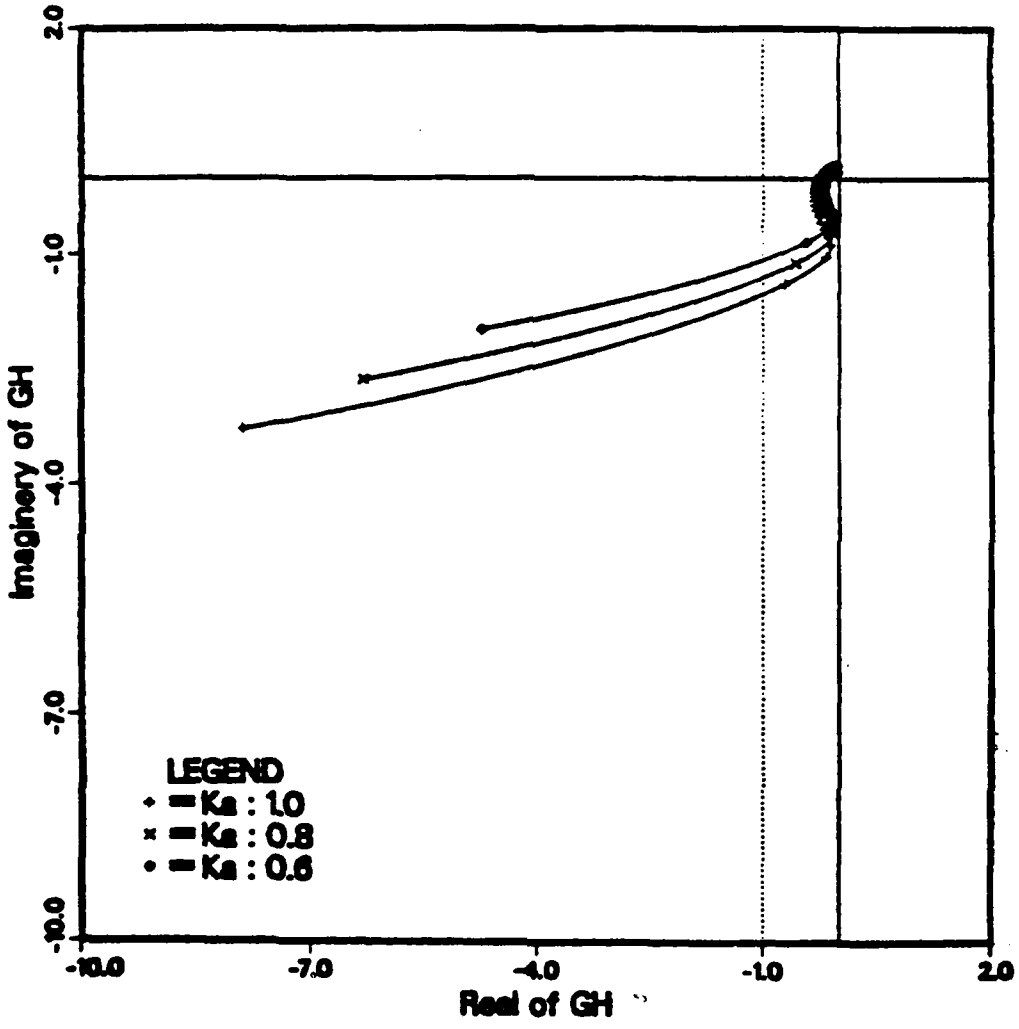


Figure 3.33 Nyquist Plot (Td = 0.05 sec, g = 0.1).

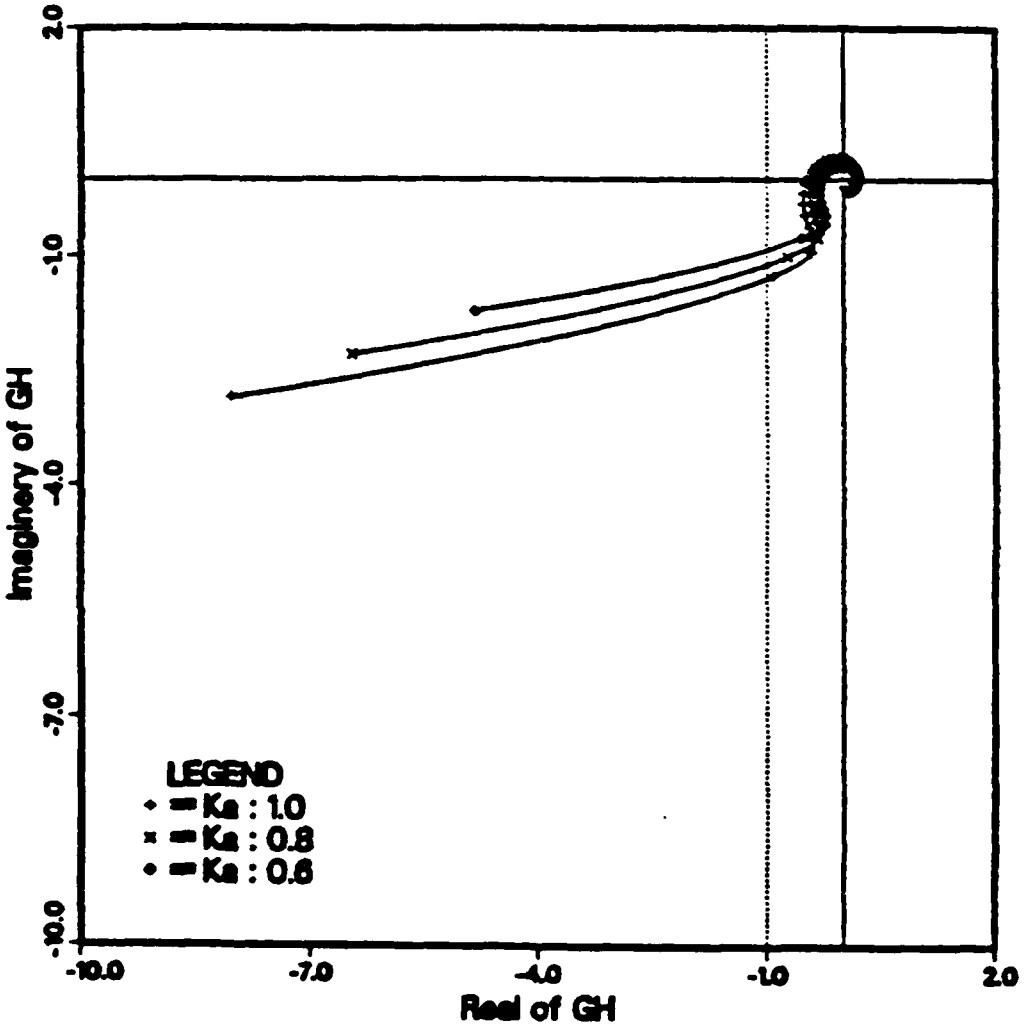


Figure 3.34 Nyquist Plot(Td = 0.1 sec, g =0.1).

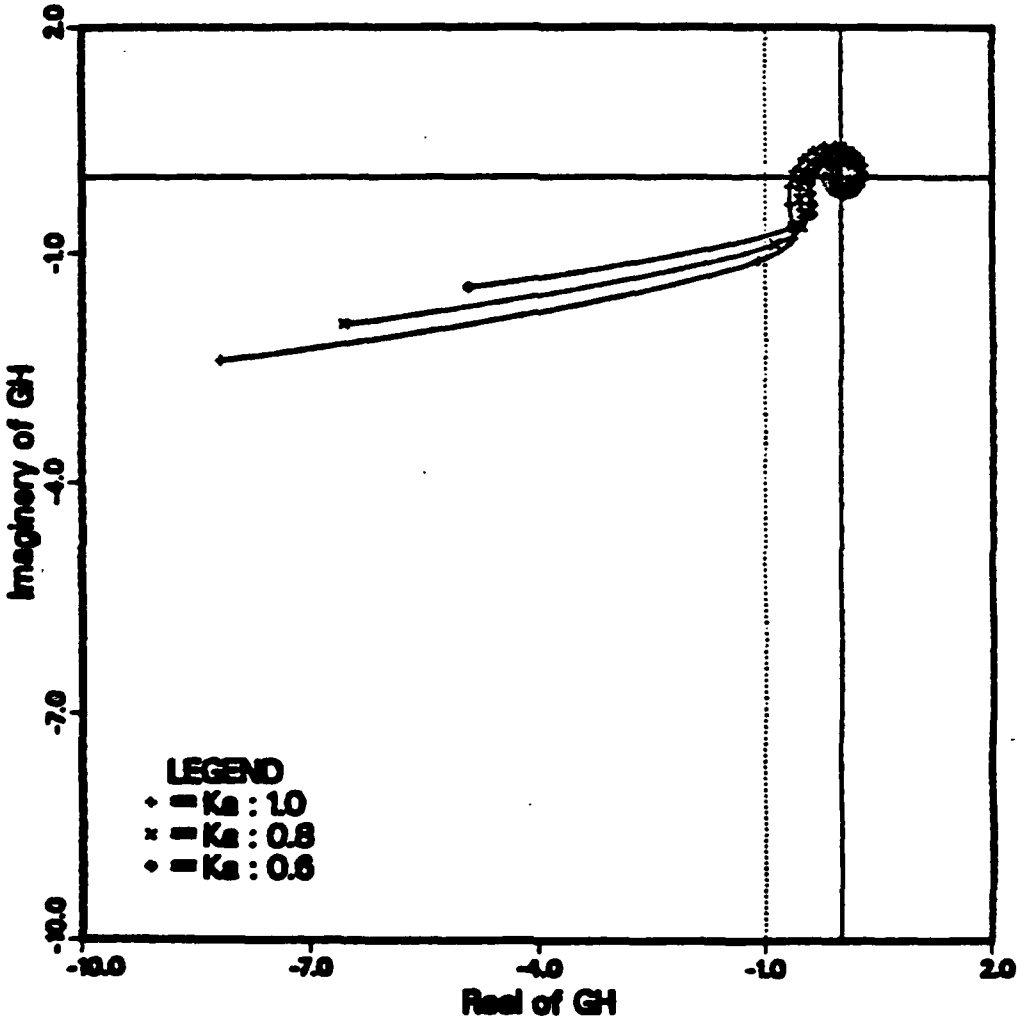


Figure 3.35 Nyquist Plot(Td = 0.15 sec, g =0.1).

IV. EMPIRICAL ANALYSIS

A. INTRODUCTION

Based upon the theoretical analysis of the design method, measurements were performed to demonstrate that the system design using a microcomputer controller can be applied to an analog(Rep-op mode) computer plant. The analog computer can simulate the physical system with wide bandwidth, i.e., high performance system. The description of the analog computer is represented in Ref. 5. The digital controller is implemented by using the Feedback Inc. MAT385, microprocessor-based computer [Ref. 6,7,8,9]. Since the characteristics of the microcomputer is important in designing the digital controller, it will be described in more detail in the next section. The A/D and D/A converters are built into the MAT385 microcomputer and provide single-input single-output signal ports for the controller. Because of the restrictions imposed by hardware limitations such as single-input/single-output and software considerations such as Assembly language programming, certain special programming techniques were required.

B. MICROPROCESSOR CHARACTERISTICS

1. MAT385 Hardware

A typical microcomputer system comprises a microprocessor(the CPU), a selection of memory circuits and some input-output(I/O) ports. Typically the memory is made up of some read-only memory(ROM) to hold the fixed application program and some random-access(read/write) memory(RAM) for temporary data storage.

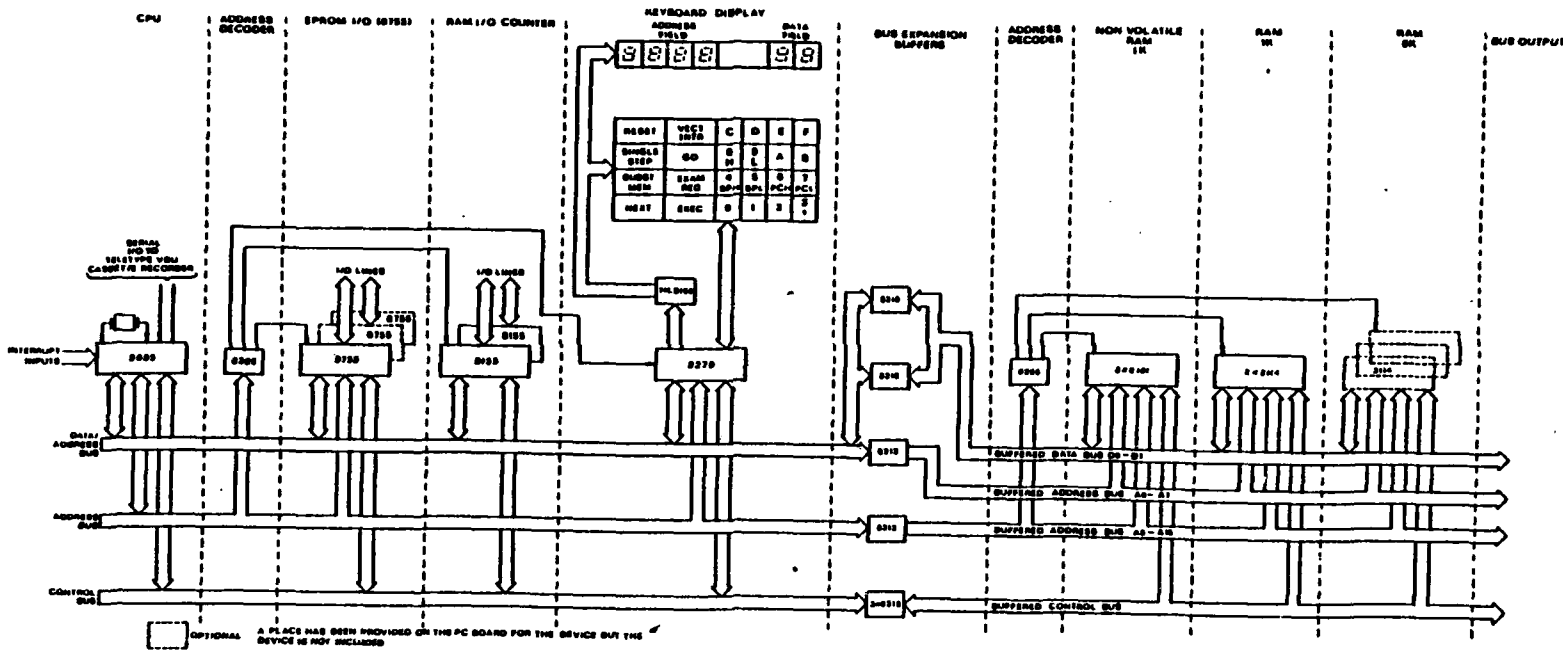


Figure 4.1 The MAT385 Functional Block Diagram.

During the development of a program, however, it is advantageous if the program is initially stored in RAM so that it may be readily changed if any errors are found when the program is executed. Since a typical program(machine) instruction takes only a few microseconds to execute, it is also helpful and instructive if, during program development, the execution of the program can be controlled.

The MAT385 microprocessor-based system used in this experiment has been designed to enable the user to write and store a program in memory and readily follow and monitor the state of the complete system while the program is being executed.

The MAT385 is a self-contained microcomputer system designed around the INTEL 8085 microprocessor and its family of peripheral components. Figure 4.1 is a functional block diagram of the MAT385.

The 8085 CPU and The System Buses

The 8085 CPU is an evolutionary enhancement of INTEL's industry-standard 8080A. It is 100% software compatible with the 8080A while offering the benefits of single power supply, higher integration, higher performance, and improved system timing.

As the system block diagram shows, the 8085 derives its timing inputs directly from a crystal. In addition the 8085 drives the system with control signals available on-chip. No addition status decoding circuitry is required for most small-to-medium sized systems. The 8085 multiplexes its data bus with the low 8 bits of its address bus. The 8155 and 8355/8755 Memory I/O components are designed to be compatible with this bus structure, precluding the need for external bus latches.

TABLE II

Input/Output Ports of MAT385

	Port Address	Function	Dedication
8755 Monitor ROM (BK)	00	Port A (8 bit I/O)	Front Panel Socket
	01	Port B (8 bit I/O)	
	02	Port A Data direction register	
	03	Port B Data direction register	
8155 RAM (BH)	20	Command/Status register	
	21	Port A (8 bit I/O)	Binary or analogue input
	22	Port B (8 bit I/O)	Binary & analogue output
	23	Port C (6 bit I/O or Control)	Binary I/O control
	24	Timer count low order byte	Keyboard monitor-single step
	25	Timer count high order byte	
8155 RAM (BJ)	28	Command/Status register	
	29	Port A (8 bit I/O)	Front Panel Socket
	2A	Port B (8 bit I/O)	
	2B	Port C (6 bit I/O or control)	
	2C	Timer count low order byte	Counter output on front panel socket
	2D	Timer count high order byte	

Four vectored interrupt inputs are available in addition to the standard 8080A type interrupt. There is also a serial input and serial output data line pair that is exercised under program control to provide the MAT385s simple teletype/VDU/cassette I/O.

The basic clock frequency of the 8085 in the kit is 3.072 MHz.

The 8155

The 8155 is a highly integrated IC designed for compatibility with the 8085's bus structure. It contains 256 bytes of static RAM, 22 programmable I/O lines, and a 14 bit timer/counter.

Two 8155's are used on the MAT385. On one the RAM is available for storage of user programs as well as for temporary storage of information needed by system programs, and the timer is used by the MAT385 monitor's Single Step routine to interrupt the processor following the execution of each instruction. The programmable I/O lines(Port 21 and 22) are used by the binary and analog interfaces. The second 8155 has all its RAM memory available for storage of user programs and the I/O lines(Port 21,22 amd 23) and the timers are available for user applications through a socket on the front panel.

The 8755

The 8755 is an EPROM; it contains 2048(2K) 8bit words(bytes) of memory and 16 I/O lines. The MAT385 is supplied with one 8755 which is programmed with the system monitor program, the I/O lines(port 00 and 01) are available for user applications through a socket on the front panel.

Sockets are provided on the PWB for two further 8755's one of which may be programmed with a 'Tiny Basic' interpreter program, and the other with user subroutines.

The 8279

The 8279 is a keyboard/display controller IC that handles the interface between the 8085 and the keyboard and LED display. The 8279 refreshes the display from an internal memory while scanning the keyboard to detect keyboard inputs.

The 8205

The MAT385 also contains three 8205 IC's (one-out-of-8 decoder) that decode the 8085's memory address bits to provide chip enables for the memories.

Main Memory

The main memory has 2K bytes. The lower 1K bytes is nonvolatile so that data can be written into this memory and it will not be destroyed when the power is switched off. This is achieved by using CMOS memory IC's which consume a very low current and using a battery to maintain the power to the IC's when the MAT385 is disconnected from the mains. The CMOS memory is a 5101, each of which stores 256 X 4 bits; eight are used to make up the 1K bytes. The second 1K bytes of RAM uses two 2114 IC's each of which has a capacity of 1K X 4 bits.
This memory is volatile.

The main memory may be extended to 8K bytes by inserting further 2114 IC's in the spare positions provided on PWB.

Monitor reserved RAM locations

The RAM whose address starts at 2000 is partly used by the Monitor for storage. The locations below 20C8 are normally used for the Stack. It is not possible to define the number of locations used as it depends on the number of

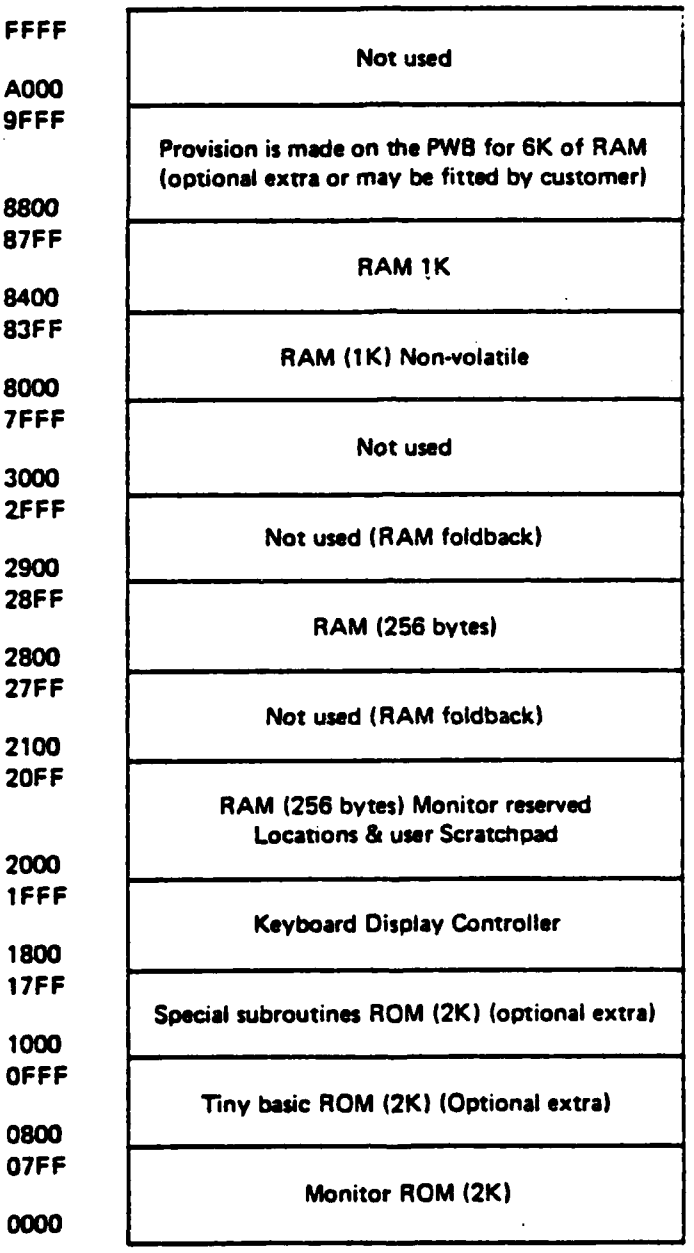


Figure 4.2 The MAT385 Memory Map.

subroutines that may be called at one time. The allocation of memory addresses(the memory map) in the system is shown in Figure 4.2 . The Table II lists the port addresses and functions.

Analog Input

The MAT385 microprocessor-based computer has A/D and D/A converters which are necessary for the design of a digital controller. The analog input is bipolar and may be set to any sensitivity up to a maximum of +200V. The input is set up on test of 35mV per binary step. This gives a full scale sensitivity of +4.445V to -4.48V and is used to give compatibility with the analog output which is limited +5V.

The zero voltage in is set to give a digital reading of 80H(10000000). The positive voltages give a digital reading from 80H up to FFH, the negative voltages give a digital reading from 80H down to 00H. The clock speed of the A/D converter is set to 1.536MHz which gives a maximum conversion time of 200 microseconds.

Three control signals from Port 23 control the operation of the A/D converter.

Port 23-4 : Must be '1' to enable data from A/D converter to Port 21

Port 23-5 : Must go to '1' and then to '0' before the conversion will start

Port 23-2 : Goes to '0' when conversion complete. The port may be sampled to confirm that the conversion is complete before the program continuous.

Analog Output

The digital output from Port 22 is fed via gates to the D/A converter. The gates and therefore the analog

output are enabled when Port 23-3 is '1' and the output relays are disabled. The analog output has a range of +5 V to -5 V.

2. Static and Dynamic Conversion Test

Basic experiments are performed to test static and dynamic conversion for confirmation of the A/D and D/A conversion characteristics. This A/d and D/A converters are built in MAT385. MAT385 used successive approximation A/D conversion technique that is widely used when fast conversion is required. The basic technique is one of successively approximating a value for the digital output, altering this value each time in such a way as to approach the correct output. The converters that employ hardware to implement this successive approximation algorithm are widely available(conversion times range from submicroseconds to 200 microseconds). It is possible however, to use a microcomputer program to implement this algorithm and implement A/D conversion using a D/A converter and a comparator. An outline of the hardware for an 8-bit converter is shown in Figure 4.3 .

Since A/D converter may take a relatively long time to perform a conversion operation, after starting a conversion it is necessary for the programmer to decide what action the program should take while the conversion is being performed. There are three possibilities.

(1) After a conversion is started, the program can enter a delay loop which is equal to or greater than the conversion time of the A/D converter. The conversion time is

$$\frac{1}{\text{A/DC clock freq.}} \cdot 255s = \frac{255}{1.536 \times 10^6} = 160 \text{ microseconds}$$

The delay should therefore exceed 160 microseconds.

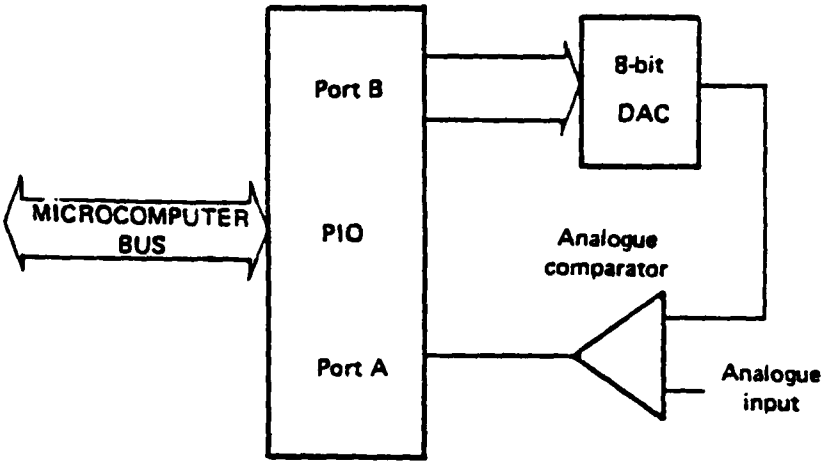


Figure 4.3 Hardware for the Software-Based A/D Converter.

(2) The A/D converter generates a 'conversion-complete' signal after conversion has been performed and the program loops until this signal is detected. This technique used on our experiment.

(3) After starting the conversion, the program continuous and the 'conversion-complete' signal generated by the A/D converter is used as an interrupt input.

A fixed analog input and ramp analog input are used in static and dynamic test, respectively. In the static test, the resolution of the output to input is almost one which means input and output voltages are almost the same. But, if the input voltage exceeds 4.45 V, the output is zero. The conversion test diagram for both experiments is shown in Figure 4.4 . The input and output comparision are shown in Figure 4.5 and Figure 4.6, The result of the dynamic test is identical to that of the static test.

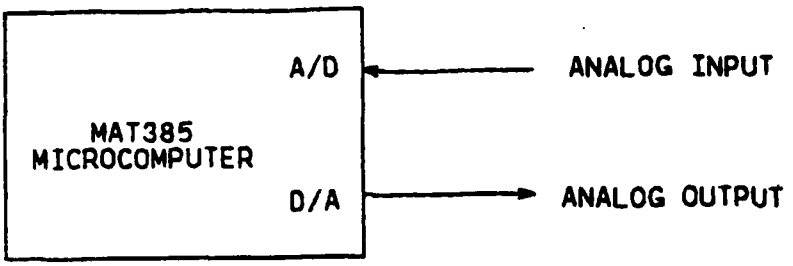


Figure 4.4 Conversion Test Block Diagram.

3. Sampling Period Test

The sampling period of discrete data can be measured by using the signal generator and oscilloscope. MAT385 has a DELAY subroutine which can be used for delay of program execution time in ROM space [Ref. 7]. Using the DELAY subroutine, execution time can be changed, thus sampling time can be adjusted in the control algorithm. The range of delay factor is from 0001H(1) to FFFFH(65535). The control algorithm can be delayed as much as 6 NOP operations by one step of delay factor. For some delay factors to be chosen, each sampling period was measured as Table III .
In the Equation 2.8 , the control algorithm execution time could be calculated by the following.

$$T_{ex} = T_{mp} + T_{da}$$

where T_{mp} : Main program execution time

T_{da} : DELAY algorithm execution time

Figure 4.7 shows the analog sinusoidal wave input and discretized output for the case of 01H delay factor. For

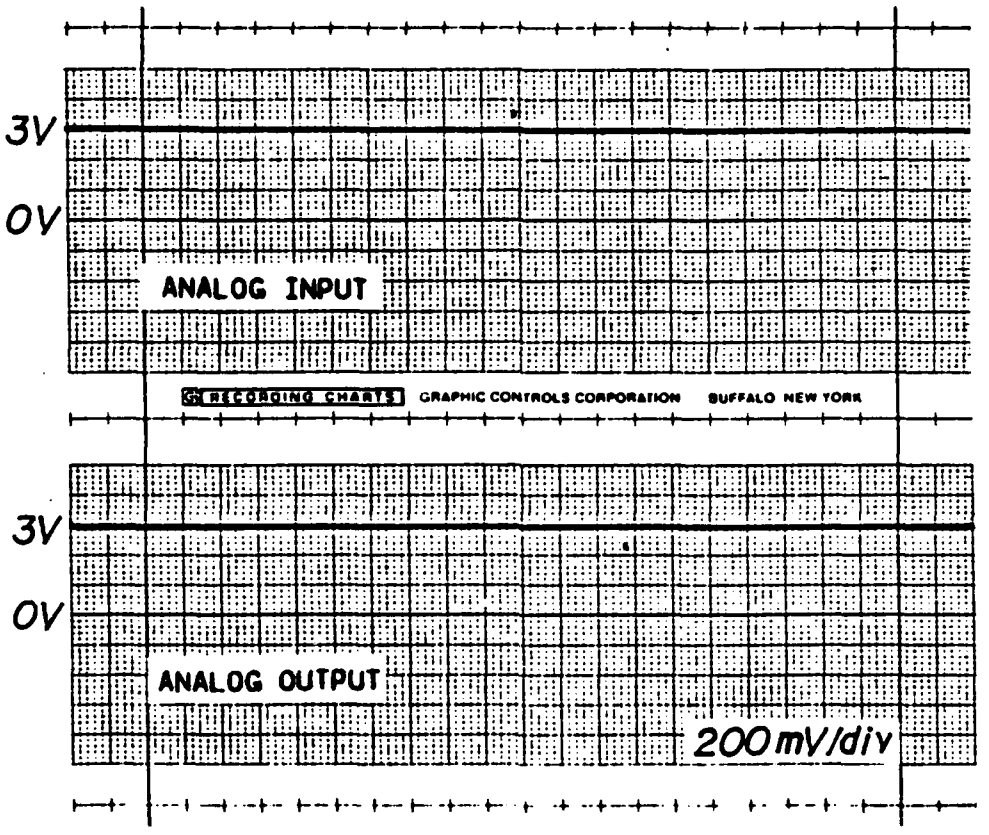


Figure 4.5 Result of Static Conversion Test.

implementation of these tests which are static, dynamic and sampling period test, the small program of test algorithm is necessary. That program is written in Appendix B.

C. DESIGN OF STATE FEEDBACK CONTROL SYSTEM WITH MICROCOMPUTER

1. Control Algorithm

The finite precision nature of the digital hardware makes it necessary to choose a computational structure that will perform adquately with regard to the initial objectives

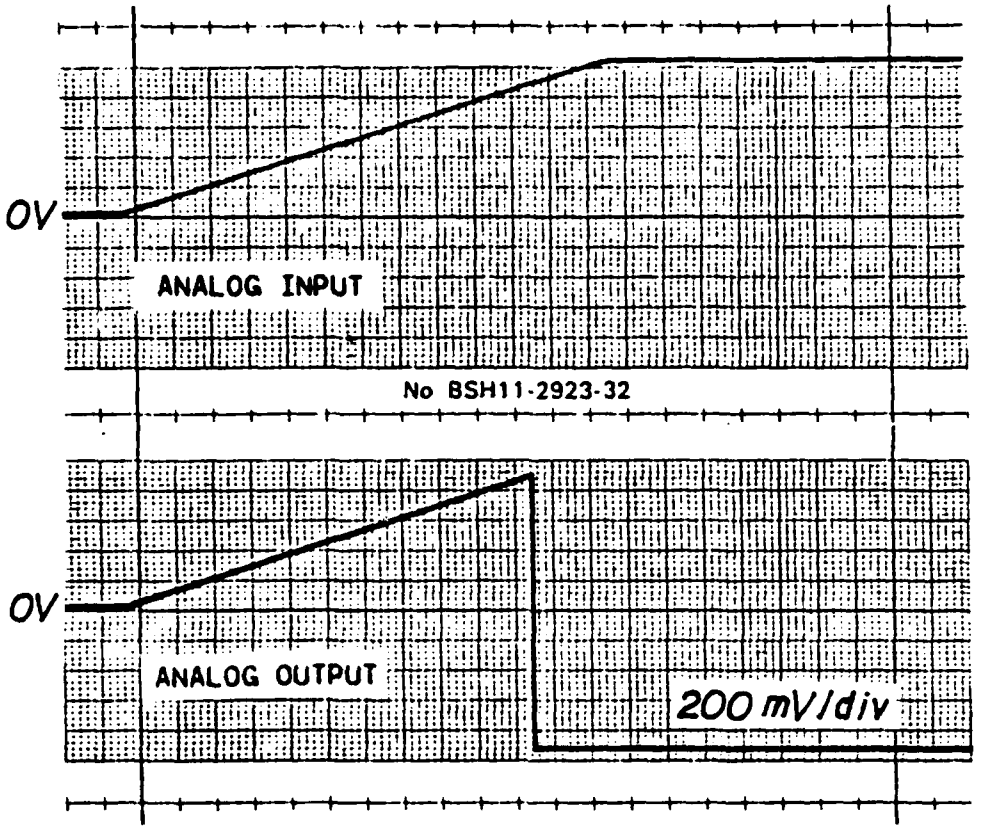


Figure 4.6 Result of Dynamic Conversion Test.

of the design. This section describes a procedure for the programming of a control algorithm with 8 bit word length limitation.

The error caused by finite word length and their influence on the behavior of the digital controller is reported in Ref. 2. The major effort of the digital controller designer is concentrated on achieving appropriate numerical implementation of the algorithm. Using an 8 bit or 16 bit computer the designer enters the fields of finite word length arithmetic and sampled signals. The programming for a control algorithm is one of real time application. Thus, the fixed point arithmetic should be used in

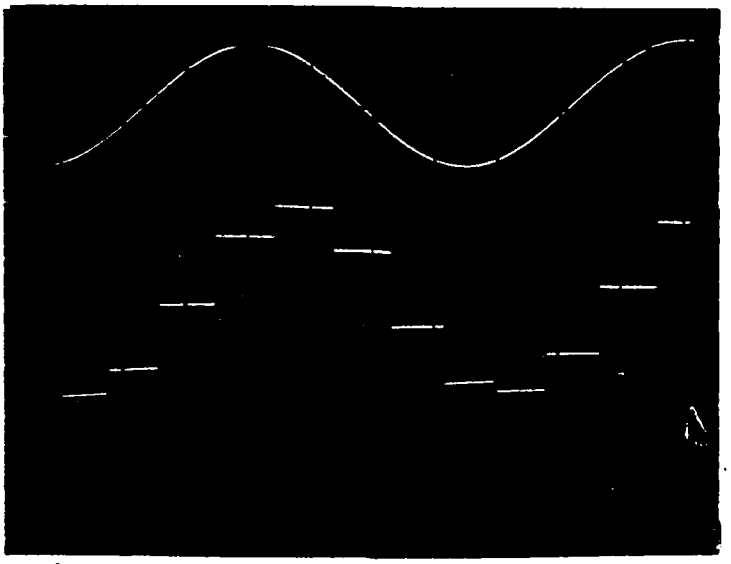


Figure 4.7 Analog vs. Digital(Delay Factor:01H).

programming in order to reduce the program execution time. This section describes the procedure of how to design the two state feedback gain control algorithm using single-input single-output microcomputer controller. This program is attached as Appendix C.

The state equation of Figure 2.1 is

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ -g_1 & -g_2 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

where x(t) is the state vector of the plant.
Although $x_1(t)$ or $x_1(k)$ (where t or k means present time for continuous and discrete system respectively) can be meas-

TABLE III

Sampling Time Measurement

Delay Factor	Decimal	Sampling Period
0001H	1	0.00045 sec
000FH	15	0.0006 sec
007FH	127	0.0014 sec
00FFH	256	0.002 sec
01FFH	511	0.0044 sec
02FFH	767	0.006 sec
0300H	768	0.0061 sec
030FH	783	0.0062 sec
03FFH	1023	0.008 sec

ured, $x_2(k)$ (= $\dot{x}_1(k)$) cannot be obtained because the time derivative of $x_1(k)$ is not realizable in the practical experiment because of only a single-input port. Therefore, $x_2(k)$ is approximated by

$$x_2(k) = \dot{x}_1(k) = \frac{x(k) - x(k-1)}{T} \qquad \text{(eqn 4.1)}$$

Thus, the estimation of $x_2(k)$ can be obtaind by using the above approximation. Figure 4.8 shows how the block diagrams are developed for desiging of the control algorithm. Here, one of the algorithm design idea is that the derivative gain (Kd) is always set to the sampling period. Then the denominator of estimation block can be deleted, This approach means that the computation of estimation block

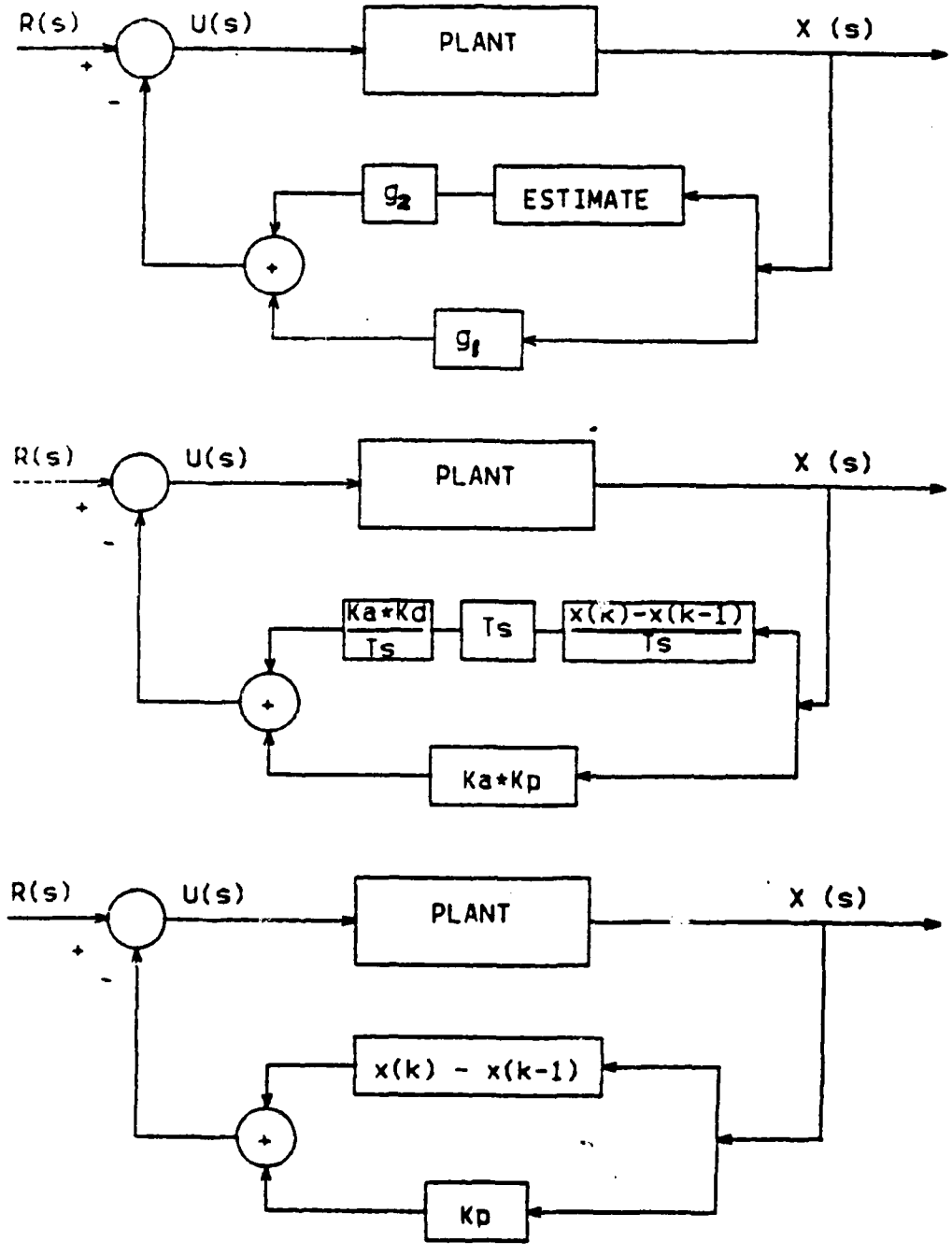
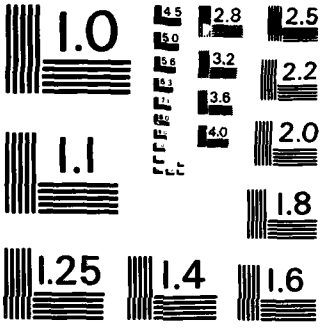


Figure 4.8 Control Algorithm Block Diagrams.

AD-A142 101 A DESIGN METHOD FOR A STATE FEEDBACK MICROCOMPUTER CONTROLLER OF A WIDE BANDWIDTH ANALOG PLANT(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA K C KIM DEC 83 2/2

UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

could be decreased which is an important factor when fixed point assembly language programming is required.

The attenuation factor as a part of the correction factor uses a potentiometer outside of the microcomputer. If the attenuation factor is manipulated within the control algorithm, then floating point arithmetic would be required. Placing the attenuator outside the microcomputer then allows fixed point arithmetic to be used. The flow chart of control algorithm is shown in Figure 4.9 .

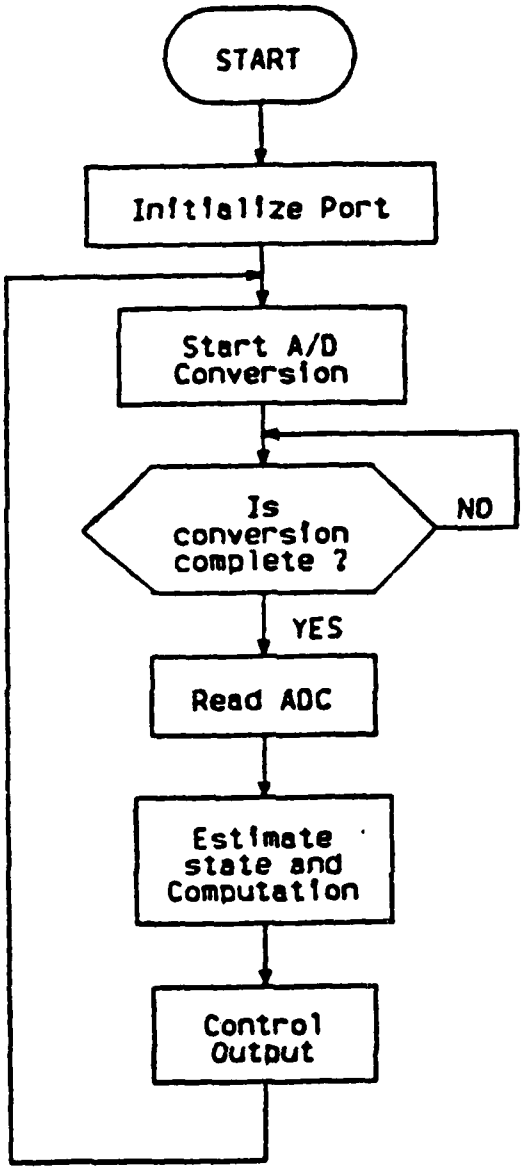


Figure 4.9 Flow Chart of Control Algorithm.

V. CONCLUSIONS AND RECOMMENDATIONS

Results obtained both experimentally and by simulation have shown that a continuous system design method of using state feedback can be applied to a high performance microcomputer based regulator system. The regulator can be designed to meet specifications approaching the level available from a direct analog feedback controller.

Although a low order system was used as a convenience in hardware implementation, digital computer simulation did confirm that the method is general enough to apply to higher order systems. The method applied to higher order systems would probably provide more flexibility in the controller design if $k \leq (n-2)$ state gains were used rather than $k \leq (n-1)$. The additional state feedback gain would provide more lead phase shift to offset the transport lag phase shift due to the microcomputer.

It is well known that assembly language programming as used in this work provides fact implementation of the control algorithm. In those cases where additional time is available, use of higher level programming such as the On-Chip Tiny Basic [Ref. 11,12]. would provide convenience in developing the controller design and is recommended as an area for future study. National Semiconductor's INS 8073 Tiny Basic microinterpreter is one example of a single IC chip that can be programmed in a high level language. Use of Tiny Basic would allow application of assembly language programs where speed is required such as with input/output and multiplication operations. Basic would then be used to develop the estimation or optimal control law or filter design at a language level more familier to design engineers.

APPENDIX A
EXAMPLE PROGRAM OF CSMP

```
INITIAL
INCON       XO = 3.0
PARAMETER   KP = 1.0,  KA = 1.0
PARAMETER   DEL = (.00045,.002,.005,.008)
DYNAMIC
            KD = DEL
            U = 0.0 * STEP(0.0)
            F = U - FDA
            V = REALPL(0.0,0.004,F)
            V0 = 250. * V
            XP = INTGRL(XO,V0)
            V1 = DERIV(0.0,XP)
            F1 = KD * V1
            F2 = KP * XP
            F3 = F1 + F2
            FD = DELAY(10,DEL,F3)
            FDA = KA * FD
TIMER     FINTIM=.05,DELT=.0004,PRDEL=0.0004,OUTDEL=0.0004
PRINT     XP
       TITLE STATE FEEDBACK MICROCOMPUTER CONTROLLER
OUTPUT    XP
       LABEL  STATE FEEDBACK MICROCOMPUTER CONTROLLER
       PAGE MERGE
END
STOP
```

APPENDIX B

ASSEMBLER PROGRAM FOR STATIC AND DYNAMIC TEST

```
;*****************************************************
;* STATIC AND DYNAMIC CONVERSION TEST PROGRAM
;* PROGRAM BY  KIM,  KI  CHUL
;* DATE : 9/OCT/1983
;*****************************************************
;
UPDDT   EQU     036EH   ; Monitor update data subroutine
DELAY   EQU     05F1H   ; Monitor delay subroutine
        ORG     8300H   ; Starting position
;
        LXI     SP,20C8H; Initialize stack pointer
        MVI     A,06H   ; Initialize PIO
                        ; Port A -> input port
                        ; Port B -> output port
                        ; Port c -> ALT3
        OUT     20H     ; Command port
START:  MVI     A,38H   ; Port C bit 5 to '1'
        OUT     23H     ;
        MVI     A,18H   ; Port C bit 5 to '0'
        OUT     23H     ; Start conversion
CHECK:  IN      23H     ; Conversion complete?
        ANI     04H     ; Check Port C bit 2 = '0'
        JNZ     CHECK   ; Return & wait
        IN      21H     ; Read from A/D
        OUT     22H     ; Out to D/A
        CALL    UPDDT   ; Display
        LXI     D,00FFH ; Set delay factor
        CALL    DELAY   ; Delay
        JMP     START   ; Next scmpling
        END
```

APPENDIX C

ASSEMBLER PROGRAM OF STATE FEEDBACK CONTROLLER

```
;*********************************************************
;* PROGRAM FOR STATE FEEDBACK MICROCOMPUTER CONTROLLER
;* PROGRAM BY KIM,  KI  CHUL
;* DATE : 10/NOV/1983
;*********************************************************
;
;*********************************************************
;* This program is written for the microcomputer
;* controller which is state feedback control.
;* The function of this controller is work such as
;* PD controller in the second order system.
;*
;* STATE 1 : the input of controller
;* STATE 2 : the estimation by the controller
;*********************************************************
;
UPDDT   EQU     036EH   ; update display subroutine
SMULT   EQU     1280H   ; fixed point multiply
DELAY   ERU     05F1H   ; delay subroutine
        ORG     8100H   ; set up in non-volatile RAM space
;
XOLD:   DB      0       ; Memory location for old state
                        ; value
EST:    DB      0       ; Memory location for estimation
;
        LXI     SP,20C8H; load stack pointer
        MVI     A,06H   ; load command
                        ; Port A -> input port
                        ; Port B -> output port
                        ; Port C -> ALT3
```

```
        OUT     20H     ; Command status register in
                        ; 8155 RAM
START:  MVI     A,38H   ; enable analog input & output
        OUT     23H     ; port C
        MVI     A,18H   ; start A/D conversion
        OUT     23H     ;
CHECK:  IN      23H     ; Is conversion complete?
        ANI     04H     ;
        JNZ     CHECK   ;
        IN      21H     ; load sampling data into ACC.
        PUSH    PSW     ; save on stack
        PUSH    H       ;
        CALL    UPDDT   ; display value in data field
        POP     H       ; restore value from stack
        POP     PSW     ;
        MOV     B,A     ; save in B
        LDA     XOLD    ; load ACC. with previous sampling
                        ; value
        MOV     C,A     ;
        MOV     A,B     ;
        SUB     C       ; New value - old value
        STA     EST     ; store in memory with difference
        MOV     A,B     ; save in ACC. with new value
        ANI     80H     ; check the negative number
        JNZ     NEGA    ; If negative jump to NEGA:
        MOV     H,B     ; multiplicand(new value)
        MVI     E,01H   ; changable proportional gain
        CALL    SMULT   ;
        LDA     EST     ; save in ACC. with estimation
        ADD     L       ; proportional + derivative
                        ; truncate H reg.
        OUT     22H     ; output the control to D/A
        MOV     A,B     ;
        STA     XOLD    ; store sampling data in memory
        LXI     D, 00FFH; set delay factor
```

```
        CALL    DELAY   ;
        JMP     START   ; repeat the process
NEGA:   MOV     A,B     ;
        CMA             ;
        INR     A       ;
        MOV     H,B     ; multiplicand(new value)
        MVI     E,01H   ; changable proportional gain
        CALL    SMULT   ;
        LDA     EST     ; save in ACC. with estimation
        SUB     L       ; proportional + derivative
                        ; truncate H reg.
        OUT     22H     ; output the control to D/A
        MOV     A,B     ;
        STA     XOLD    ; store sampling data in memory
        LXI     D, 00FFH; set delay factor
        CALL    DELAY   ;
        JMP     START   ; repeat the process
        END


;       *** SUBROUTIONs AND FUNCTIONS ***

;*****************************************************
;* FUNCTION : UPDDT - update data field of display
;* INPUT : B - DOT FLAG -
                1 means put dot at right edge of field
;*              0 means not dot
;* OUTPUT : none
;* CALLS : HXDSP, OUTPT
;* DESTROYS : A,B,C,D,E,H,L,F/F's
;* DESCRIPTION :
;*      UPDDT updates the data field of the display
        using the current data byte.
;*****************************************************
;
```

```
DTFLD   EQU     1       ; indicate use of data field of
                        ; display
        ORG     0       ;
UPDDT:  LDA     CURDT   ; get current data
        MOV     D,A     ; put current data in D
        CALL    HXDSP   ; expand current data for display
                        ; address of expanded data in
                        ; H & L
        MVI     A,DTFLD ; use data field of display
                        ; dot flag is in B
        CALL    OUTPT   ; output current data to data field
        RET
```

```
;*********************************************************
;* FUNCTION : HXDSP - expand hex digits for display
;* INPUT : DE - 4 hex digits
;* OUTPUT : HL - address of output bufer
;* CALLS : nothong
;* DESTROYS : A,H,L,F/F's
;* DESCRIPTION :
;*      HXDSP expands each inpput byte to 2 bytes in
;*      a form suitable for display by the output
;*      routines.
;*      Each input byte id divided into 2 hex digits.
;*      Each hex digit is placed in the low order 4
;*      bits of a byte whose high order 4 bits are
;*      set to zero.
;*      The resulting  byte is stored in the output
;*      buffer. The function returns the address of
;*      the output buffer.
;*********************************************************
;
```

```
        ORG     026CH   ;
HXDSP:  MOV     A,D     ; get first data byte
        RRC             ; convert 4 high order bits
        RRC             ;
        RRC             ;
        RRC             ;
        ANI     0FH     ;
        LXI     H,OBUFF ; get address of otuput buffer
        MOV     M,A     ; store character in output buffer
        MOV     A,D     ; get first data byte and convert
                        ; 4 low order bits to a single
                        ; character
        ANI     0FH     ;
        INX     H       ; next buffer position
        MOV     M,A     ; store character in buffer
        MOV     A,E     ; get second data byte and convert
                        ; 4 high order bits to a single
        RRC             ;
        RRC             ;
        RRC             ;
        RRC             ;
        ANI     0FH     ;
        INX     H       ; next buffer position
        MOV     M,A     ; store character in buffer
        MOV     A,E     ; get second data byte and convert
                        ; 4 high order bits to a single
                        ; character
        ANI     0FH     ;
        INX     H       ; next buffer position
        MOV     M,A     ; store character in buffer
        LXI     H,OBUFF ; return address
        RET
```

```
;*****************************************************
;* FUNCTION : OUTPT - output characters to display
;* INPUT : A - DISPLAY FLAG - 0 = use address field
;*                          - 1 = use data field
;*         B - DOT FLAG - 1 = output dot at right edge of
;*                            field
;*                      - 0 = no dot
;* CALLS : nothong
;* DESTROYS : A,B,C,D,H,L,F/F's
;* DESCRIPTION :
;*      OUTPT sends characters to the display.
;*      The address of the characters is received
;*      as an argument. Either 2 characters are sent
;*      to the address field, depending on the display
;*      Eflag argument. The dot flag argument
;*      determines whether or not the last output
;*      character.
;*****************************************************
;
ADISP   EQU     0       ; indicate use of address field of
                        ; display
DDISP   EQU     94H     ; control character to indicate
                        ; output to data field of display
CNTRL   EQU     1900H   ; address for sending control
                        ; characters to display chip
DTMSK   EQU     08H     ; mask for turning on dot in display
DSPLT   EQU     1800H   ; address for sending characters
                        ; to display
        ORG     02B7H   ;
OUTPT:  RRC             ;
        JC      OUT05   ;
        MVI     C,4     ;
        MVI     A,ADISP ; control character
        JMP     OUT10   ;
OUT05:
```

```
        MVI     C,2     ;
        MVI     A,DDISP ; control character for data field
OUT10:
        STA     CNTRL   ;
OUT15:
        MOV     A,M     ; get out put character
        XCHG            ; save output character
        LXI     H,DSPTB ; get display format table address
        ADD     L       ;
        MOV     L,A     ;
        MOV     A,M     ;
        MOV     H,C     ;
        DCR     H       ;
        JNZ     OUT20   ;
        DCR     B       ;
        JNZ     OUT20   ;
        ORI     DTMSK   ;
OUT20:
        CMA             ;
        STA     DSPLY   ;
        XCHG            ;
        INX     H       ;
        DCR     C       ;
        JNZ     OUT15   ;
        RET


;*****************************************************
;* SUBROUTINE : SMULT
;* DESCRIPTION :
;*      The 8 bit multiplicand is entered in reg H.
;*      The 8 bit multiplier is entered in reg E.
;*      The 16 bit product is returned in reg H-L.nt
;*      Saves B,C,E.
;*****************************************************
```

```
;
        ORG     1280H   ;
SMULT:
        MVI     A,08H   ; loop counter
        MVI     L,00    ;
        MOV     D,L     ;
SMULT1:
        DAD     H       ;
        JNC     SMULT2  ;
        DAD     D       ;
SMULT2:
        DCR     A       ;
        JNZ     SMULT1  ;
        RET


;*****************************************************
;* FUNCTION : DELAY
;* INPUT : DE - 16 bit integer donating number of
;*              times to loop
;* OUTPUT : none
;* CALLS : nothong
;* DESTROYS : A,D,E,F/F's
;* DESCRIPTION :
;*      DELAY does not return to caller until
;*      input argument is counted down to zero.
;*****************************************************
;
        ORG     05F1H   ;
DELAY:
        DCX     D       ; decrement input argument
        MOV     A,D     ;
        ORI     E       ;
        JNZ     DELAY   ;
        RET
```

LIST OF REFERENCES

1. Kuo, C. Benjamin, Automatic Control Systems, Fourth Edition, Prentice-Hall, 1982.

2. Katz, Paul, Digital Control using Microprocessors, Prentice-Hall, 1981.

3. Kuo, C. Benjamin, Digital Control Systems, Holt, Rinehart and Winston, Inc, 1980.

4. Johnsrud, F. Roger, Interactive Microcomputer Control System Modeling and Realization using a Database, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, 1982.

5. TR-20 Computer operator's Reference Handbook, Publ., No. 0800 2003 01, Electronic Associates Inc., May 1964.

6. Microprocessor Application Trainer MAT385, vol. 1, Feedback Instruments Limited, 1974.

7. MAT385 Interfacing & Applications Programming Techniques, vol. 2, Feedback Instruments Limited, 1974.

8. MAT385 Installation & Maintenance, vol. 3, Feedback Instruments Limited, 1974.

9. MAT387 Service Subroutines ROM for use with MAT385, Feedback Instruments Limited, 1974.

10. Clark, N. Robert, Instruction to Automatic Control Systems, John Wiley and Sons, 1962.

11. "On-Chip Tiny Basic Dumps Development Systems," Electronic Design, vol. 28, pp. 235-240, 22 November 1980.

12. Handy, Jim, "An Introduction to NSC Tiny Basic, The Language of the INS8073," Byte, vol. 7, pp. 472-481, April 1982.

13. Speckhart, H. Frank, and Green, L. Walter, A Guide to using CSMP-The Continuous System Modeling Program, Prentice-Hall, 1976.

BIBLIOGRAPHY

Auslander, M. David, and Sagues, Paul, Microprocessors for Measurement and Control, Osbone/McGraw-Hill, 1981.

Jury, I. Eliahu, Sampled-Data Control Systems, John Wiley & Sons, Inc., 1958.

Ledgewood, K. Byron, Control Engineering Manual, McGraw-Hill, 1957.

Monroe, J. Alfred, Digital Processes for Sampled Data Systems, John Wiley & Sons, Inc., 1962.

Moroney, Paul; Willsky, S. Alan, and Houpt, K. Paul, The Digital Implementation of Control Compensators: The Coefficient Wordlength Issue, IEEE Automatic Control, Vol. AC-25, No. 4, pp. 621-630, August 1980.

Newton, C. George Jr.; Gould, A. Leonard, and Kaiser, F. James, Analytical Design of Linear Feedback Controls, John Wiley & Sons, Inc., 1957.

Rembold, Ulrich, Seth, K. Mahesh, and Weinstein, S. Jeremy, Computers in Manufacturing, Marcel Dekker, Inc., 1977.

Shinskey, G. F., Process Control Systems, McGraw-Hill, 1967.

Tsuchiya, Takeshi, Improved Direct Digital Control Algorithm for Microprocessor Implementation, IEEE Automatic Control, Vol. 27, No. 2, pp. 295-306, April 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Professor Alex Gerba, Jr., Code 62Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	2
5.	Professor R. Panholzer, Code 62Pz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	2
6.	LCDR. Kim, Ki chul 238-10 Bul Kwang-dong Eun pyung-Ku Seoul, Korea zip code 120	5
7.	Chief of Electronical Engineering Naval Academy Jin Hae, Korea	1
8.	Professor Park, Jung Ki Department of Electronical Engineering Korea University Seoul, Korea	1
9.	Professor Cha, Kyun Hyen Department of Electronical Engineering Korea University Seoul, Korea	1
10.	Professor Kim, Duck Jin Department of Electronical Engineering Korea University Seoul, Korea	1
11.	Professor Lee, Tae Won Department of Electronical Engineering Korea University Seoul, Korea	1

END

FILMED