1965

Control of a real system by a time-shared computer

Browne, Edward R.; Smith, John S.

CONTROL OF A REAL SYSTEM BY A TIME-SHARED COMPUTER

EDWARD R. BROWNE
and
JOHN S. SMITH

CONTROL OF A REAL SYSTEM BY A TIME-SHARED COMPUTER

* * * * *

Edward R. Browne

and

John S. Smith

CONTROL OF A REAL SYSTEM BY A TIME-SHARED COMPUTER

by

Edward R. Browne

Captain, United States Marine Corps

and

John S. Smith

Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS

United States Naval Postgraduate School
Monterey, California

1 9 6 5

CONTROL OF A REAL SYSTEM BY A TIME-SHARED COMPUTER

by

Edward R. Browne

and

John S. Smith

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ENGINEERING ELECTRONICS

from the

United States Naval Postgraduate School

ABSTRACT

Advances in digital computer technology have led to employment of the computer as a control device. Although many theories for utilization of the digital computer as the compensator in a position feedback control system have been proposed, there is little documentation of digital control performance in a hardware environment. This thesis is an investigation of the performance of a real system under sampled-data control. The variable gain amplifier sampled-data control theory is first simulated and then tested on a real system. The hardware and software developments necessary for implementing this control theory are discussed in detail. Additionally, this method is incorporated in a time-sharing computer program for controlling many systems simultaneously from a single computer. Results of the tests are presented and evaluated.

The authors wish to express their appreciation to Professor Robert D. Strum of the U. S. Naval Postgraduate School for his guidance and assistance during this investigation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I

1.1 Introduction

During the past 15 years, advances in digital computer technology have developed the digital computer into an invaluable tool for all branches of the scientific world. In almost every scientific discipline, the capabilities of the digital computer for vast computational tasks, data processing, and simulation have been profitably employed. In addition to these uses, some engineers in the automatic control field have considered the digital computer, operating in a real time mode, as a tool for improving control techniques and capabilities. It is in this area of computer application that the authors are extremely interested.

There are many theoretical expositions on sampled data control techniques, and some include results from the digital control of simulated plants. However, with the exception of the technical reports of the Navy Electronics Laboratory [1, 2,] the authors were unable to find any information on sampled-data control of an actual plant. It is the opinion of the authors that unless the theory is tested in a hardware environment, the results are academic because of the difficulty in accurately simulating the inherent non-linearities in a real system. It was decided at the outset, therefore, that a prime goal of the thesis was to test the theories of sampled-data control on a real plant.

A second area in which data was lacking involved one of the primary advantages claimed for the sampled-data control techniques: the capability to control many systems simultaneously from the same computer. Although implementing such a multi-control system is primarily a computer

programming task, certain basic techniques must be developed to insure flexibility and adequate performance. It was felt that the lack of documentation in this area was a serious shortcoming. Thus, a second goal of this thesis was to develop and implement techniques for multi-system sampled-data control.

Having established the basic goals of the thesis, the authors made a decision which essentially limited the scope of the thesis. Throughout the literature, there are many proposed theories for sampled-data control systems. It was decided to select only one of these theories as the basis for the laboratory experimentation. Rather than testing many theories in a less rigorous fashion, the authors preferred to make exhaustive hardware tests on the basis of one theoretical solution to the control problem. In general, the difficulties encountered in one theoretical approach would be comparable to those of another, and the techniques developed may be extended to other theories.

After examining many of the proposed theories, the authors, in conjunction with their advisor, Professor Robert Strum, selected the theory proposed by Benjamin C. Kuo. The basis of this theory is the concept of a variable gain amplifier in the control loop. The theory provides for minimum time, ripple free response to deterministic inputs. The prime reasons for selection of this theory as a basis for the laboratory experimentation are the theoretical advantages it offers, and the relatively simple tasks the digital computer is required to perform.

In considering the laboratory portion of the thesis, the authors

had one more basic decision to make. In selecting a digital computer for the experiments, the facilities at the Postgraduate School offered a choice between a general purpose computer, the Control Data Corporation 1604, and a small data processing computer with an auxiliary arithmetic capability, the CDC-160 in conjunction with the CDC-168 arithmetic unit. Due to the fact that any practical control method must not only perform well, but also must be economically feasible, the smaller, less expensive 160 computer was selected. It is conceded that the larger computer could more efficiently perform the required controlling tasks, but the authors preferred the more realistic practical approach. This led to a secondary goal of the thesis, to be able to comment on the general feasibility of sampled-data control systems on the basis of the results obtained.

1.2 Digital Controller Design

Before proceeding with work on the real plant, the authors decided to select a technique for the design of digital controllers for sampled-data control systems. A design theory that would give accurate results and would be readily adaptable to the existing hardware was desired. It was determined that the digital controller design described in Section 9-9 of Analysis and Synthesis of Sampled-Data Control Systems by Benjamin C. Kuo would be well suited to this problem. [3] Kuo's design theory makes use of state space and state transition techniques. It also enables derivation of a D(Z) that will yield a minimum time, ripple free response to the designated deterministic input (i.e., "deadbeat response"). This approach to digital controller design describes the digital controller as a variable gain amplifier which outputs certain values each sampling period in order to control the continuous system

in the desired manner. It is easily adaptable to implementation on a digital computer, and the use of state variables makes the manipulation of the mathematics involved relatively easy.

To explain this "variable gain amplifier" design theory it is best to look at a sample problem. Consider the sampled-data control system of Figure 1. The variables x_1 and x_2 are the state variables for the controlled process where $\dot{x}_1 = x_2$; h(t) is the output of the zero-order hold and e(t) is the actuating signal. Note: $e(t) = r(t)-c(t); x_1=c(t)$

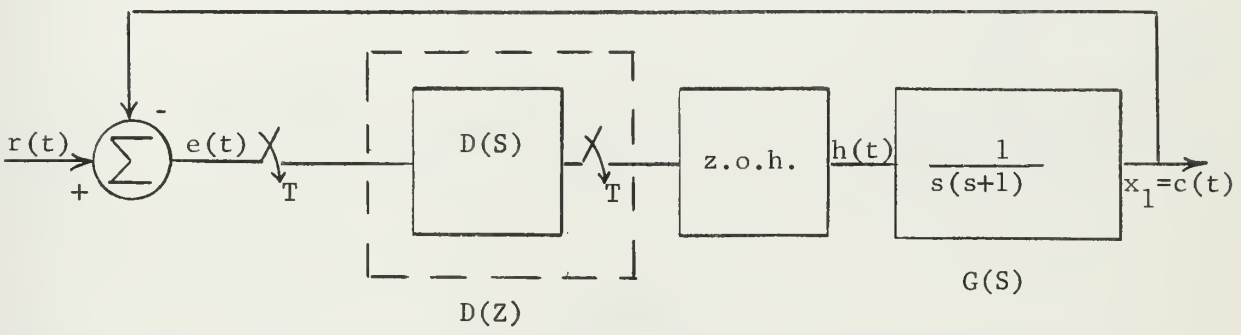


Figure 1

To determine a D(Z) for "deadbeat response" to a given input:

(1) Draw the state transition flow graph of the system with the digital controller represented by a variable gain amplifier K(kT). $K(kT) = h(kT) / e(kT)$.

(2) For a "deadbeat response" the system error must be zero for $t > nT$ where n is the smallest possible integer.

i.e., $x_1(nT) = r(nT)$

$x_2(nT) = x_3(nT) = \ldots = x_p(nT) = 0.$ unit step

$x_2(nT) = 1;\ x_3(nT)=x_4(nT)=\ldots=x_p(nT) = 0.$ unit ramp

From this information the variable gains (K_k's) and the minimum n can be determined.

(3) Compute the $h(0^+)$, $h(T^+)$, ...$h(nT^+)$ using the K_k 's determined and recalling that $e(kT^+)=r(kT)-x_1(kT)$.

(4) Calculate D(Z) using the relationship $D(Z)=H(Z)/E(Z)$.

Returning to the example, let us draw the state transition flow graph of the system.

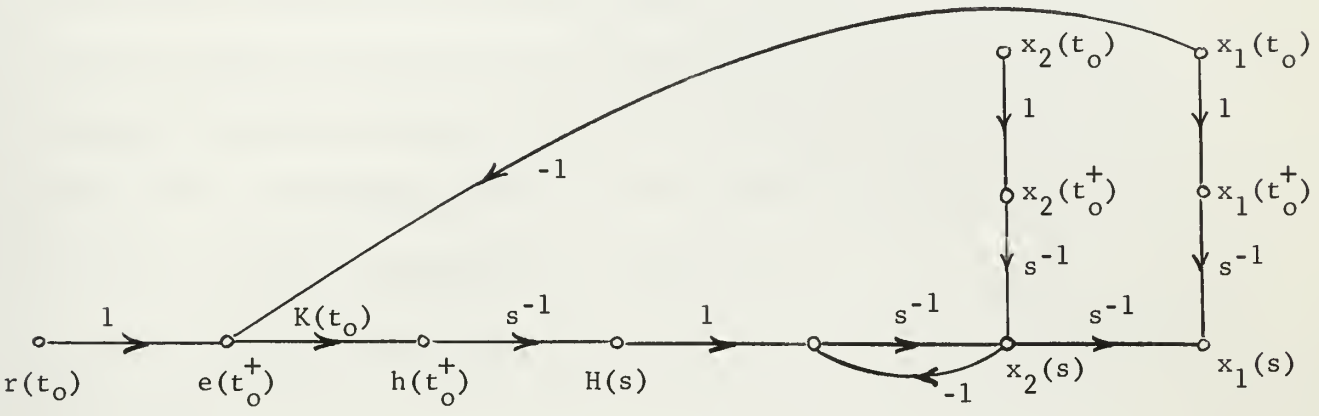


Figure 2

Using Figure 2 and applying flow graph techniques and the inverse Laplace transformation we arrive at the following:

$$x_1\left[(k+1)T\right] = \left[1-K_k(T-1+e^{-T})\right] x_1(kT)+(1-e^{-T})x_2(kT)+K_k(T-1+e^{-T})r(kT)$$

$$x_2\left[(k+1)T\right] = -K_k(1-e^{-T})x_1(kT)+e^{-T}x_2(kT)+K_k(1-e^{-T})r(kT)$$

Now let us assume a unit step input and a sampling period of one tenth of a second. ($r(kT) = 1.0$, $T=0.1$)

$$x_1\left[(k+1)T\right] = (1 - 0.005K_k)x_1(kT) + 0.095\, x_2(kT) + 0.005K_k$$

$$x_2\left[(k+1)T\right] = -0.095K_k x_1(kT)+ 0.905x_2(kT) + 0.095K_k$$

For k=0 and $x_1(0) = x_2(0) \triangleq 0$.

$$x_1(T) = 0.005K_o$$

$$x_2(T) = 0.095K_o$$

From the constraints placed on the problem (i.e., $x_1(nT) = r(nT)$ and $x_2(nT)=0.$) it may be seen that the solution can not be obtained from the above. In practice this means that the output of the controlled process cannot be made equal to a step input in one sampling period. We must therefore proceed to the second sampling period and let k=1. Doing this and setting $x_1(2T) \triangleq 1.0$ and $x_2(2T) \triangleq 0.$ yields:

$$x_1(2T) = 0.014K_o - 0.000025K_oK_1 + 0.005K_1 \triangleq 1.0$$

$$x_2(2T) = 0.086K_o - 0.000475K_oK_1 + 0.095K_1 \triangleq 0.0$$

from which

$$K_o = 105.6 \; ; \; K_1 = -203.0$$

therefore

$$e(0^+) = r(0)-x_1(0)=1 \quad ; \quad e(T^+)=r(T)-x_1(T) = 0.472$$

$$h(0^+) = 105.6 \quad ; \quad h(T^+) = K_1e(T^+) = -95.8$$

and

$$D(Z) = \frac{h(0^+) + h(T^+)Z^{-1}}{e(0^+) + e(T^+)Z^{-1}} = \frac{105.6 - 95.8\,Z^{-1}}{1 + 0.472\,Z^{-1}}$$

The above example demonstrated the technique of designing a digital controller for a sampled-data control system using the variable gain amplifier concept. It should be observed that in implementing this controller using a digital computer, the D(Z) does not have to be actually calculated. In fact, only the K_k's need be entered in the computer.

This proves to be quite advantageous since for a given sampling period the K_k's are only a function of the continuous system gain and the G(S). Thus for a digital computer controlling a number of plants the K_k's could be pre-computed and stored in the computer ready for use. These variable gains when multiplied by the sensed error and output at the sampling rate will provide the desired system response to the designated input. In fact, the digital controller designed above will yield the desired "deadbeat response" to any step input. It must also be pointed out that the design given is only for a step input and will not give optimum response for a ramp or parabolic input.

The design problem as stated and solved above was for the control of a type one system in response to a step input. In studying this design technique a digital controller was also designed for the same type one system ($G(S)= \frac{1}{s(s+1)}$) for "deadbeat response" to a unit ramp input. With respect to a ramp input, optimum response was obtained in three sampling periods. It was also noted that once zero error had been achieved a constant output of 1.0 was needed from the controller. This was required because a type one system has an inherent steady state error in response to a ramp input. In this context it is important to realize that with the use of a digital controller a continuous system with an inherent steady state error can be made to respond with "deadbeat response" and no steady state error to any particular deterministic input. (See Appendix I for details.)

In Appendix I a digital controller is also designed for a type zero system ($G(S) = \frac{1}{(s + 1)\ (s + 2)}$) which has an inherent steady state

error to both step and ramp inputs. With the designed controller, these errors were reduced to zero.

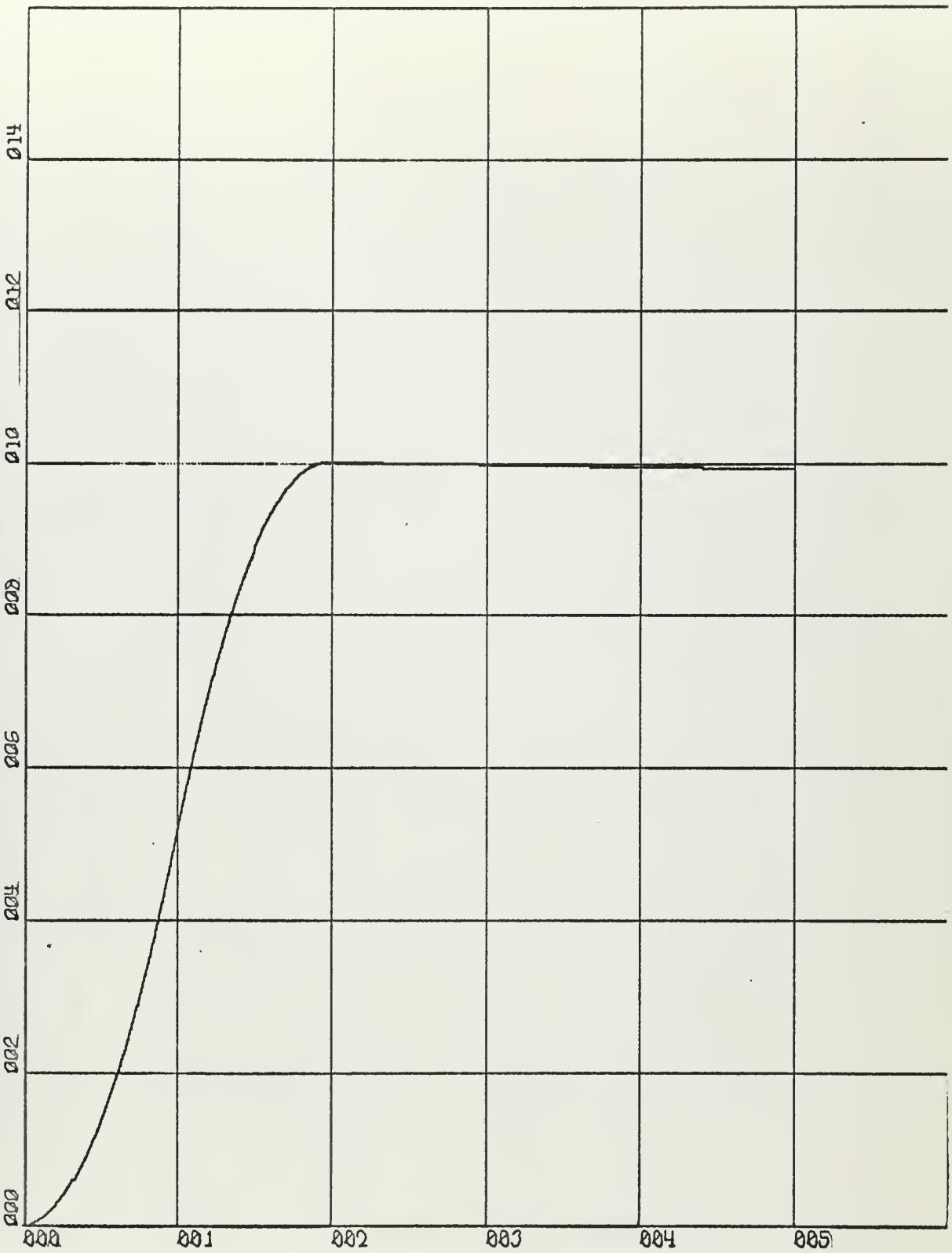
With regard to type two systems some difficulty was encountered. For a system of the type $G(S) = \frac{1}{s^2}$ a digital controller was readily designed; however, for a system of the type $G(S) = \frac{1}{s^2(s+1)}$ the mathematics involved in evaluating the K_k's proved to be cumbersome. An iterative solution to the non-linear simultaneous equations was used in evaluating the K_k's. This proved to be feasible. Further investigation into the problem was not carried out since the authors wished to direct their efforts to the control of the real system.

1.3 Simulation and Results

After completing the design of a digital controller for a "dead-beat response" to a unit step input, it was decided to simulate the system response. This was done to provide a check on the computations and to verify the design theory itself. The simulation programs used were written in Fortran 60 and employ a library routine known as INTEG 1. INTEG 1, written by Dr. J. R. Ward, provides a fourth order Runge-Kutta solution to ordinary differential equations and was well suited to the simulation problem encountered. (Appendix II contains a complete simulation program.)

The graphical results to the simulation are on the following pages and do verify the design theory as well as the computations involved. (Appendix I contains graphical results for a unit ramp input as well as results for a type zero system in response to both unit step and ramp inputs.)

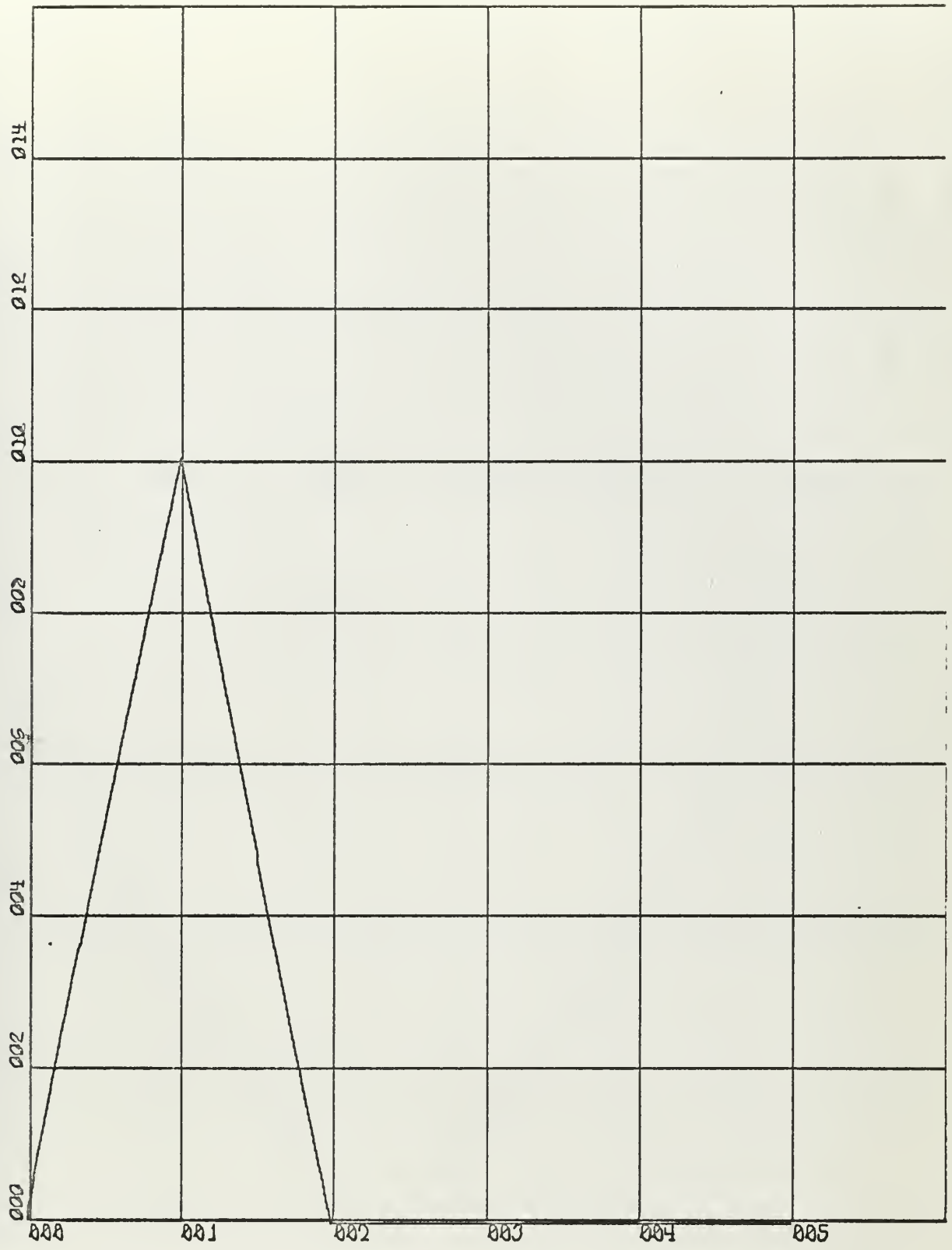
It should be mentioned that a certain degree of accuracy is needed in the determination of the h's used. In some simulations, the rounding of the h value to two decimal places resulted in a response that was slightly less than optimum.



X-SCALE = 1.00E-01 UNITS/INCH.
Y-SCALE = 2.00E-01 UNITS/INCH.

BROWNE DIGIT 1
RUN 1 OUTPUT VS. TIME

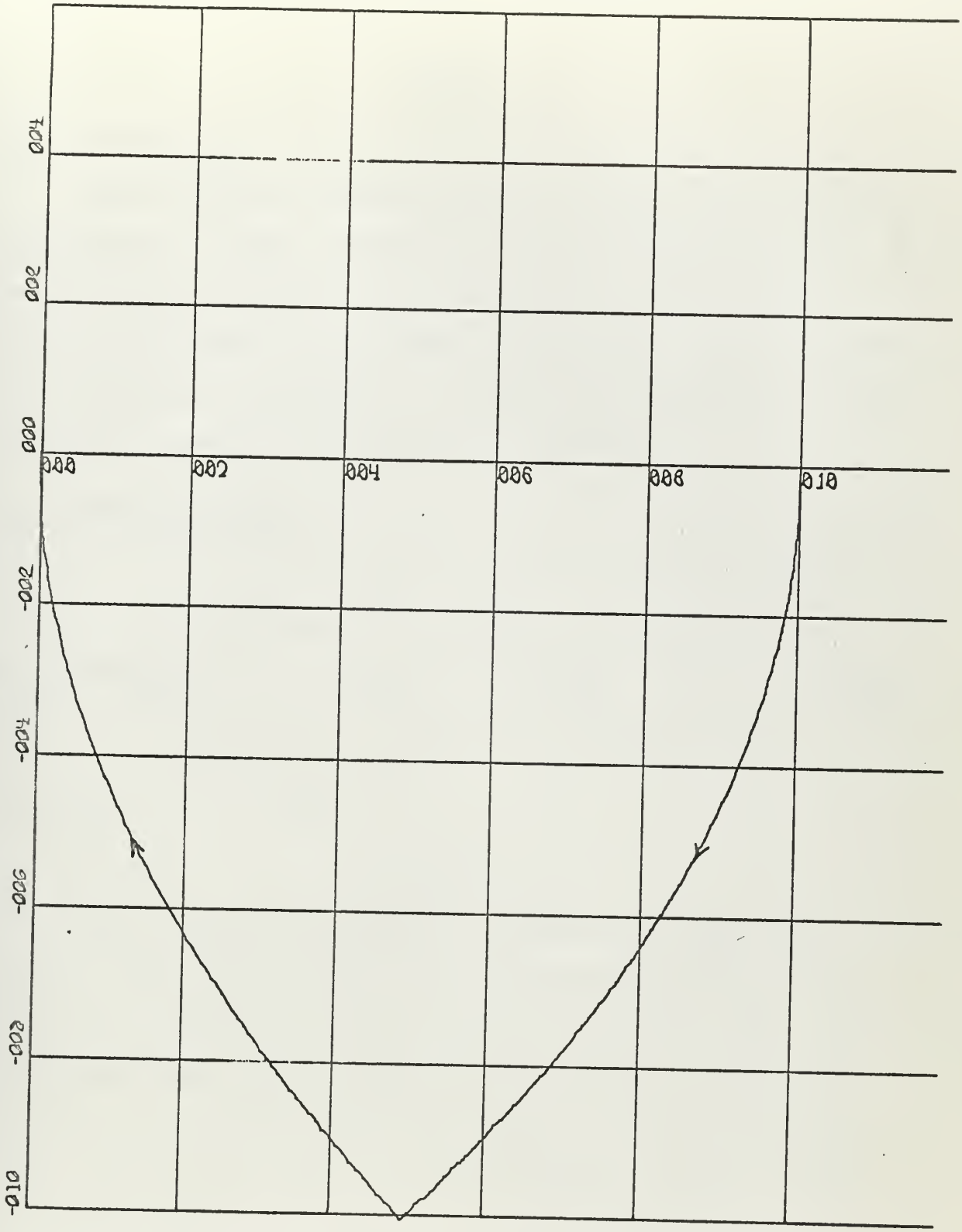
Figure 3



K-SCALE = 1.00E-01 UNITS/INCH.
Y-SCALE = 2.00E+00 UNITS/INCH.

BROWNE DIGIT 1
RUN 1 VELOCITY VS.TIME

Figure 4



X-SCALE = 2.00E-01 UNITS/INCH.
Y-SCALE = 2.00E+00 UNITS/INCH.

BROWNE DIGIT 1
RUN 1

EDOT VS. ERROR

Figure 5

CHAPTER II

2.0 Investigation

The investigation conducted during the course of this thesis was developmental in nature. Using the variable gain amplifier control theory discussed in Section 1.2, techniques for accomplishing the sampled-data control of a real plant were developed. The investigation also included the development of a computer time sharing system through which many systems can be simultaneously controlled.

2.1 Theoretical Design

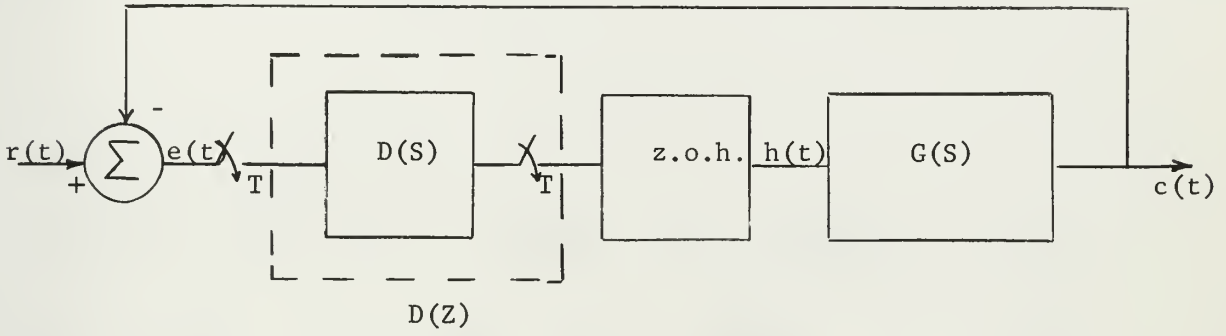


Figure 6

Figure 6 shows the general block diagram of the sampled-data control system that was used. The system breaks down into two main components which are the continuous plant and the D(Z), sampler, and zero-order hold. The continuous plant is described in Appendix III. The CDC-160, CDC-168, and A/D, D/A converters function as the sampler, D(Z), and zero-order hold. This operation will be treated in detail in Section 2.2.1.

The digital controller, D(Z), was designed to give a "deadbeat response" to a step input using the theory described in Section 1.2.

It was decided to solve for the K_k's in general terms initially and then substitute the known G(S) determined in Appendix III.

Figure 7 is shown for a generalized $G(S) = \frac{A}{s(s+\alpha)}$.

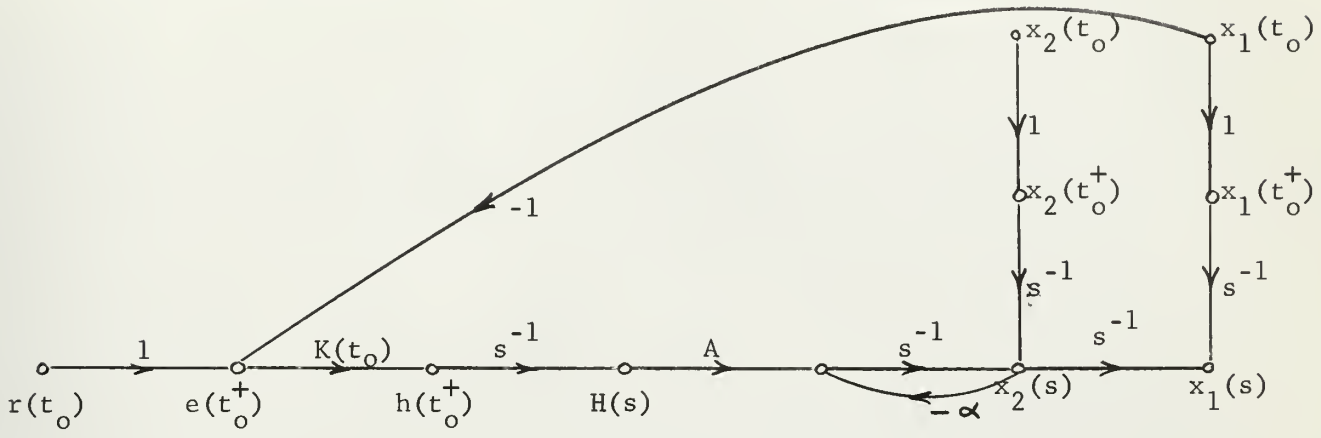


Figure 7

From the flow graph we have

$$x_1(s)=\left[\frac{1}{s} - \frac{KA}{s^2(s+\alpha)}\right]x_1(t_o)+ \frac{1}{s(s+\alpha)}\ x_2(t_o)+ \frac{KA}{s^2(s+\alpha)}\qquad r(t_o)$$

$$x_2(s)= \frac{-KA}{s(s+\alpha)}\ x_1(t_o) + \frac{1}{s+\alpha}\ x_2(t_o) + \frac{KA}{s(s+\alpha)}\qquad r(t_o)$$

and after taking the inverse Laplace transformation we arrive at

$$x_1\left[(k+1)T\right]=\left[1 - \frac{KA}{\alpha^2}\left(\alpha T-1+e^{-\alpha T}\right)\right]x_1(kT)+(1-e^{-\alpha T})x_2(kT)+ \frac{KA}{\alpha^2}\left[\alpha T-1+e^{-\alpha T}\right]r(kT)$$

$$x_2\left[(k+1)T\right]= \frac{-KA}{\alpha}(1-e^{-\alpha T})x_1(kT)+e^{-\alpha T}x_2(kT)+ \frac{KA}{\alpha}(1-e^{-\alpha T})\ r(kT)$$

For $x_1(0) = x_2(0) \triangleq 0.0$

and for a step input of $r(0) = r(T) = (2T)\ldots\ldots\ldots \triangleq r$

we have

$$K_o = \frac{\alpha}{AT(1-e^{-\alpha T})} \qquad K_1 = \frac{\alpha^2 e^{-\alpha T}}{A\left[e^{-\alpha T}(1+\alpha T)-1\right]}$$

$$h_o = \frac{\alpha}{AT(1-e^{-\alpha T})}\ \Delta r \qquad h_1 = \frac{-\alpha e^{-\alpha T}}{AT(1-e^{-\alpha T})}\ \Delta r$$

where $x_1(2T) \triangleq r$: $x_2(2T) \triangleq 0.0$

It is seen that a "delta" term is present in the solution for the K_k's and the h's. This "delta" term arises from the fact that the error signal was attenuated by a 0.0121 factor before being sent to the A/D converter. This will be discussed in detail in Section 2.2.2. Now for the $G(S) = \frac{40}{s(s+3)}$; T = 0.3; where $r \triangleq 5.0$ we arrive at:

$$K_o = 0.421 \qquad K_1 = -0.398$$

$$h_o = 0.0252 \qquad h_1 = -0.0102$$

The h's computed were used in an earlier trial program. The computed K_k's were used in the final control program and as such were entered directly in the computer. The CDC-160 together with the CDC-168 were programmed to serve as the digital controller using these K_k's. The CDC-160 also operated as the sampler since the sampling period was also programmed into the computer.

The real system response was also simulated using the same simulation scheme as Section 1.3. The simulation results are in Section 2.3.

After controlling the real plant in a "deadbeat response" manner, it was decided to work on a time sharing routine for optimum control of two plants. The second plant was an analog simulation set up on the EAI TR-20 analog computer with a G(S) equal to $\frac{1}{s(s+1)}$. The CDC-160 was then used to provide optimum control to both systems. The design of the digital controller for the second plant followed the same theory used before.

2.2 Development of Sampled-Data Control Technique

The step from the theoretical control solution to a working sampled-data controller involves the development of methods through which the requirements of the theoretical solution may be implemented in a real controller. This is a two-fold development involving the software of computer programming and the associated hardware requirements. Development in each area will be discussed in detail.

2.2.1 Hardware Development

Having developed, theoretically, the performance of the D(Z) portion of the control system block diagram, it was necessary to develop a hardware equivalent to the previously described mathematical model of D(Z). To accomplish this hardware development, the following equipment was used:

(1) A two-channel digital to analog converter. (D/A)

(2) A four-channel analog to digital converter. (A/D)

(3) A Control Data Corporation 160 computer, in conjunction with a Control Data Corporation 168 arithmetic unit.

(4) Operational amplifiers used for summing of signals coming in and out of the converter units.

The equipment used was chosen, not because it was thought to be most efficient for the desired tasks, but because it was readily available in the Digital Control Laboratory of the U. S. Naval Postgraduate School. Comments on the relative merits of the equipment in performing these tasks will be included where appropriate.

Development of the hardware equivalent of the mathematical model of D(Z) will now be discussed. It was decided immediately that the need

for analog hold circuits could be eliminated by replacing them with more accurate computer programmed holds. This is possible because of the operating characteristics of the converter units in conjunction with the 160 computer. First, the 160 computer can sample a converted digital input only when specifically commanded, and then it samples only one value. Similarly, the 160 computer can output a digital value for conversion only upon command, and furthermore, that value remains constant in the output register until it is changed by inserting a new value. Thus, by programming the computer to sample converted analog inputs and provide digital outputs for conversion at specified time intervals, hold circuits become unnecessary. In fact, the holds become an inherent part of the sampling rate of the computer and are varied automatically as the sampling rate changes.

Examining Figure 6, one sees that the input to D(Z) is the error signal, e(t), which is the sum of the reference signal, r(t), and the negative control signal, -c(t). There are two methods by which e(t) may be transmitted to the computer. The first method is to sample r(t) on one channel of the A/D and to sample -c(t) on another channel. The two inputs could then be summed in the computer to obtain the error signal. This method was discarded for two reasons. First, since it is impossible to sample r(t) and -c(t) simultaneously, the error signal computed would have some inherent error. Second, the A/D conversion time is of the order of 120 microseconds, and it was felt that this additional time delay might become prohibitive at sampling rates approaching 0.1 seconds.

The second method of supplying the error signal to the computer, and the one which was adopted, involved summing r(t) and -c(t) in an operational amplifier prior to transmission to the A/D. This method has the advantage of being fast, accurate, and readily adaptable to a restriction imposed by the physical characteristics of the converter unit. This restriction is that the converter units accept only negative voltages in the range of zero to minus ten volts. Thus, to handle both positive and negative values of e(t), the sum of r(t) and -c(t) is added by means of an operational amplifier to minus five volts. A zero error signal would then be sampled by the computer as an input of minus five volts, and an error signal of three volts would be sampled by the computer as an input of minus two volts. Similarly, the converted D/A value which is transmitted to the amplifiers of the system must be added to five volts prior to insertion in the system.

A block diagram of the hardware system used to provide the equivalent to the mathematical model of D(Z) is shown below:

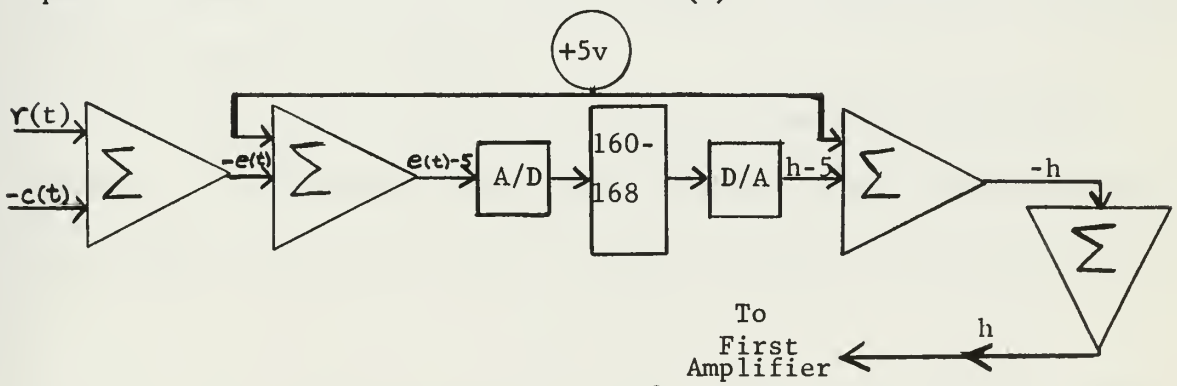


Figure 8

With the above hardware system established, the rest of the development

of a unit equivalent to the mathematical model of D(Z) must be accomplished by programming the computer.

2.2.2 Software Development

Programming the computer to accomplish the aforementioned task was done in a progressive fashion, starting with programs to accomplish the simplest control functions and expanding the basic programs to provide for accomplishing more complex tasks. This was done for two reasons. The primary reason was that at the outset the authors wanted to isolate quickly any faults in the digital control system, and to analyze carefully system response at each stage of control function complexity. A secondary reason for adopting this progressive programming technique was the fact that the 160 computer is rather limited in its capability[1]. Due to this limitation, the authors desired to bring the control capabilities of the equipment used to a maximum before computer saturation problems were encountered. Following is a discussion of these progressive programs.

The first program written was intended primarily to check the operation of the hardware equivalent of the D(Z) under minimum capability requirements, and also to check the plant response with the D(Z) in the

[1]The 160 has limited storage capacity (4K), and a relatively slow memory cycle (6.4 microseconds). It can only multiply or divide in conjunction with the 168 and the 900 microsecond time for such operations is quite slow.

loop. The essence of Program 1 is to enable the plant to run as a continuous system with the D(Z) block in the loop. Mathematically, such a system may be described as D(Z) = 1., and the sampling period, T, approaching zero (approximately 200 microseconds).

The programming aspects of Program 1 were quite simple. The program was an iterative loop in which the current error signal was sampled from the A/D converter, read into the computer, stored temporarily, and output to the D/A converter for return to the system. A flow chart for Program 1 is shown below. A complete text of Program 1 is presented in Appendix VI.

Program 1

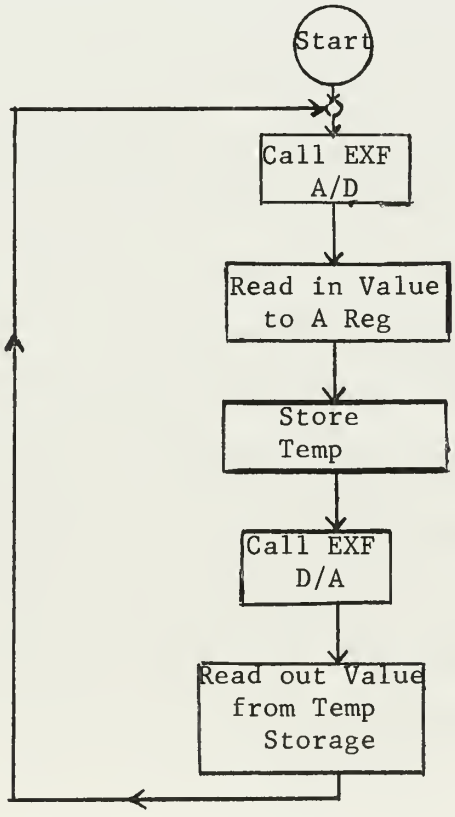


Figure 9

The results of the test using Program 1 were quite satisfactory. System responses were obtained from step, ramp, and sinusoidal inputs, and these responses matched those obtained from the system when operated as a purely continuous system with unity feedback. This, of course, was predictable with D(Z) = 1., and T approaching zero. However, the results do validate the selection of hardware used to implement the realization of the mathematical D(Z).

The results of Program 1 opened another avenue of approach to the problem of implementing the variable gain samplifier method of digital control. It was noted, when working with the system in the Digital Control Laboratory, that the system was subject to many random inputs of small magnitude from various sources of noise. Program 1, when tested, maintained stability, and accurate and fast response in the presence of these noise sources. It was further noted that the advantages of the variable gain amplifier method in regard to minimum time response are lessened when the inputs are small and random.

In the light of the foregoing facts, a temporary control philosophy was developed. This philosophy states that for step inputs which exceed in magnitude a threshold, a standard variable gain amplifier control solution would be employed. For small inputs, such as noise, the system would be controlled as it was with Program 1. The threshold would be set at a level which would prevent the implementation of a solution of the type described in Section 1.2 for the random noise inputs, which were restricted to a relatively low magnitude. In other words, the

system would run in an essentially continuous mode D(Z)=1., until receipt of a valid step input, at which time it would be controlled by a variable gain amplifier digital solution.

To implement this type of control philosophy, an intermediate program, Program 2, was written to modify Program 1 to provide for detection of a definite step input and an exit to a solution. This detection is accomplished by comparing the new value of the sampled error signal with the immediately preceding value of the sampled error signal. If the difference between these values is greater than the established threshold, a step input is detected. To assist the reader in understanding the programming procedure for determining the sign of a detected step input, the following table of octal number values for analog voltages into the converter is presented:

ANALOG VOLTAGE	OCTAL NUMBER
0	4000
-1	4632
-2	5463
-3	6314
-4	7144
-5	0000
-6	0631
-7	1463
-8	2314
-9	3144
-10	3777

A flow chart for Program 2 is presented on the following page. A complete text of Program 2 is presented in Appendix VI.

The results of Program 2 were excellent. It proved to be a fast and accurate method of sensing step inputs having a magnitude greater

Program 2

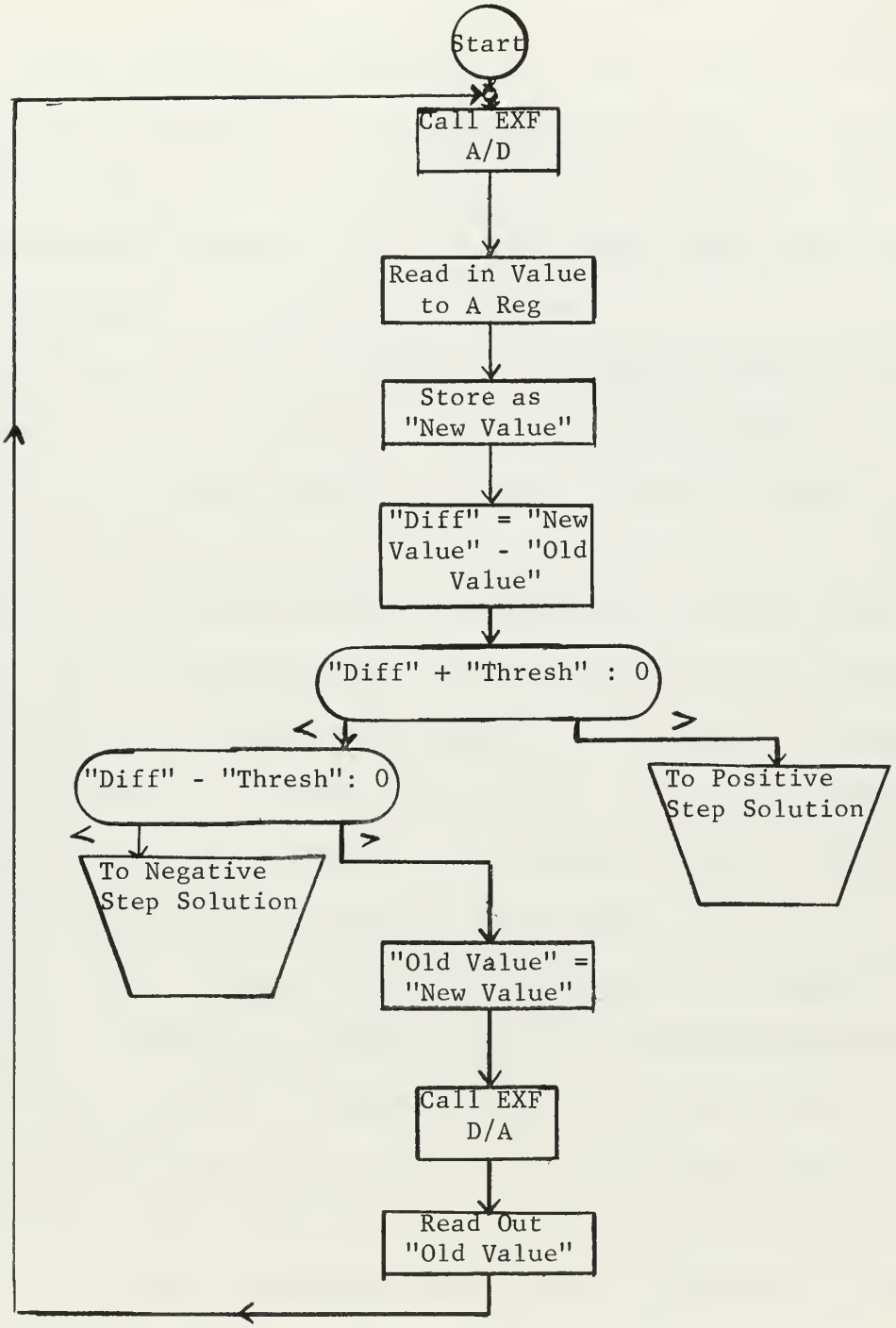


Figure 10

than the threshold level. In one sense Program 2 was too sensitive. When certain knife switches were used to provide zero to five volt step inputs, negative transients appeared at the instant the switch was thrown. Program 2 was so sensitive that it sensed the transient as a negative step input. This imposed the requirement that step inputs be obtained from an electronic switching device or from a high quality knife switch which eliminates the aforementioned transient.

The next step in the development of the temporary control philosophy involved utilizing the computer in a real time mode to implement a variable gain amplifier solution. The iterative loop of Program 2, which maintains continuous control until a step input is sensed, employs the computer in a free-running mode. The variable gain amplifier solution, however, requires that the computer insert control voltages of definite real time length into the system. To do this, the computer must output a control voltage to the system, hold the voltage for a specified length of time, and then output another control voltage. Program 3 was written to develop techniques for accomplishing this task.

Program 3 was a simple program which called for the computer to output a given voltage, delay through a timing chain, and output another voltage. Theoretically, the length of the delay could be calculated from the execution time for the instructions which constitute the timing chain. However, exact timing data for D/A conversion were not available. Therefore, approximate timing chain delay was calculated and adjusted by experimental results for exactness. The timing chain was formed by constructing an iterative loop in which an index was increased by one on each pass and compared to a preset total. When the index equaled the

preset total, the time delay was complete. A flow chart for Program 3 is shown on the following page. A complete text of Program 3 is presented in Appendix VI.

In testing Program 3, it was decided to design a basic delay block of 0.1 second. With an accurate delay block established, delays which were integer multiples of 0.1 second could be achieved by iterating through the basic delay the required number of times. Delays of less than 0.1 second could be achieved by setting the preset total at a proper fraction of that required for a 0.1 second delay.

Satisfactory results were obtained from Program 3 in that a delay of exactly 0.1 second was produced between the output of the first and second voltages. For this delay, the preset total was set at 3403_8. Results were checked on a high speed Mark II Brush Recorder and an oscilloscope.

Having developed Programs 2 and 3, the basic tools for implementing a solution within the constraints of the temporary control philosophy were ready. By using Program 2 to maintain continuous type control in the absence of an input signal and to sense a step input, and by employing the features of Program 3 to output the required controlling voltages, the desired solution to a step input may be attained. To test the basic validity of their approach, the authors wrote Program 4 as a first trial of the variable gain amplifier control method.

Program 4 was an unsophisticated approach which started with the system being controlled in the undisturbed state by Program 2 which will hereafter be referred to as the Sense Loop. When an input is sensed, h_o

Program 3

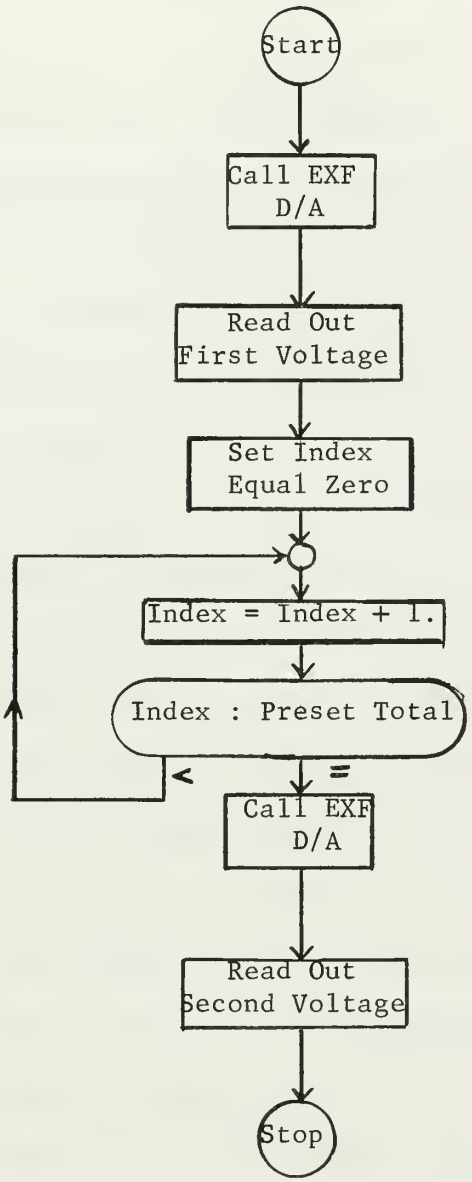


Figure 11

is transmitted immediately to the system. h_o is held for one sampling period, and then h_1 is transmitted and held for one sampling period. At the completion of the second sampling period, the system returns to

continuous operation as the computer returns to the Sense Loop. To insure a return to the continuous mode of operation at the completion of the second sampling period, it was necessary to "bootstrap" the Sense Loop by sensing the current error and inserting it in the Sense Loop as the "old value". Were this procedure not followed, an imperfect solution, one which did not have zero error at the end of two periods, would cause the Sense Loop to sense another step input and thus cause the system to go into unstable, erratic operation.

The unsophisticated aspect of Program 4 stems from the fact that the h's were not computed by multiplying e(t) by K during the solution, but were pre-calculated and inserted in memory as constants to be transmitted to the system at the proper time. K_0 and K_1 have been determined in Section 2.1. h_o was simply K_0 times $e_0(t)$, and h_1 was K_1 times $e_1(t)$ (predicted). Due to the fact that r(t) and -c(t) were transmitted to the summing operational amplifier after passing through dropping resistors in the front end of the system, the magnitude of the error signal was considerably decreased. For a five volt step input, $e_0(t)$ sensed by the computer was 0.0600 volts and predicted $e_1(t)$ was 0.0258 volts. On the basis of these values, h_0 = 0.0252 volts and h_1 = -0.0102 volts. A 0.3 second sampling interval was used.

A flow chart for Program 4 is shown on the following page. A complete text of Program 4 is presented in Appendix VI.

Complete results of the responses obtained when using Program 4 will not be presented because these tests were in the nature of feasibility checks. The results may be summarized by stating that they provided

Program 4

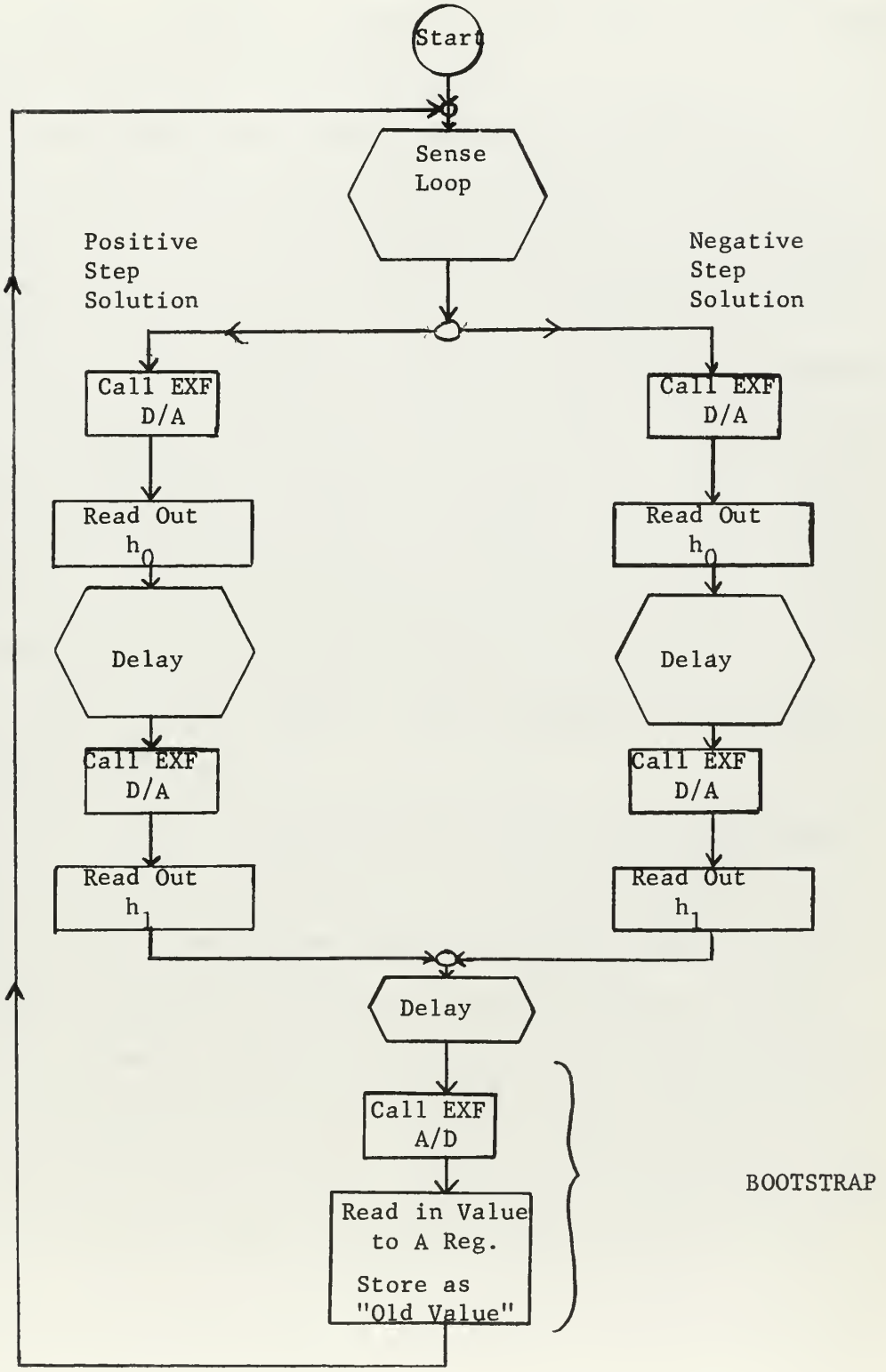


Figure 12

definite proof of the validity of the variable gain amplifier approach, and led the authors to adopt certain techniques which were used in later programs to improve performance. Some of these lessons learned from these tests will now be discussed.

Accuracy problems were caused by the small magnitude of the error signal received at the D(Z) portion of the system for a five volt step input. Due to this fact, the magnitude of h's to be transmitted to the first amplifier was so small that the resolution of the D/A converter prevented sufficient accuracy for these h's. The resolution for the converter is 0.0024 volts per octal number. With h's of the magnitude of 0.02 volts, accuracy was limited to the first digit of the h. This condition was barely satisfactory due to the stringent demands of the variable gain amplifier method for accuracy in the h's. This demand was demonstrated when solutions were simulated on a general purpose digital computer. This demand was discussed more thoroughly in Section 1.3.

To correct the problems associated with converter resolution, the obvious solution was to increase the voltage levels in and out of the converter. The simplest method of accomplishing this would have been to eliminate the dropping resistors before the summing amplifiers. However, when this was tried, system noise increased to a level which was not tolerable. The method of achieving increased accuracy, which produced the best results, was to multiply the error signal by ten in the summing operational amplifiers prior to A/D

conversion, and to divide the D/A converted voltage by ten in the summing operational amplifiers before transmittal to the system. This enabled the use of h's of ten times computer value for conversion which alleviated the resolution problem. This scheme also maintains operation in the continuous mode of the Sense Loop to continue as before. For all further tests, the hardware shown in Figure 8 was modified to provide for the proper multiplication and division in the summing operational amplifiers.

A second lesson learned in this test program related to the "bootstrap" procedure discussed previously. This feature was originally inserted in the program to prevent system runaway for incorrect solutions resulting from tests using improper h's. It was found, however, that this "bootstrapping" return to continuous operation at the completion of the second period was an integral part of a satisfactory solution. This is due to the fact that a real system has inherent non-linearities such as coulomb friction and backlash. In view of these non-linearities, a perfect theoretical solution can not be obtained. (It is possible to approach more closely the perfect solution by slight modification of the size of the h's.) Thus, a rapid, smooth return to the continuous mode after the second sampling period is essential to a satisfactory solution. It is noted that by properly adjusting the size of the h's, the error at the end of the second sampling period is small (less than 10% of the step size), and the use of the continuous mode reduces error to zero rapidly. Specific examples of the foregoing may be noted in the presentation

of results in Section 2.3. At that time, the reader may note that the necessity to go to the continuous mode to bring error to zero does not significantly degrade system performance in comparison to a theoretical solution.

With the information derived from Programs 1 through 4, the authors developed Program 5 which was intended to culminate efforts in the development of the temporary control philosophy. Program 5 was a generalized version of Program 4, the primary difference being that in the newest program the h's were computed during the solution instead of using precomputed values. The solution technique may be summarized by stating that each h is computed by sampling current error and multiplying by the appropriate gain constant (K_k) by using the CDC-168 arithmetic unit in the multiply integer mode in conjunction with the CDC-160 computer.

Programming arithmetic operations for these two units is quite straightforward when Subroutine Arith, written by Professor M. L. Cotton, is used. However, the characteristics of the CDC-168 required manipulation of the sensed error signals. Specifically, for an arithmetic operation, the 168 requires two 22-bit operands, and supplies a 22-bit solution. Each operand is composed of two 12-bit 160 words, with the least significant half of the number in an even numbered cell and the most significant half of the number in its odd numbered mate. The first bit in each cell pair is a sign bit. It is this sign bit in the first bit of each of the two words composing the operands that require the manipulation. For the K multipliers which remain constant, proper values may be inserted into both storage cells

manually before the program runs. However, for the sensed error signal, negative numbers received from the converter create a problem. For the 168 to function properly, the sign bits for both portions of the operands must agree. To handle this problem, the following procedure was employed. To derive the proper h, the product of K_k and e(t) is required. After e(t) is sensed, it is placed in cell 0024 as the least significant half of an operand. The most significant half, cell 0025, is preset to zero. For negative step inputs, the error signals read in from the A/D converter are positive numbers, and the multiplication process may proceed immediately. For positive step inputs, however, the error signals sensed from the converter are all negative numbers. To maintain sign bit consistency with cell 0025, this negative number is complemented and placed in cell 0024. The multiplication operation then proceeds. The product, of course, is of the wrong sign so it must be complemented prior to being transmitted to the system. This procedure will be defined completely in the flow chart for Program 5.

The other major innovations of Program 5 involved attempts to obtain greater accuracy in the solution and maintain high resolution in the converter. As previously discussed, the multiplication and division by ten in the operational amplifiers were included. To achieve greater accuracy in computing the h's, 100 times K_k was set in the computer as the multiplying constant. The products were then divided by 100 prior to being output to the D/A.

As before, a sampling interval of T = 0.3 seconds was used. The gain constants for this system were computed to be: K_0=0.421 and K_1 = -0.398. A flow chart for Program 5 is shown below. A complete text of Program 5 is presented in Appendix VI.

Program 5

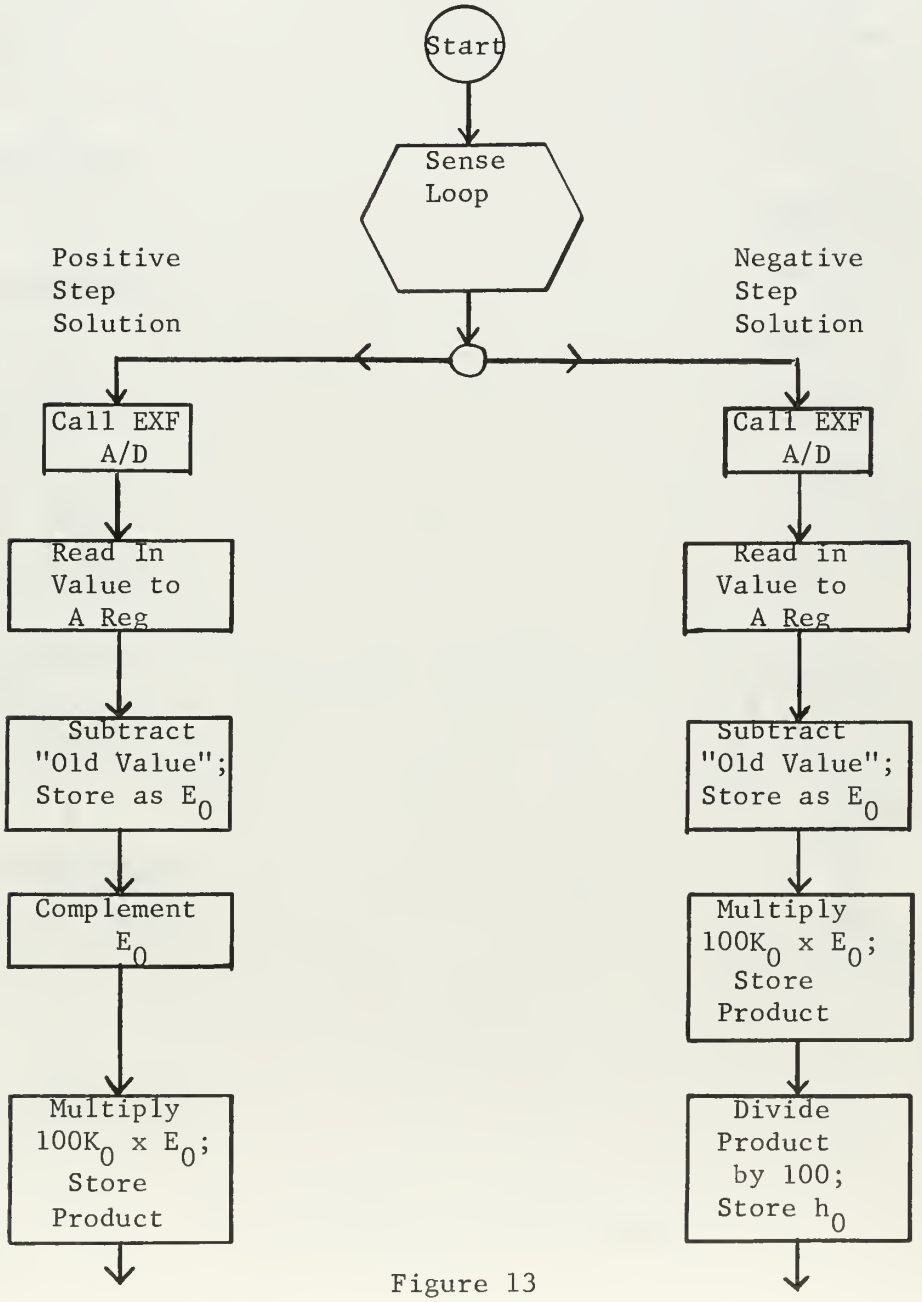


Figure 13

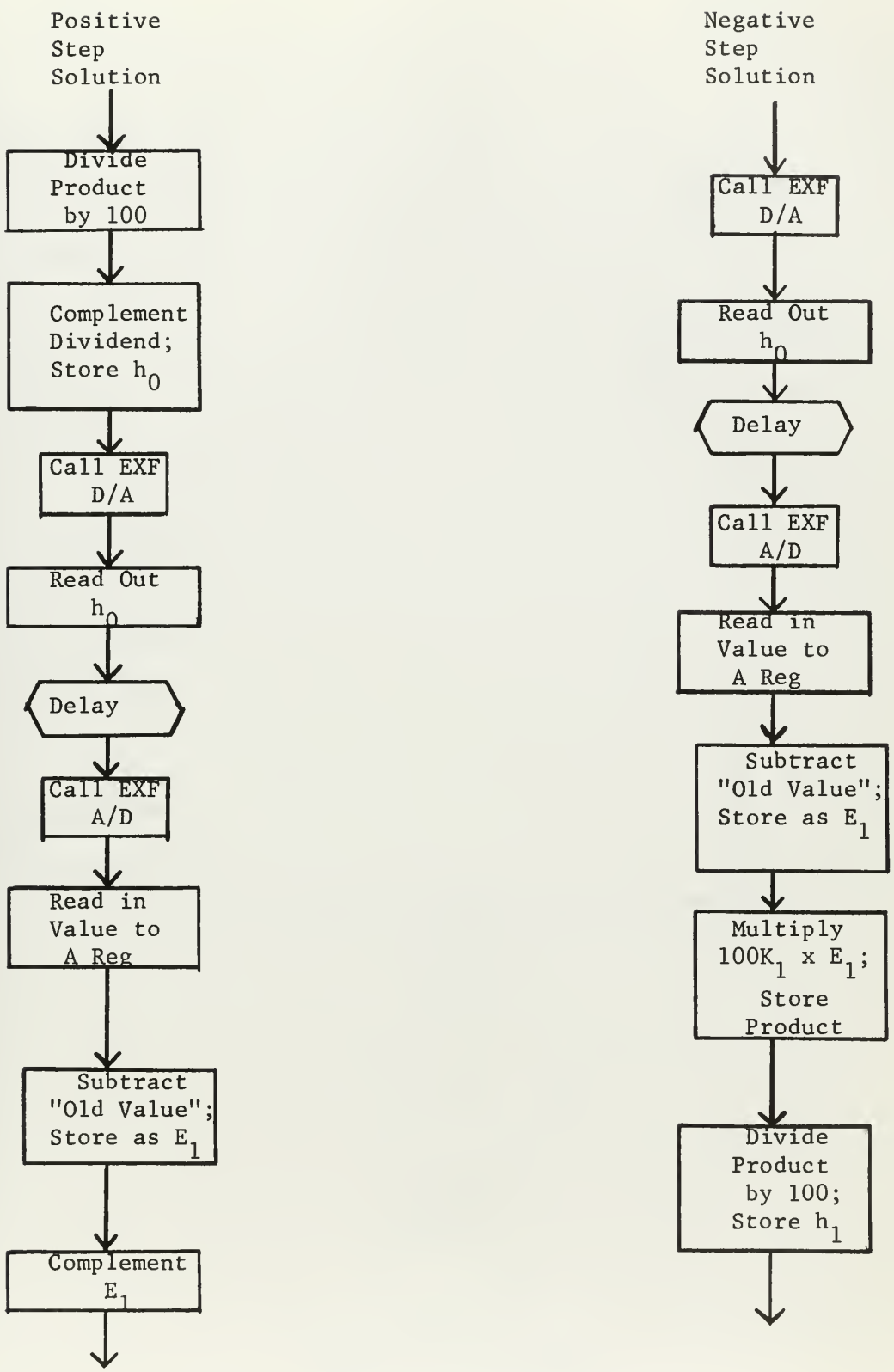


Figure 13 (continued)

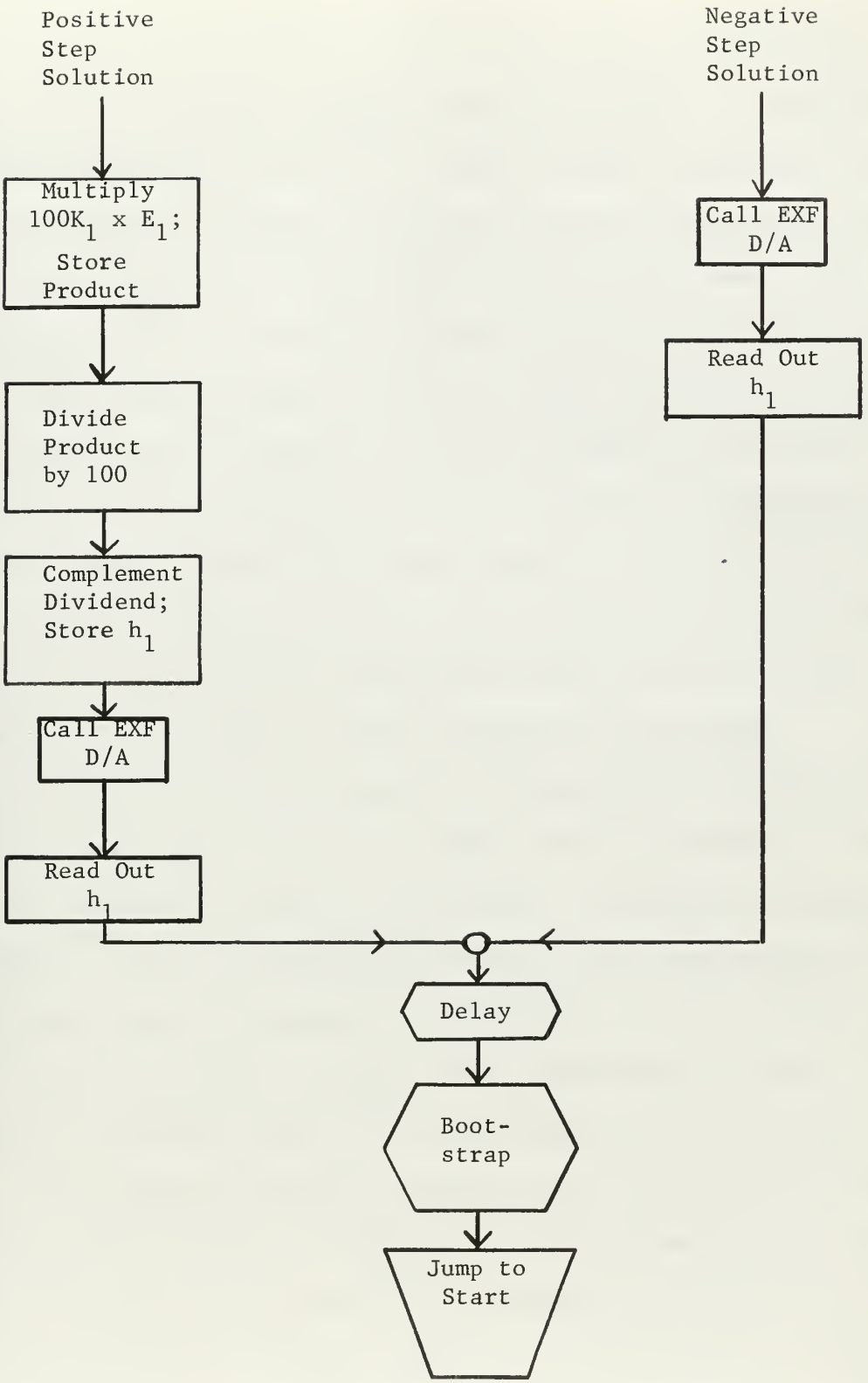


Figure 13 (continued)

As noted before, Program 5 marked the culmination of efforts in conjunction with the temporary control philosophy. Although the results of Program 5 were good, and gave valuable insight into digital control technique, Program 5 was merely a vehicle through which the validity of variable gain amplifier theory could be tested. The temporary control philosophy, although perfectly satisfactory for control of a single system, is unsatisfactory as a general philosophy for digital control because it excludes utilization of one of the primary advantages claimed for digital controllers: simultaneous control of many systems by the same computer. The authors believe that the realistic application of digital control methods is closely tied to the capability for simultaneous control of many systems. Toward this end, a final control philosophy was developed.

The goal of this final control philosophy was to achieve this simultaneous control. This goal implies that the computer be "time-shared" among the systems to be controlled. Therefore, the significant difference between the temporary and final control philosophies is one which enables the computer to time-share its control function.

Before discussing the final control philosophy in detail, specific performance goals for this philosophy will be discussed. The authors wanted to develop techniques for controlling two or more systems simultaneously by employing one digital computer on a time-share basis. The control was to be achieved by the variable gain amplifier method. To achieve generality, it was desired to include

the capability for the systems to be controlled at different sampling rates. Furthermore, it was considered necessary to maintain the capability to handle simultaneous step inputs to each system without performance degradation.

It may be recalled that the temporary control philosophy called for a return to a free-running "continuous" mode at the completion of the solution. The major difference in the final control philosophy is that at the completion of the second sampling period, the computer returns to a mode in which D(Z) is still 1.0, but instead of a sampling interval approaching zero, a sampling period of definite finite length is employed. Although this innovation does not change the response characteristics of the control solution, it marks the change from a partially discrete, partially continuous system, to one which is completely discrete. It is this change which enables the development of the required time-sharing computer operation for multiple control.

Having discussed the major difference in the control philosophies, the final control philosophy will be covered in detail. As before, with no signal input to either system, the computer resides in a Sense Loop. In the Sense Loop each system under the control of the computer is sampled and tested for a step input. If a step input greater than a threshold is not detected, the exact signal sampled is returned to the system. (D(Z) = 1.0) Each of the systems under control is handled in this fashion. When all systems have been sampled, the computer goes through a basic delay. This basic delay

is so named because it is a building block for all delays required for implementing variable gain amplifier solutions of any length on any system controlled by the computer. All required solution delays are formed as integer multiples of the basic delay. It is noted that the basic delay is much longer than the time required to sample and test all systems being controlled.

When a step input is received by one of the systems, an error flag for the appropriate system is set and the computer exits from the Sense Loop to a Solution Routine. In the Solution Routine, the first control voltage for the flagged system is transmitted from the computer, and all other system flags are checked for possible simultaneous step inputs. If not, the computer goes through one basic delay and then senses all other systems for possible step inputs. If no other systems have received signals, the computer returns to the start of the Solution Routine and begins to count the number of times the basic delay has been entered. After each pass through the basic delay, all other systems are checked for possible step inputs. When the pass count reaches a number which indicates that the first control voltage has been in the proper length of time, the computer computes and outputs the second control voltage, and a new pass count begins. If, at any time during the solution, another system receives a step input,it is flagged, and on the next pass through the Solution Routine, a solution similar to the one explained above is begun.

When a variable gain amplifier solution for one of the systems is completed, the computer jumps to a Return Routine. In this Return

Routine all counters for the appropriate system are zeroed and the error flag is set to zero. The system is then bootstrapped by the procedure previously explained, and returned either to the Sense Loop or the Solution Routine, if the error flag of another system is set. In any case, after being bootstrapped, the system returns to a mode in which the characteristics of the control function are D(Z) = 1.0, and T = the basic delay.

The characteristics of the final control philosophy may be summarized by stating that any single system which is being controlled operates in a state in which D(Z) = 1.0, and T = the basic delay when there is no step input above the threshold level. Thus, for small step inputs and noise inputs, the system would be controlled in the above mode. When a step input greater than the threshold level is received, a variable gain amplifier solution is executed and the control mode then returns to that specified above. In the time sharing mode, each system controlled is operated in this same manner, with the option of making the solution sampling period any integer multiple of the basic delay.

To test the final control philosophy, Program 6 was written. For purposes of this test, two systems were time-shared. One of the systems used was the real system used in previous tests. The second system was an analog simulation of a plant having the following characteristics: $G(S) = \frac{1}{s(s + 1)}$. The basic delay was established as 0.025 seconds. The solution sampling rate for the real system was 0.3 seconds; for the analog system, 1.0 seconds. The only

restriction for Program 6 was that it was written to control only two systems simultaneously. The basic delay time may be chosen arbitrarily, and the solution sampling rates may be any integer multiple of the basic delay. Computation of h_0 and h_1 was accomplished in the manner of Program 5. The hardware arrangement for the real system was the same as that employed in conjunction with Program 5, and the summing amplifier arrangement in and out of the converter for the analog system is identical to that of the real system.

A flow chart for Program 6 is shown on the following pages. It is broken down into three sections: Sense Loop, Solution Routine, and Return Routine. A complete text of Program 6 is presented in Appendix VI.

To simplify the flow chart for the Solution Routine, the negative step solution for System 1, and the positive step solution for System 2 have been omitted. The omitted processes are identical in form to those shown. Furthermore, the "D/A h_k" blocks, shown in the flow chart as predefined processes are handled exactly as in Program 5.

Before presenting the flow chart for the Solution Routine, the following indices are defined:

I = the number of passes through the basic delay for the current h for System 1.

J = the number of h's transmitted to System 1.

P = the required number of passes through the basic delay to achieve the specified variable gain amplifier solution sampling rate for System 1.

K = the number of passes through the basic delay for the current h for System 2.

L = the number of h's transmitted to System 2.

Q = the required number of passes through the basic delay to achieve the specified variable gain amplifier solution sampling rate for System 2.

The flow chart for the Solution Routine is shown on the following page.

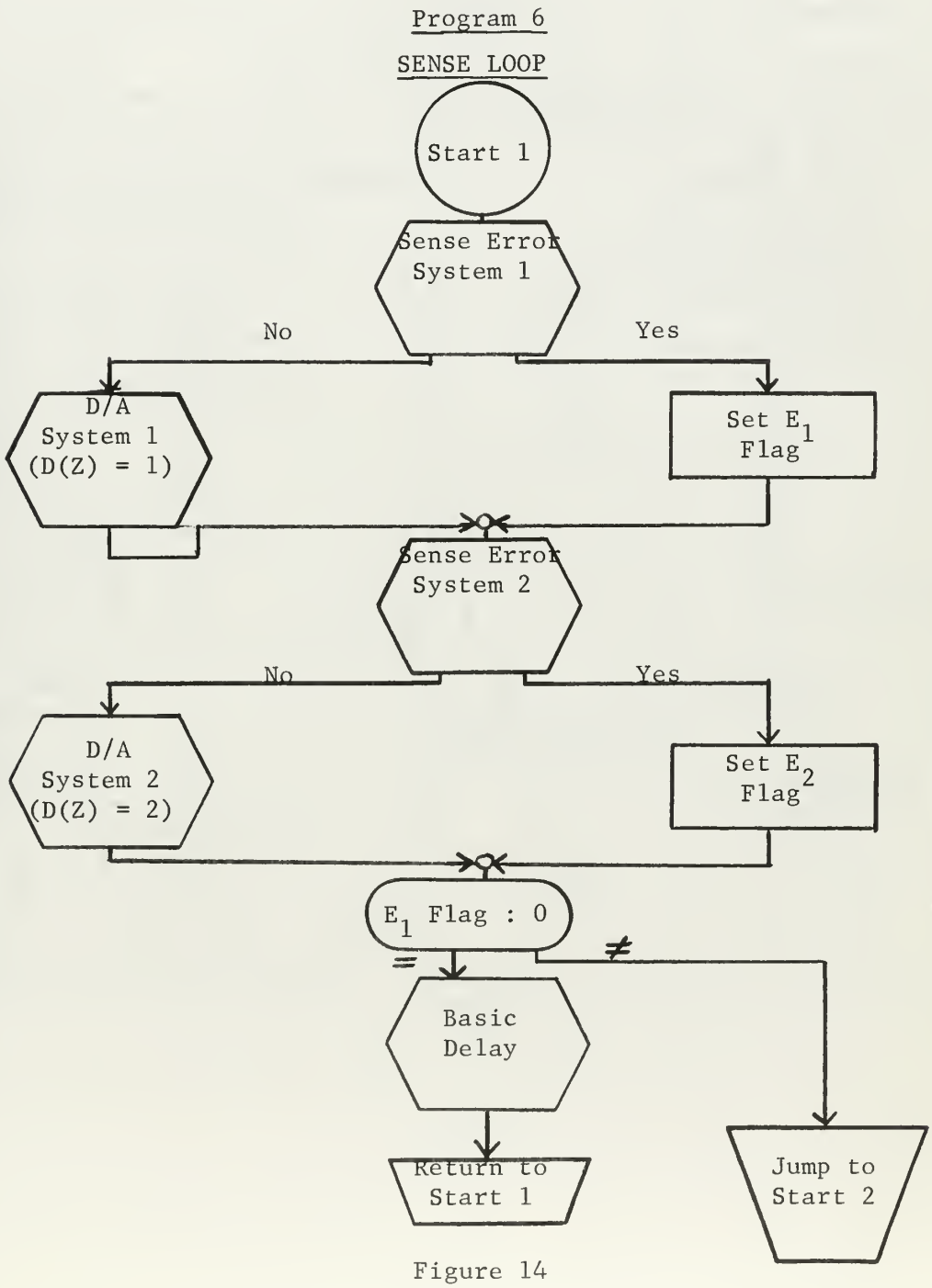


Figure 14

Program 6

SOLUTION ROUTINE

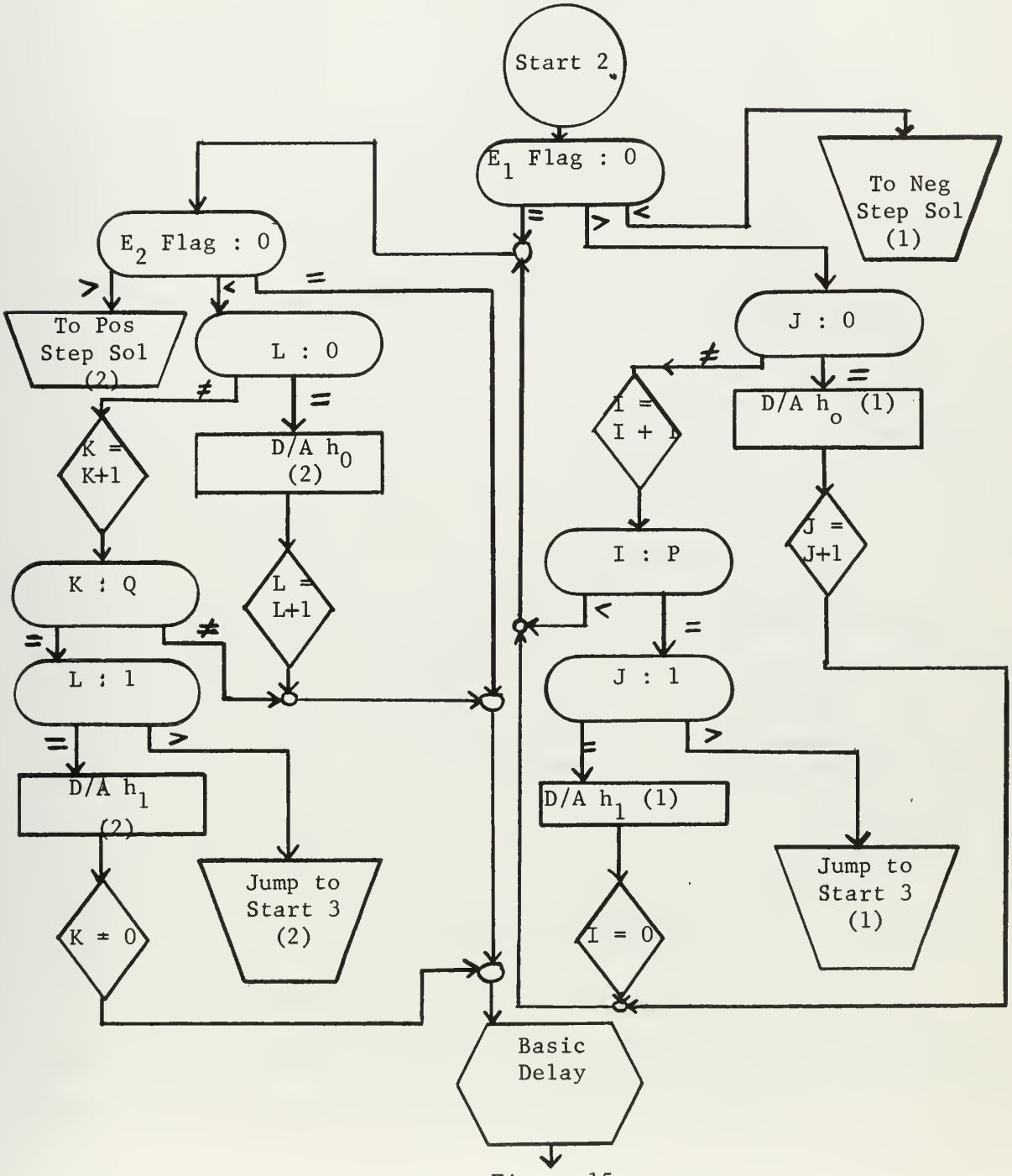


Figure 15

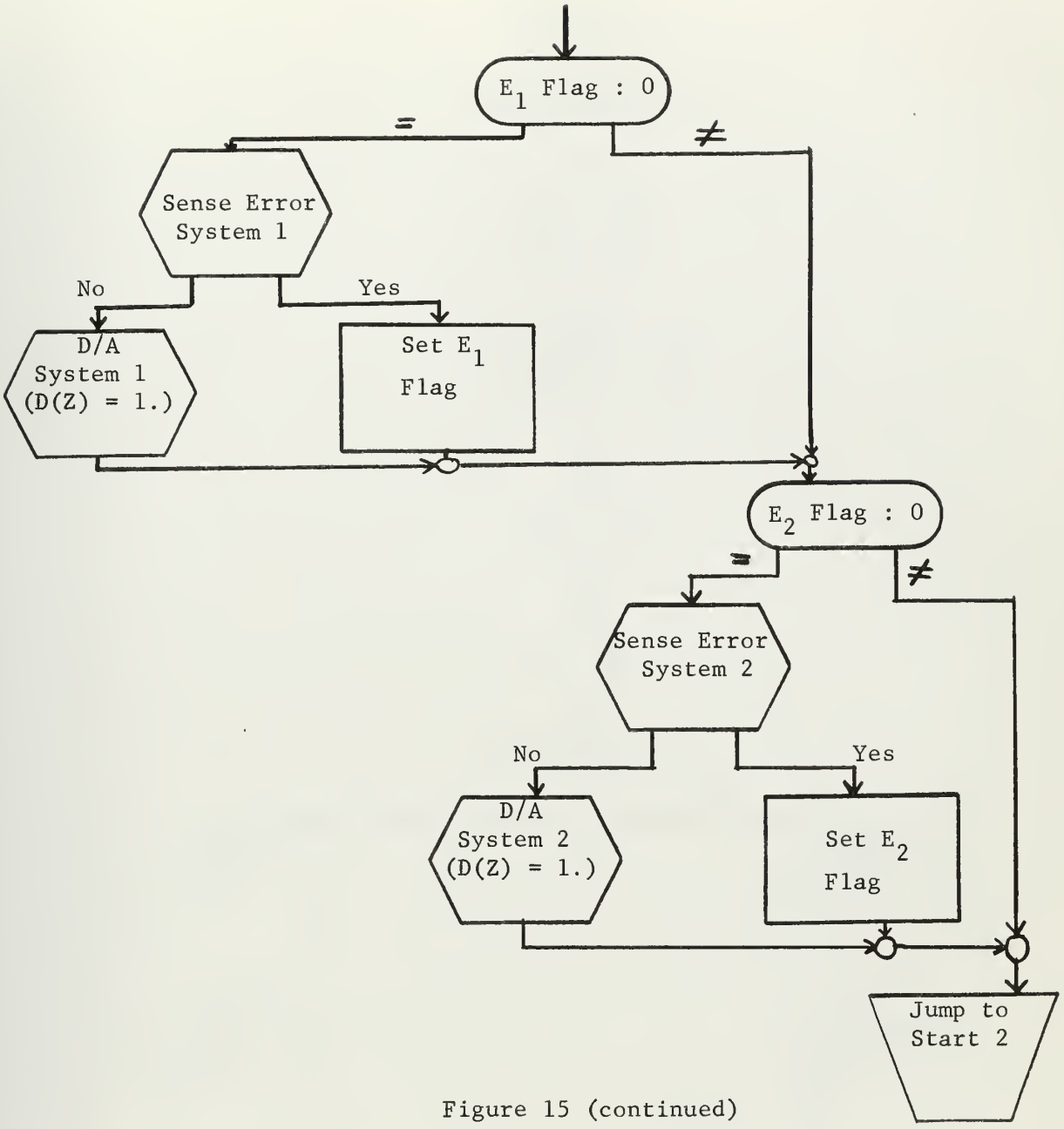


Figure 15 (continued)

The Return Routine is shown on the following page. The Return Routine will be shown for System 1 only. That for System 2 is identical except for the fact that different indices are zeroed.

Program 6

RETURN ROUTINE

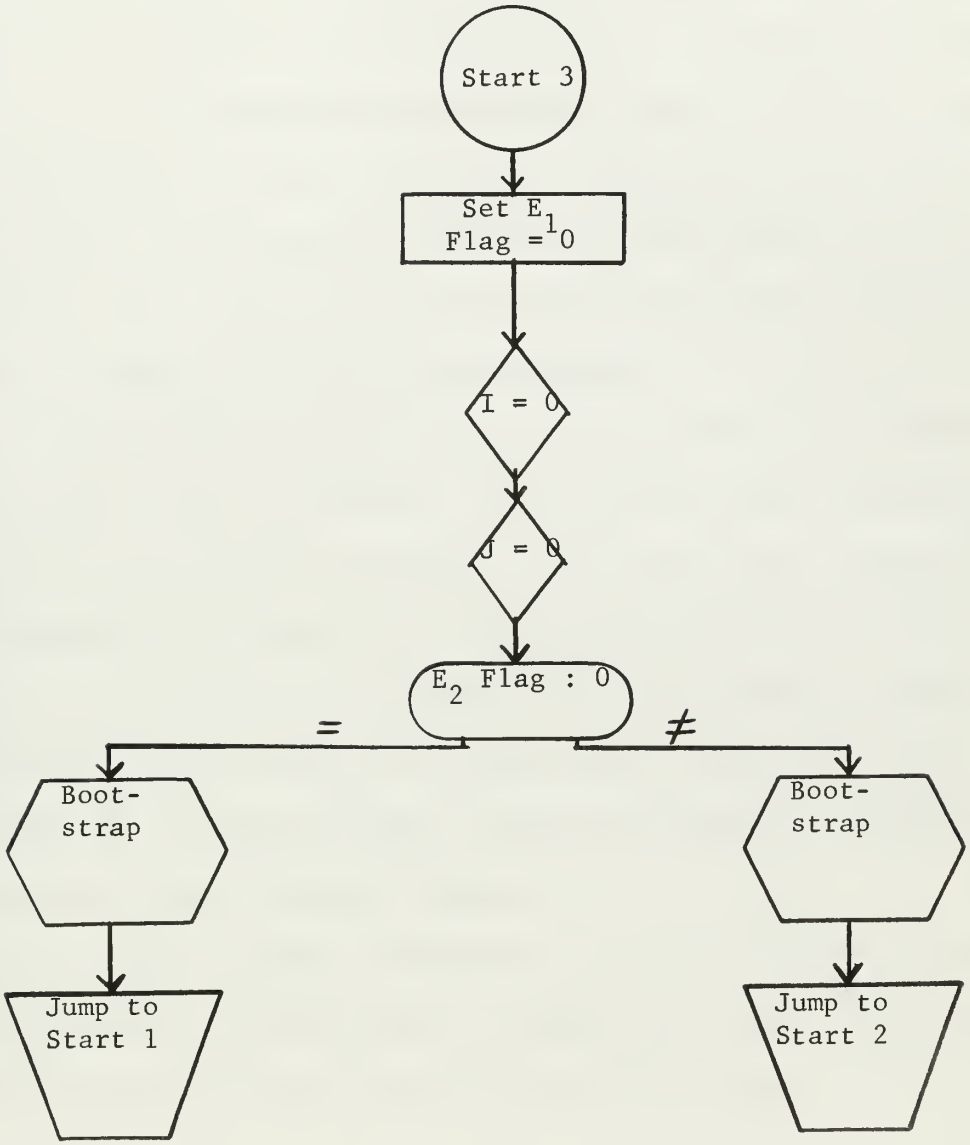


Figure 16

2.3 Results

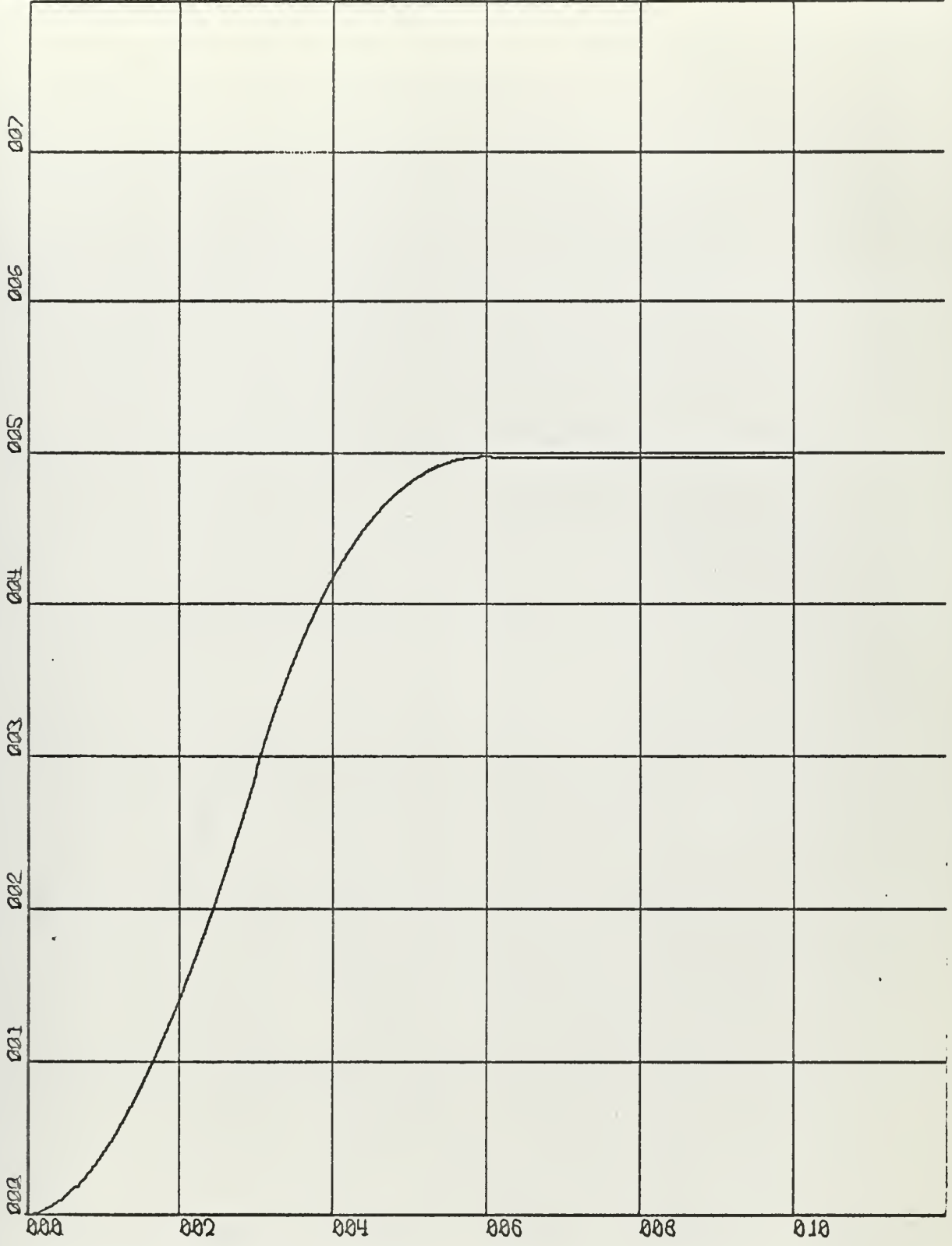
The results of Program 6 were excellent. Both systems were controlled with predicted speed and accuracy. The real system response was equivalent to that obtained from utilization of Program 5. It is interesting to note that the system simulated on the analog computer exhibited exact theoretical response to the variable gain amplifier solution. This, of course, is due to the fact that there were no non-linearities in the simulated system. However, the exact theoretical solution obtained does validate the final control philosophy employed.

The time-share program was tested for all possible time combinations of step inputs to the two systems. In no case was system response degraded once the solution was started. In the worst case, there was a 25 millisecond delay between a step input and the start of a solution. This occurred when a step was entered just as the computer started a pass through the basic delay in the Sense Loop. The delay, in this worst case, which is a random occurrence, was not discernable on the Mark II Brush Recorder used to measure response.

On the following pages, the response curves for the two systems controlled in the time-share mode by Program 6 are shown. Also shown are the theoretical response curves for the real system.

2.4 Extension of Results

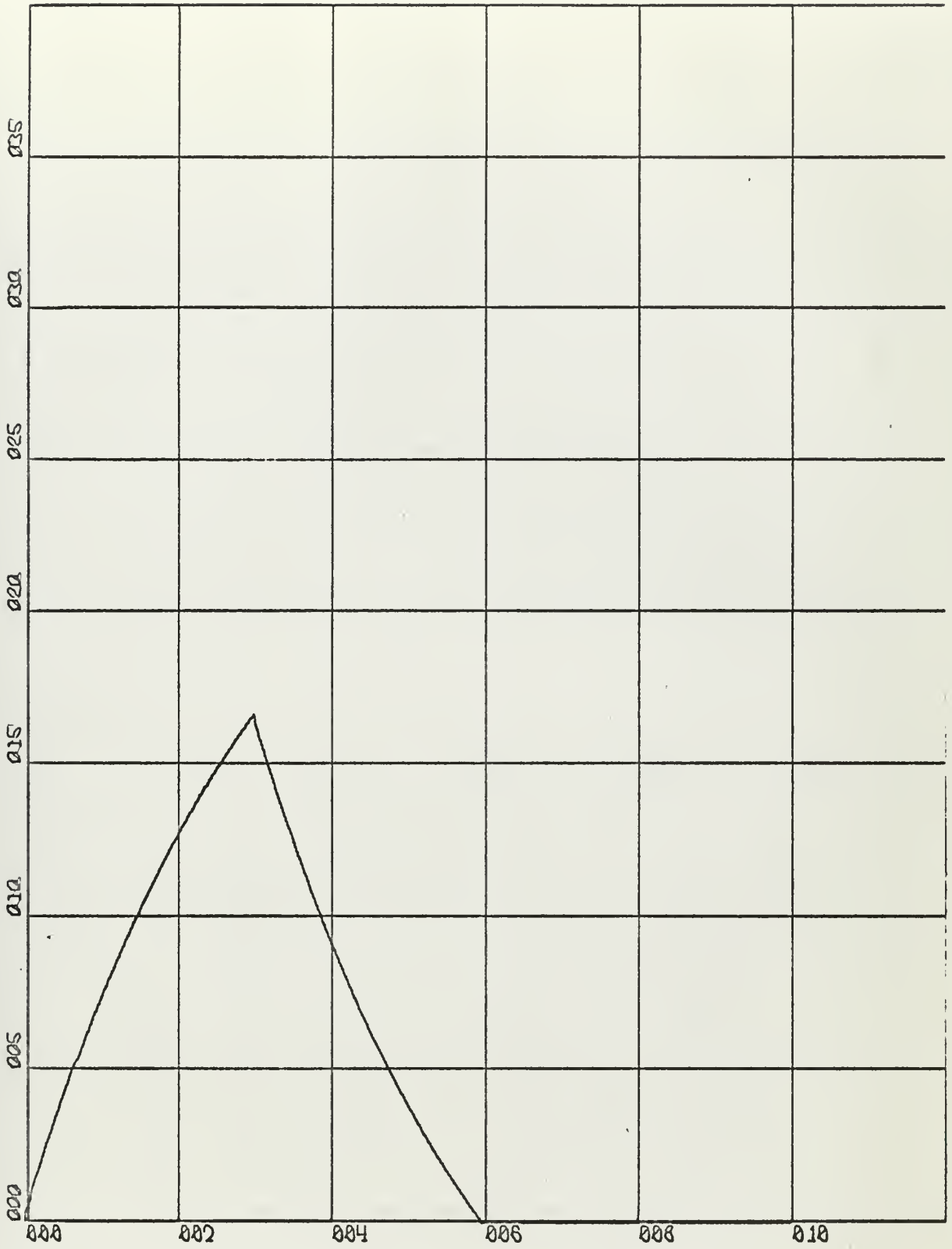
Due to the limitations imposed by the D/A, only two systems could be simultaneously controlled in the Digital Control Laboratory. To fully substantiate the validity of the final control philosophy, one must consider the feasibility of this philosophy in controlling more



X-SCALE = 2.00E-01 UNITS/INCH.
Y-SCALE = 1.00E+00 UNITS/INCH.

BROWNE E.R. REAL SIM
RUN 1 OUTPUT VS TIME

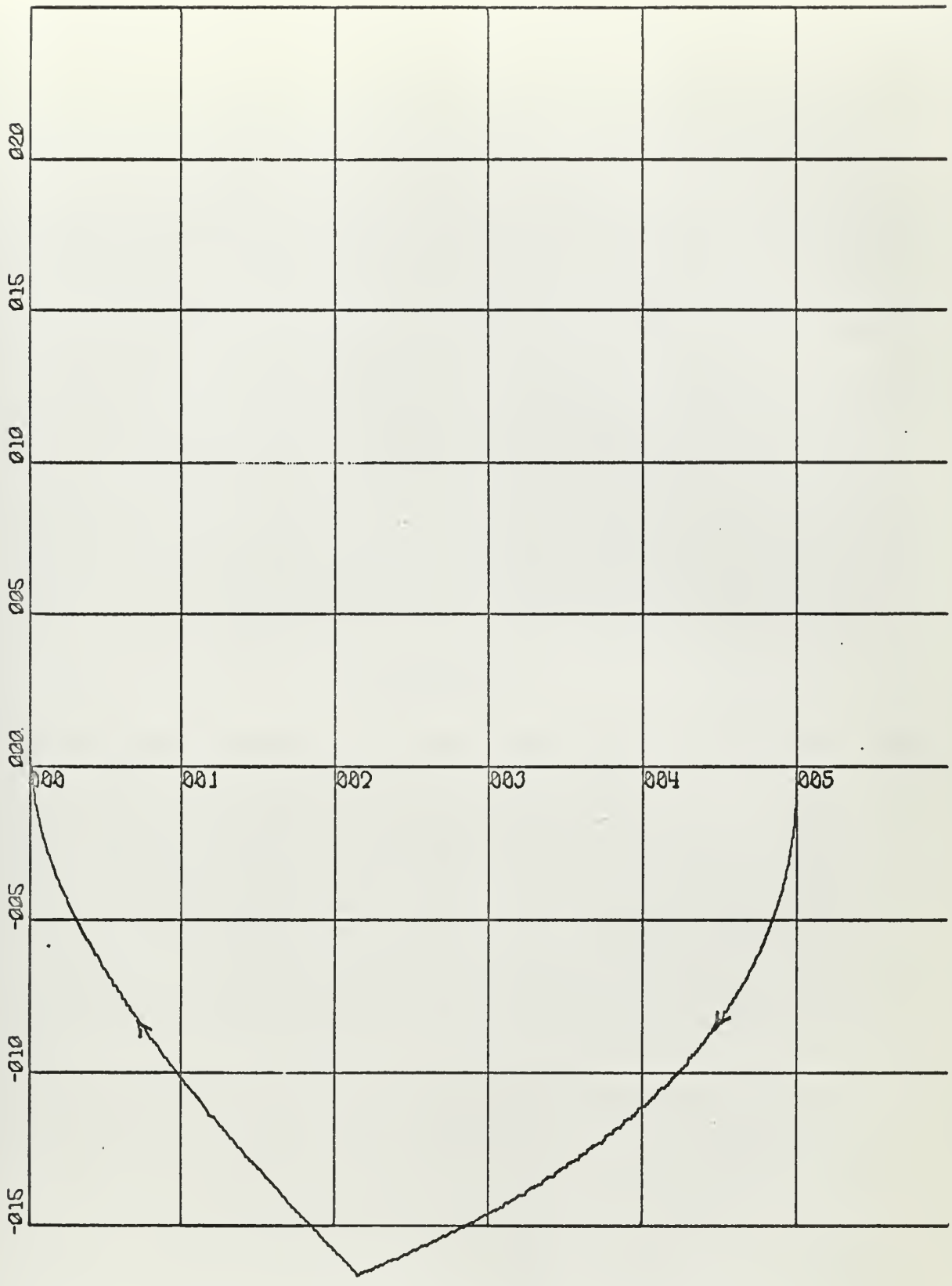
Figure 17



X-SCALE = 2.00E-01 UNITS/INCH.
Y-SCALE = 5.00E+00 UNITS/INCH.

BROWNE E.R. REAL SIM
RUN 1 VELOCITY VS TIME

Figure 18



K-SCALE = 1.0 00 UNITS/INCH.
Y-SCALE = 5.00E+00 UNITS/INCH.

BROWNE E.R. REAL SIM
RUN 1 EDOT VS ERROR

Figure 19

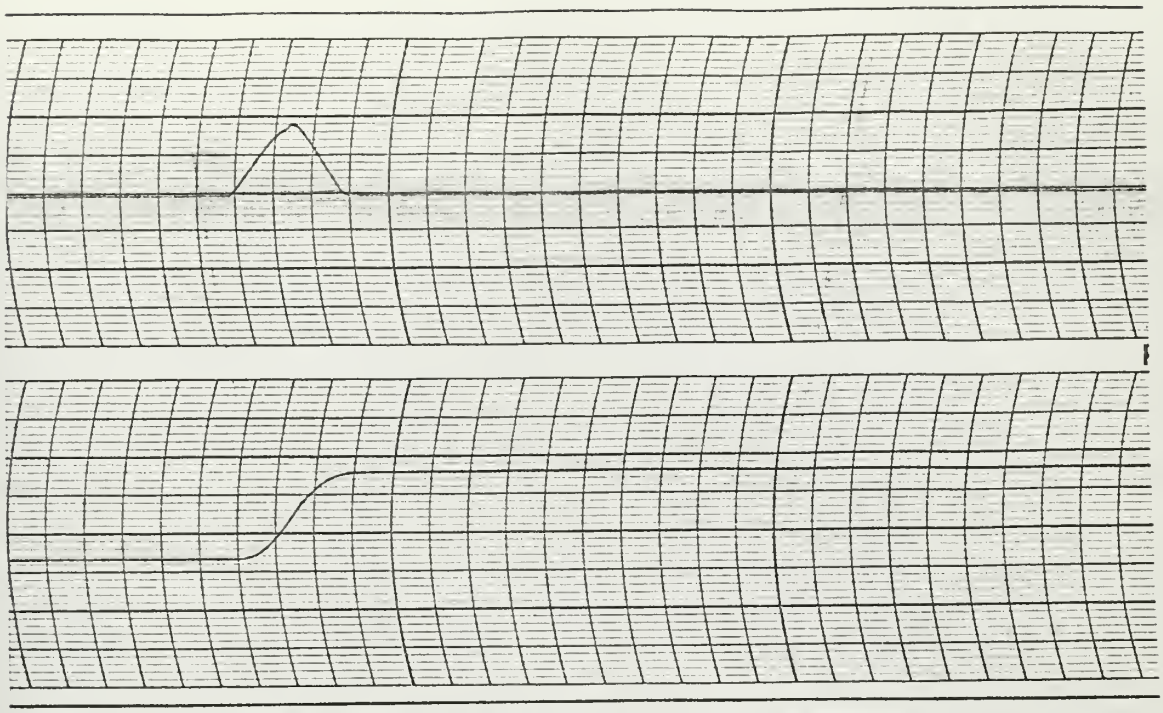


Figure 20
Real System

Upper Graph Velocity vs. Time; Lower Graph Position vs. Time

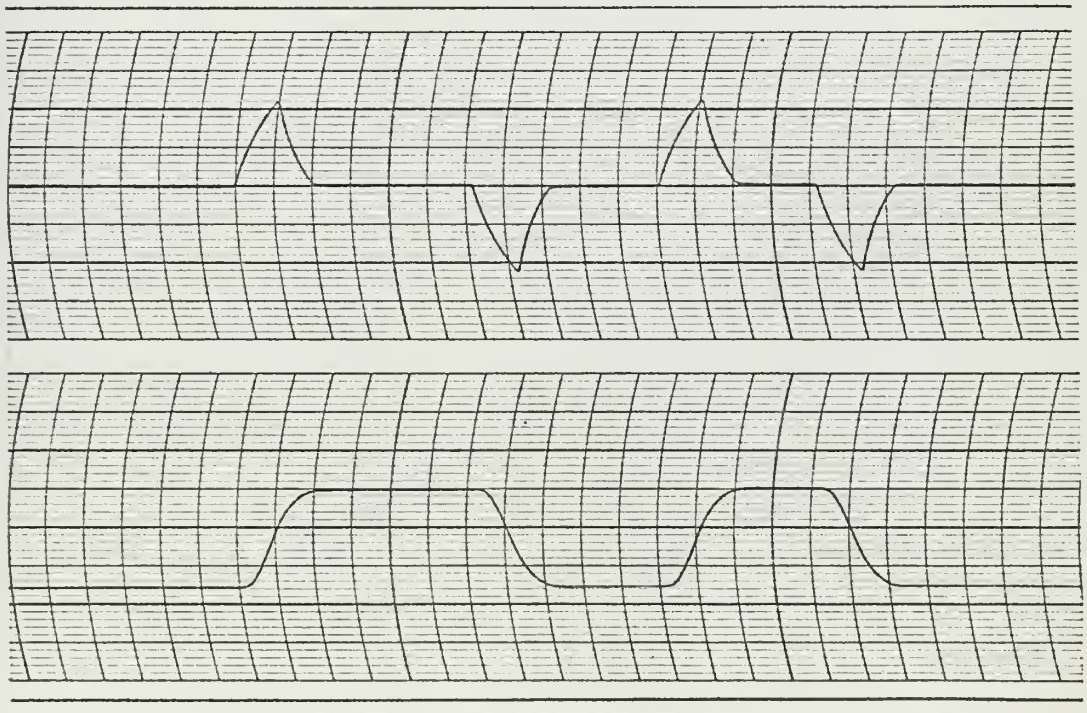


Figure 21
Analog Simulation

Upper Graph Velocity vs. Time; Lower Graph Position vs. Time

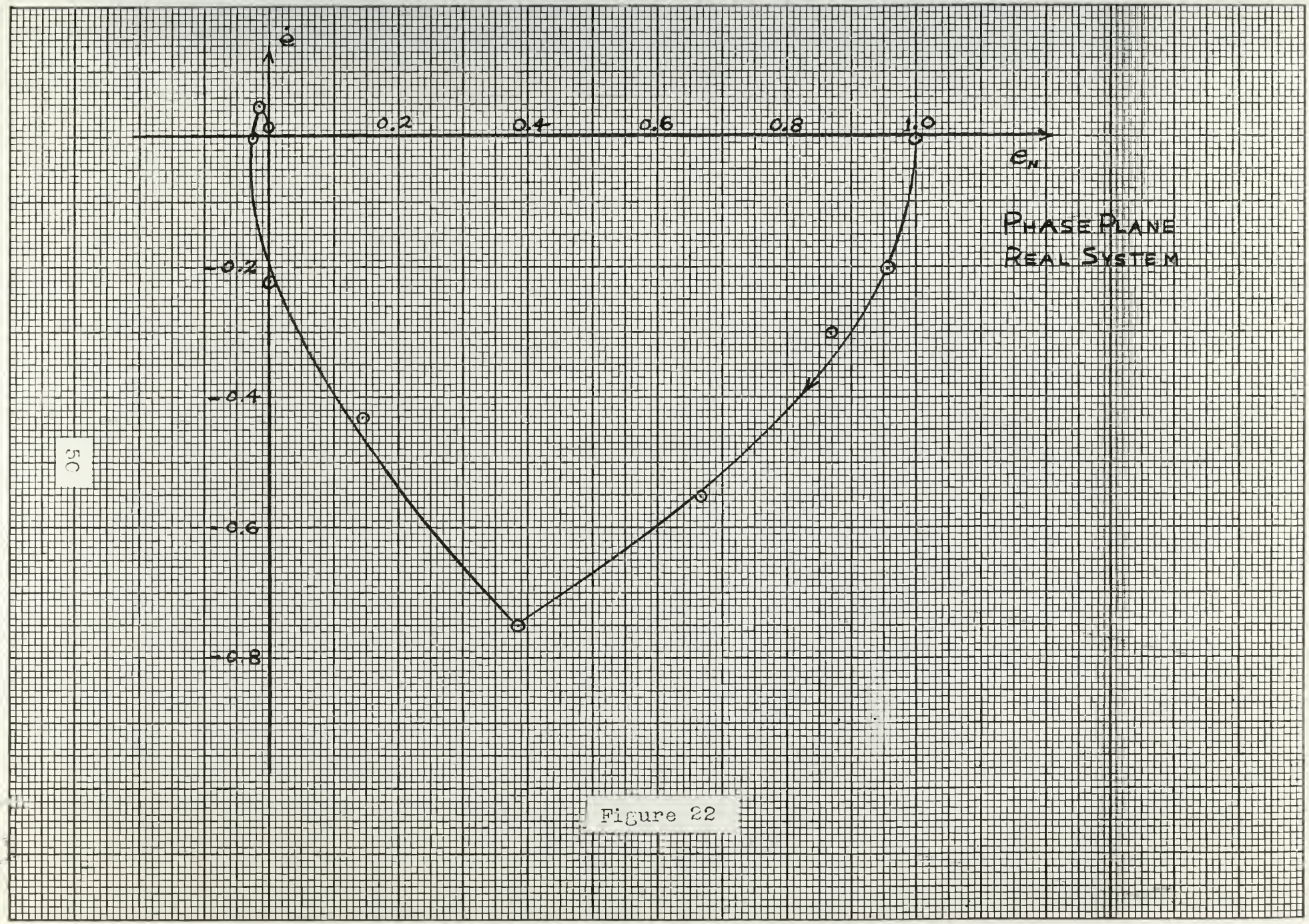
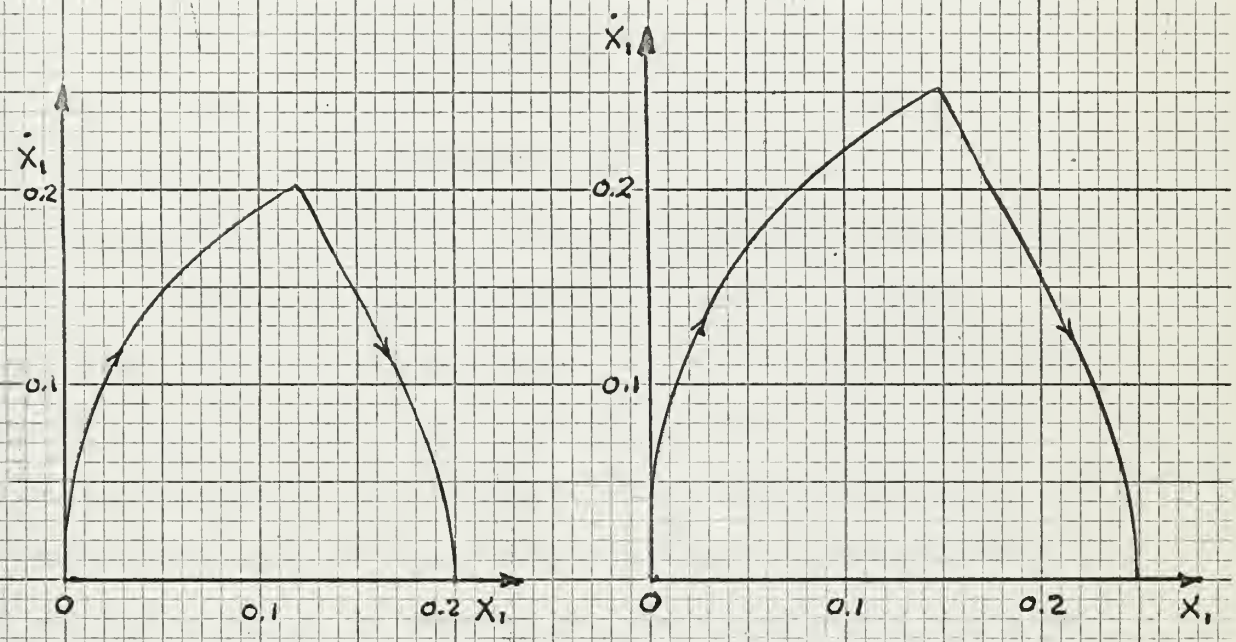


Figure 22

PHASE PLANE
ANALOG SYSTEM



0.2-VOLT STEP

0.25-VOLT STEP

Figure 23

than two systems. Let us consider this feasibility with respect to the equipment used throughout the laboratory tests, assuming only that the number of channels in the converter may be arbitrarily expanded. (The authors feel that the problems associated with more difficult tasks in the computer field are too often dismissed with a bland inference that a bigger, faster computer can do any job. The dollar economics of the control problem preclude this approach.)

Using Program 6 as a basis for time-sharing control of many systems, the CDC-160 has sufficient memory capability to handle 12 systems simultaneously. By sub-routinizing wherever possible, this number could probably be increased to 15. For 15 systems, however, the sampling time, governed primarily by the analog to digital conversion time of 120 microseconds, becomes an appreciable proportion of the basic delay time. In the Sense Loop, for example, the time to sample A/D, check for step inputs, and return D/A would be approximately two milliseconds compared to the 25 milliseconds of the basic delay. This added delay in the Solution Routine would produce significant errors in the variable gain amplifier solution. If the basic delay could be satisfactorily increased to 50 or 100 milliseconds, the two milliseconds required to sample the 15 systems would become insignificant and not seriously degrade the solution. The authors feel that employment of a 50 or 100 millisecond basic delay, which limits the fastest solution to a step input to 100 or 200 milliseconds, is not unreasonable in most practical applications.

The foregoing discussion has made one tacit assumption. That assumption is that the number of simultaneous inputs is limited to three

of the 15 systems. This is due to the 900 microsecond multiply time required by the CDC-168. This, of course, excludes the desired generality for handling any step input at any time for any system. With the equipment used, the 160 in conjunction with the 168, there is no way to compensate for this problem.

At this point it is worthwhile to look at the advantages of using a different computer. Specifically, the CDC-160A would adequately solve the problem mentioned above. The 160A is a computer quite similar to the 160 in all respects with the additional capability for fast multiplication and division. Economically, the 160A is comparable in cost to a 160-168 tandem, and makes a much more compact unit.

On the basis of the foregoing tests and theoretical extension, the authors feel that with a small, relatively inexpensive computer, such as the CDC-160A, it is quite feasible to control up to 15 systems simultaneously. The control capability will vary with the capabilities of the brand and type computer selected. The main thesis, however, is that an excellent, multi-system, digital control scheme may be implemented without going to the larger general purpose computer.

CHAPTER III

3.1 Conclusion

It was noted in the Introduction that the basic goals of this thesis were the controlling of hardware by digital methods, and the developing of techniques for implementing time-sharing control of many systems from the same computer. The results obtained have been detailed. In this section the most important findings will be summarized and their relevance to the general feasibility of digital control systems will be noted.

The most significant aspect of controlling the hardware was the necessity for going to the "continuous" mode (D(Z) = 1. and a small T) to bring the error to zero at the conclusion of the solution to a step input. This characteristic, caused by the non-linearities of the system, would seem to be a significant disadvantage to this type control. However, this was not the case. In the worst case, the servo position was within ten percent of the desired final position at the moment of the switch to the "continuous" mode. Furthermore, at the switch time, the servo velocity was in the proper direction at a decreasing magnitude. This condition permitted the system to settle rapidly to the desired position.

Of course, to realistically evaluate this control method, one must establish a performance criterion as the basis for comparison with other control methods. The criterion selected was one which measured minimum time to reach a position within ten percent of final value and

to remain within this ten percent boundary while settling. On the basis of this criterion, the time for a unity feedback continuous system to meet the conditions was 1.2 seconds, and 800 milliseconds with the addition of optimum tachometer feedback. These figures were obtained with the amplifier gain potentiometers set at the same levels used for the digital operation. By increasing the amplifier gains, a time of 500 milliseconds was obtained with tachometer feedback. Using a variable gain amplifier solution sampling rate of 300 milliseconds, a time of 600 milliseconds was required to meet the criterion. The relatively low saturation level of the amplifiers precluded the use of sampling rates of 100 or 200 milliseconds, but on the basis of responses obtained at rates of 300, 400, and 500 milliseconds, it follows that improved amplifiers would allow a solution time of 200 milliseconds. In general, the fastest speed of response obtainable by the variable gain amplifier method is limited by the saturation level of the amplifiers. Similarly, the response speed obtainable by a bang-bang technique is limited by this saturation level.

On the basis of these results, the performance characteristics of the digital control method indicate that this method is competitive with other control methods. With improved amplifiers, response speeds from the digital methods can be much faster than those obtainable from continuous methods. Optimum bang-bang techniques approach the response speeds obtainable with the variable gain amplifier method.

There are, of course, many criteria by which control systems

may be judged. The criterion above is one of speed of response. Although other criteria were not evaluated, it is noted that the digital technique is, in general, quite adaptable. For example, the basic hardware and software employed by the authors is readily adaptable to a criterion which calls for minimum fuel expenditure.

The results of the time-sharing program have been fully discussed and the feasibility of extending the program to control many systems has been outlined. The basic approach of utilizing a small computer has precluded a discussion of a further extension of the technique. That is, using a large, general purpose computer as the digital controller, with other computing tasks being time-shared with the control function. This is feasible with no change in the basic program heretofore used in conjunction with the CDC-160. In examining this program, one can readily see that the vast majority of time is spent in the basic delay block. Time spent in the basic delay is ideal for carrying out other computing tasks. This could readily be implemented on a general purpose computer with a real time clock and interrupt capability, and an executive routine to direct the proper sequence of computing tasks not associated with the control task. The ramifications of this large computer capability are indicated by Slaughter and Lackowski in a paper presented to the 1963 National Convention on Military Electronics in which they state: "In those cases in which a digital computer is available, digital control requires less hardware than does conventional control." [4] The presumption accompanying this

statement was that the digital computer was required for other tasks, and the controlling tasks could be satisfactorily time shared.

In view of the performance characteristics of the variable gain amplifier method and the time-sharing capabilities developed, the authors have come to the conclusion that for many applications, digital control methods are presently feasible and superior to other control methods. Due to the present cost of computers, these applications are presently limited to large processes or areas where the advantages of digital control are overwhelming. A specific example of such an application is in large processes such as those in the chemical or petroleum industries in which multiple systems must be simultaneously controlled. Similarly, a single large process with multiple inputs may be controlled by digital methods. A military application would be the extension of the NTDS to include the servo control of gun mounts and missile launchers from the master computer. A final application might be in the aerospace field in which the new techniques for optimum control on the basis of minimizing a given cost function are quite adaptable to digital techniques.

Although immediate feasibility for digital control is limited to processes of the type mentioned above, the decreasing size and cost of digital computers increase their field of application.

The investigation for this thesis was limited in scope to permit the extensive study of a single method. The results from this study may, in general, be extended to other digital control techniques. The authors feel that the results of this study of digital control validate its basic feasibility, and indicate the value of further study in the field.

BIBLIOGRAPHY

1. Dejka, W. J. and Kershaw, W. V. Digital Sampled-Data Feedback Systems for Shipboard Applications, Progress Report. Navy Electronics Laboratory Research and Development Report 1142, 29 October 1962.

2. Jensen, M. E. Optimum Control of a Nonlinear Second-Order Plant Using a General-Purpose Computer. Navy Electronics Laboratory Research and Development Report 1220, 24 April 1964.

3. Kuo, B. C. Analysis and Synthesis of Sampled-Data Control Systems. McGraw-Hill, 1963.

4. Slaughter, J. B. and Lackowski, D. Digital Control of a Tactical Weapon with a Shipboard General-Purpose Digital Computer. Conference Proceedings of MIL-E-CON 7 National Convention on Military Electronics, 1963.

5. Pearson, M. and Tardif, D. W. Design Analysis and Test of a D. C. Servomechanism Utilizing Direct Drive. United States Naval Postgraduate School, 1959.

6. Slaughter, J. B. Quantization Errors in Digital Control Systems. Institute of Electrical and Electronics Engineers Transactions on Automatic Control, v. AC-9, January, 1964: 70-74.

APPENDIX I

DESIGN OF DIGITAL CONTROLLERS AND SIMULATION RESULTS

In addition to the design of a digital controller for a step input to a $\frac{1}{s(s+1)}$ plant, a digital controller for a unit ramp input was also designed. Referring to Figure 1 and Figure 2 in Section 1.2 we arrive at the following equations.

$T = 1.0$

$$x_1\left[(k+1)T\right] = \left[1-0.368K_k\right] x_1(kT) + 0.632x_2(kT) + (0.368K_k)r(kT)$$

$$x_2\left[(k+1)T\right] = (-0.632K_k)x_1(kT) + 0.368x_2(kT) + (0.632K_k)r(kT)$$

For a unit ramp input $r(kT) = k$.

$k = 0$; $x_1(0) = x_2(0) = 0.0$

$x_1(T) = 0$; $x_2(T) = 0$

$k = 1$

$x_1(2T) = 0.368K_1$

$x_2(2T) = 0.632K_1$

To obtain a solution we must set $x_1(nT)=r(nT)$ and $x_2(nT)=1$. We therefore must proceed to the next period in order to solve for the given conditions.

Therefore $k = 2$

$$x_1(3T)= \left[1-0.368K_2\right] (0.368K_1) + (0.632)^2K_1 + 2(0.368K_2) \triangleq 3.0$$

$$x_2(3T)= \left[-0.632K_2\right] (0.368K_1) + (0.368)(0.632K_1)+2(0.632K_2) \triangleq 1.0$$

from which

$$K_1 = 3.82 \; ; \; K_2 = 0.31$$

and

$$h(T) = 3.82 \; ; \; h(2T) = 0.183$$

It should be noted that for this system to maintain a zero error after it has arrived "home" a constant output of h(nT)=1 must be fed into the continuous system. This will keep the system moving at a unit velocity and thus maintain a zero system error.

A digital controller using the same design theory was developed for a $\frac{1}{(s+1)(s+2)}$ plant in response to both unit step and ramp inputs. The derivations and explanations follow. The procedure used is the same as that in Section 1.2.

Plant: $G(S) = \frac{1}{(s+1)(s+2)}$

Sampling Period: T = 1.0

Block Diagram:

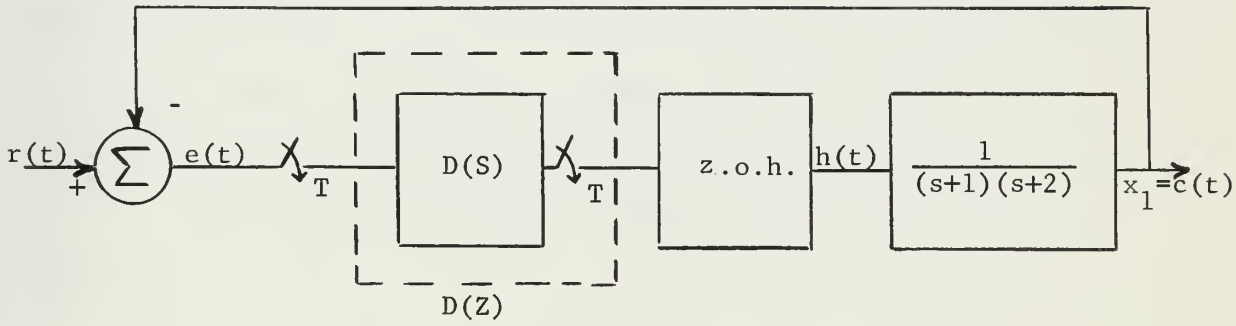


Figure 24

State Transition Flow Graph:

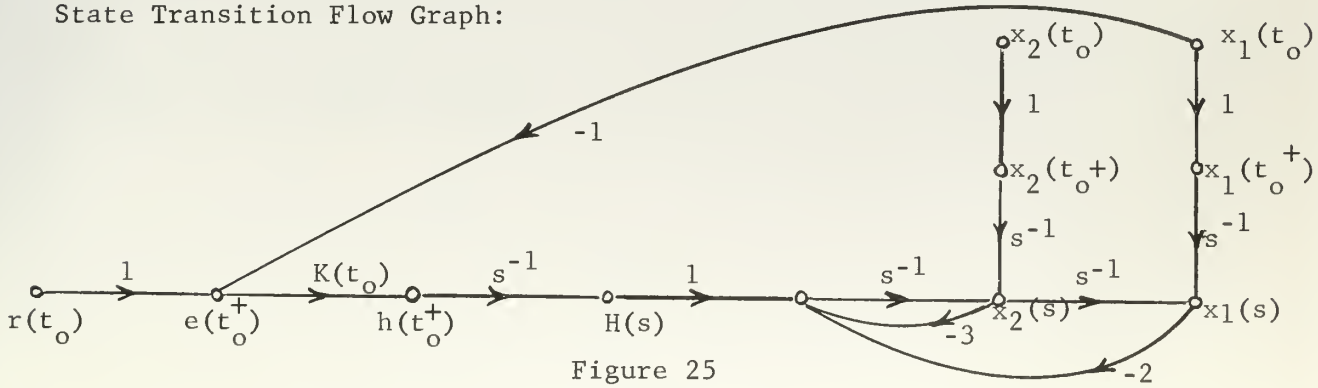


Figure 25

$$x_1\left[(k+1)T\right] = \left[2e^{-T}-e^{-2T}-K_k(.5-e^{-T}+.5e^{-2T})\right] x_1(kT)+(e^{-T}-e^{-2T})x_2(kT)+$$

$$K_k(.5-e^{-T}+.5e^{-2T})\ r(kT)$$

$$x_2\left[(k+1)T\right] = \left[2e^{-2T}-2e^{-T}-K_k(e^{-T}-e^{-2T})\right]x_1(kT)+(2e^{-T}-e^{-T})x_2(kT)+$$

$$K_k(e^{-T}-e^{-2T})\ r(kT)$$

For T = 1.0

$$x_1\left[(k+1)T\right] = \left[.601-.1995K_k\right] x_1(kT) + .233x_2(kT) + .1995K_k\ r(kT)$$

$$x_2\left[(k+1)T\right] = \left[-.466 - .233K_k\right] x_1(kT) - .098x_2(kT) + .233K_k\ r(kT)$$

(I) Let $r(kT) = 1.0$

$k=0$; $x_1(0) = x_2(0) = 0$

$x_1(T) = .1995K_o$; $x_2(T) = .233K_o$

These two equations clearly cannot be solved for the final conditions of: $x_1(T) = 1.0$; $x_2(T) = 0.0$

Therefore let $k = 1$

$$x_1(2T) = 0.1742K_o - 0.0398K_oK_1 + 0.1995K_1 \triangleq 1.0$$

$$x_2(2T) = -0.11575K_o - 0.0465K_oK_1 + 0.233K_1 \triangleq 0.0$$

from which:

$K_o = 3.65$; $K_1 = 6.73$

and

$h(0) = 3.65$; $h(T) = 1.832$

Note: Since the plant involved is of type zero it will have an inherent steady-state error to a step input. To eliminate this error a "steady state h" will have to be put into the system after the error has been initially reduced to zero. To determine this "steady state h" the final value theorem was used as follows:

Assume: $h(t) = at + b$

$$H(S) = \frac{a}{s^2} + \frac{b}{s}$$

$$x_1(s) = \left[\frac{bs+a}{s^2}\right]\left[\frac{1}{(s+1)(s+2)}\right]$$

$$\therefore \quad x_1(\infty) \quad \frac{d}{ds}\left[\frac{(bs+a)e^{st}}{s^2+3s+2}\right]\Bigg| \qquad s = 0$$

$$x_1(\infty) = \frac{2b+2at-3a}{4} \quad \triangleq \quad 1.0$$

or $a = 0 \; ; \; b = 2$

$$\therefore \; h(nT) = 2 \qquad n = 2,3....$$

(II) Now let $r(kT)=k$ i.e. unit ramp

In this case the final values desired are: $x_1(nT)=k$; $x_2(nT)=1.0$ for n the smallest possible integer.

For: $k = 0$; $x_1(0)=x_2(0)=0$

$x_1(T)=0 \; ; \; x_2(T) = 0$

Therefore let $k=1$

$x_1(2T)= 0.1995K_1$

$x_2(2T)= 0.233K_1$

These two equations cannot be solved so we must proceed to k=2.

$$x_1(3T) = 0.17425K_1 - 0.0398K_1K_2 + 0.399K_2 \triangleq 3.0$$

$$x_2(3T) = -0.11575K_1 - 0.0465K_1K_2 + 0.466K_2 \triangleq 1.0$$

From which

$$K_1 = 7.85 \; ; \; K_2 = 18.71$$

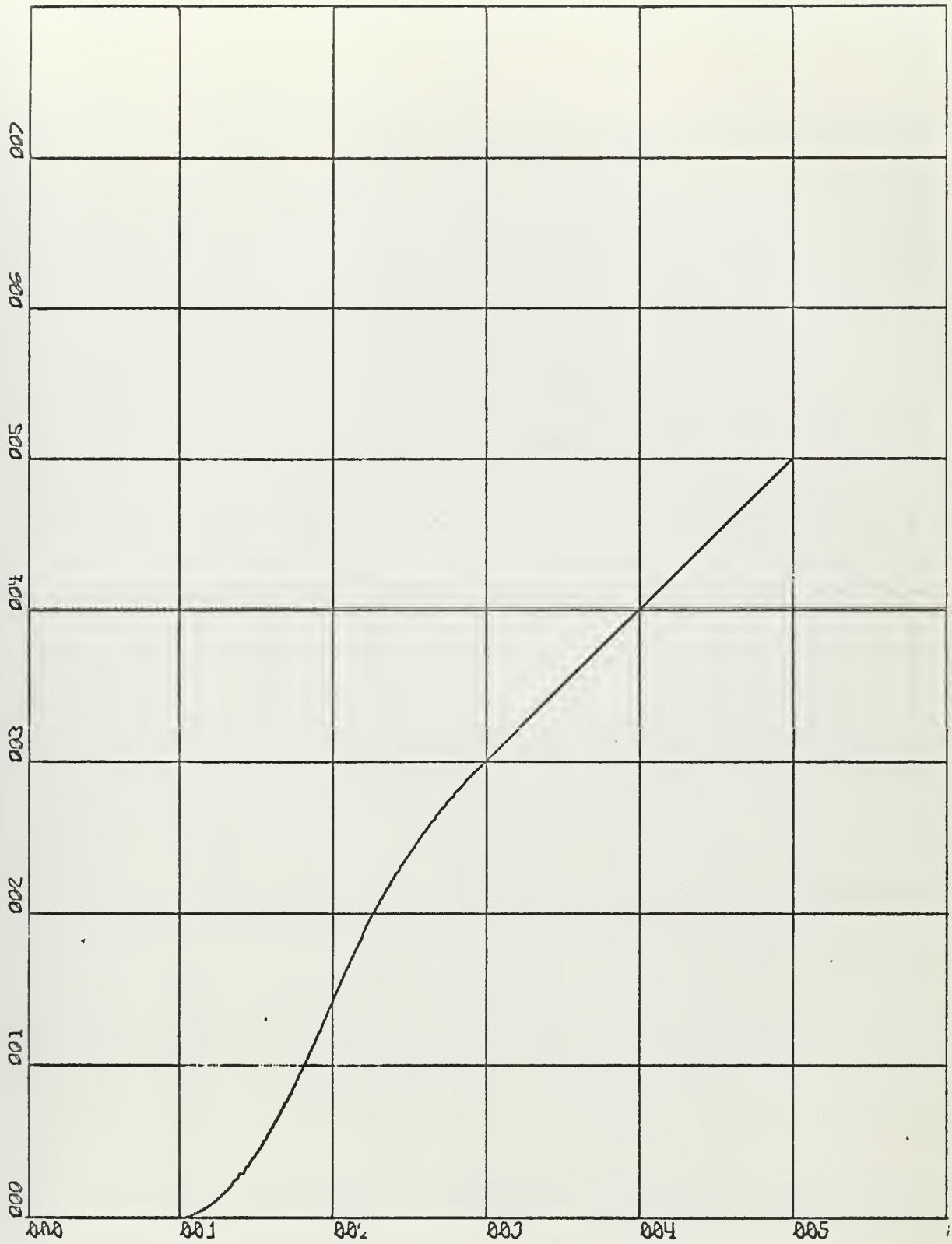
and

$$h(T) = 7.85 \; ; \; h(2T) = 8.14$$

Again to keep the output equal to the input for this type zero system we must output a "steady state h". From the final value theorem this is:

$$h(nT) = 2n+3 \quad n \; 3,4,5,....$$

The following pages contain the graphical simulation results for the type one system ($G(S) = \frac{1}{s(s+1)}$) to a unit ramp input and the responses of a type zero system ($G(S) = \frac{1}{(s+1)(s+2)}$) to both unit step and unit ramp inputs.

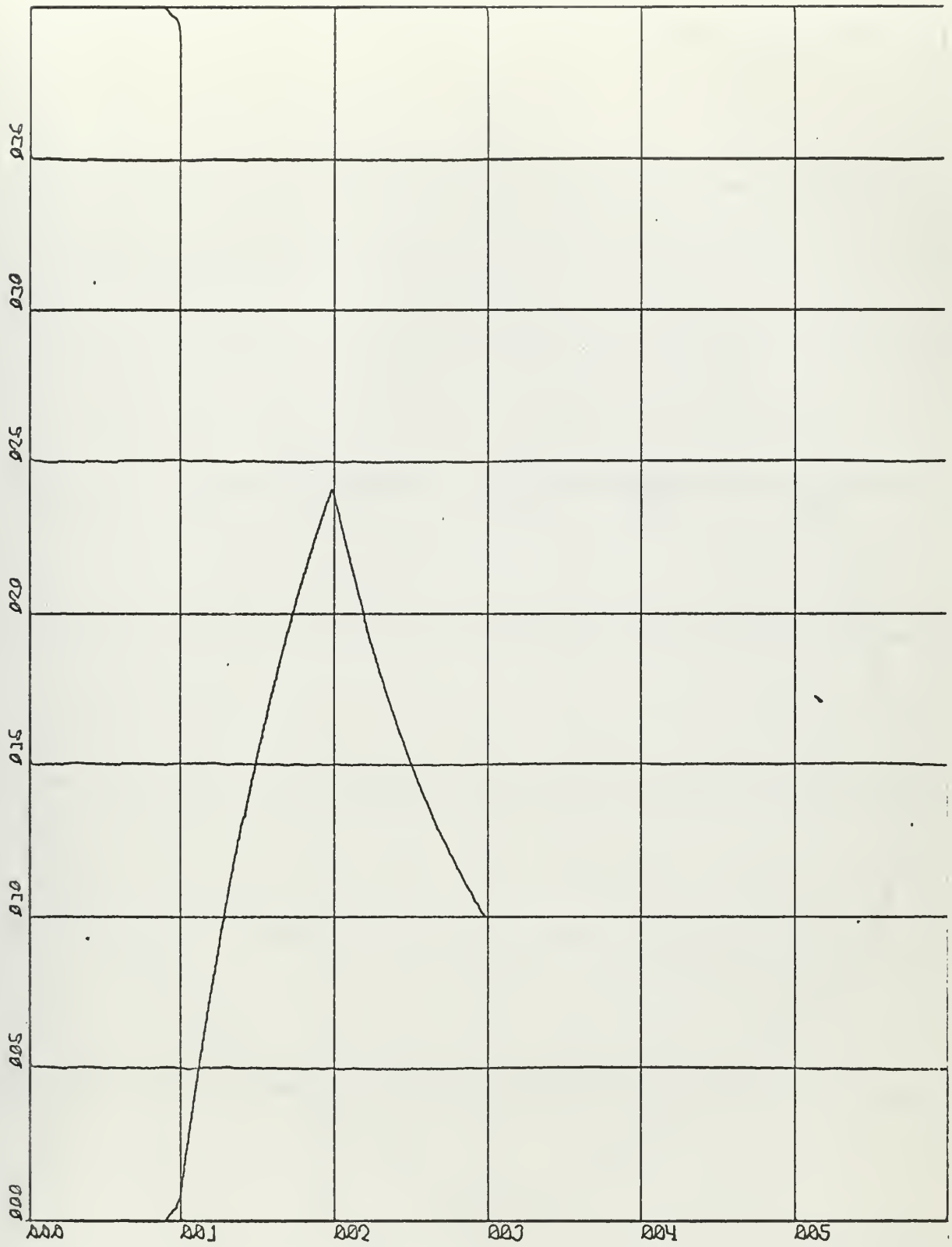


X-SCALE = 1.00E+00 UNITS/INCH.
Y-SCALE = 1.00E+00 UNITS/INCH.

BROWNE DIGIT 1R $G(S) = \frac{1}{s(s+1)}$ $r\ (kT) = k$

RUN 1 OUTPUT VS. TIME

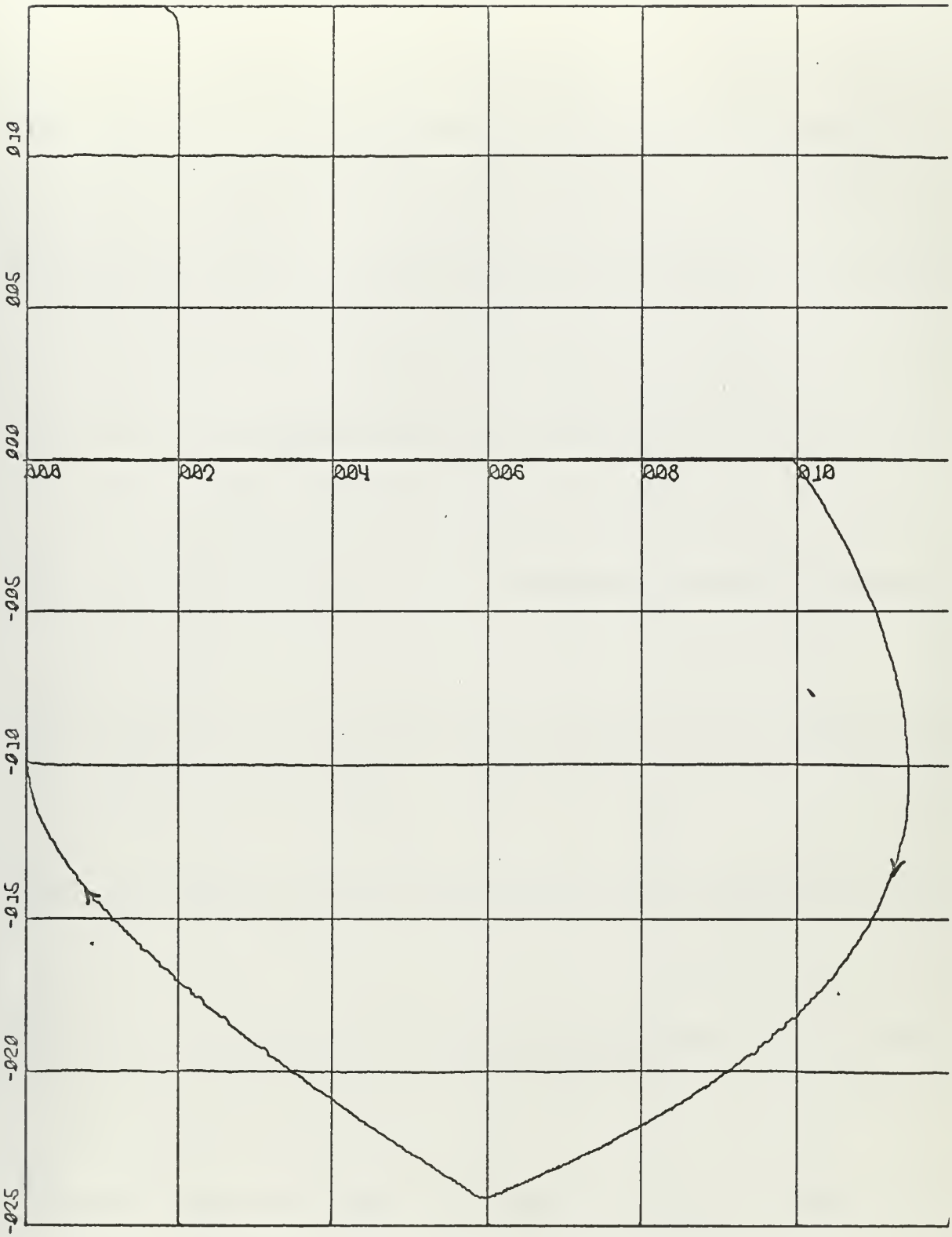
Figure 26



X-SCALE = 1.00E+00 UNITS/INCH.
Y-SCALE = 5.00E-01 UNITS/INCH.

BROWNE DIGIT 1R
RUN 1 VELOCITY VS. TIME

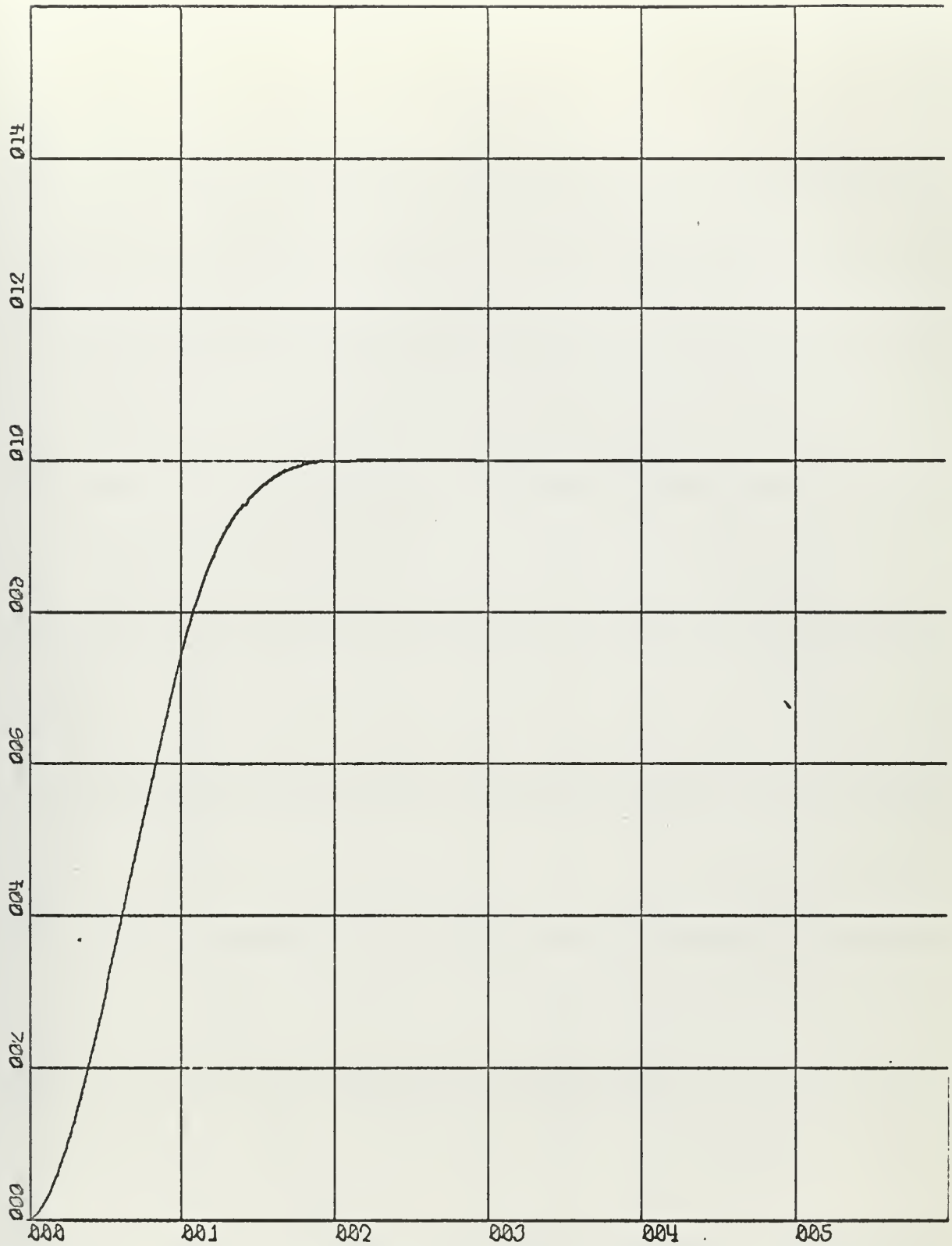
Figure 27



X-SCALE = 2.00E-01 UNITS/INCH.
Y-SCALE = 5.00E-01 UNITS/INCH.

BROWNE DIGIT 1R
RUN 1 EDOT VS. ERROR

Figure 28

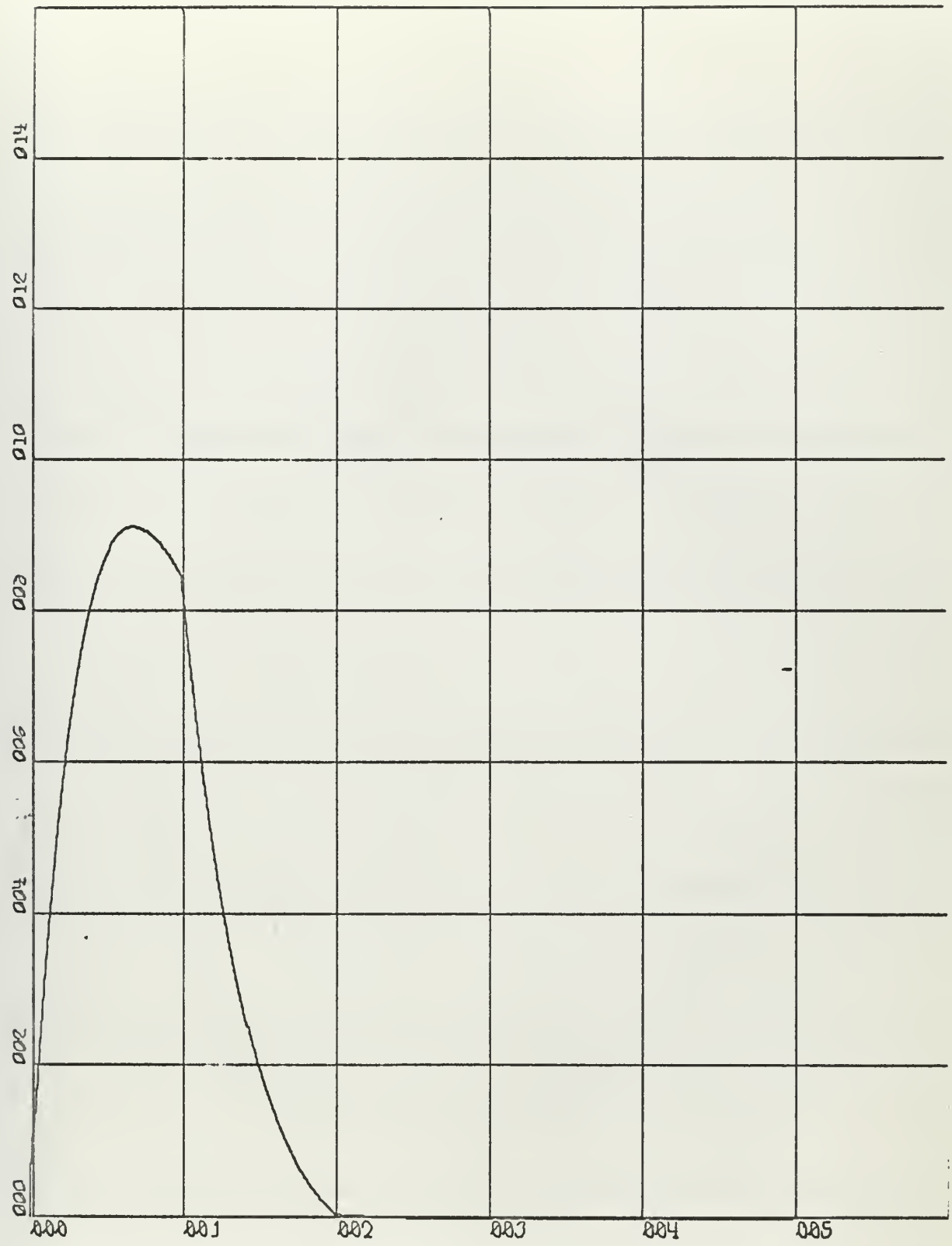


X-SCALE = 1.00E+00 UNITS/INCH.
Y-SCALE = 2.00E-01 UNITS/INCH.

BROWNE DIGIT 2 $G(S) = \frac{1}{(s+1)(s+2)}$ $r(kT) = 1.0$

RUN 1 OUTPUT VS. TIME

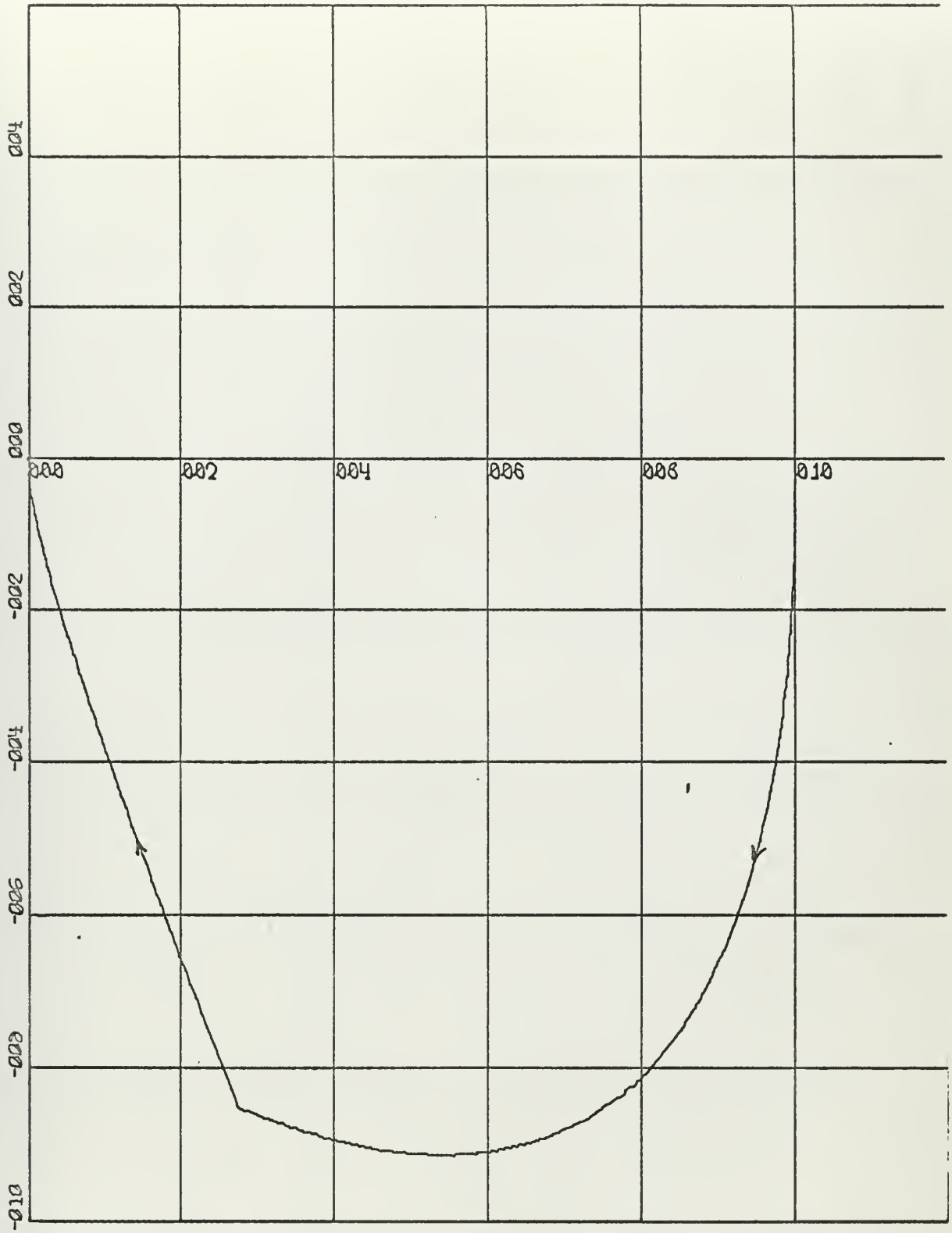
Figure 29



X-SCALE = 1.00E+00 UNITS/INCH.
Y-SCALE = 2.00E-01 UNITS/INCH.

BROWNE DIGIT 2
RUN 1 VELOCITY VS. TIME

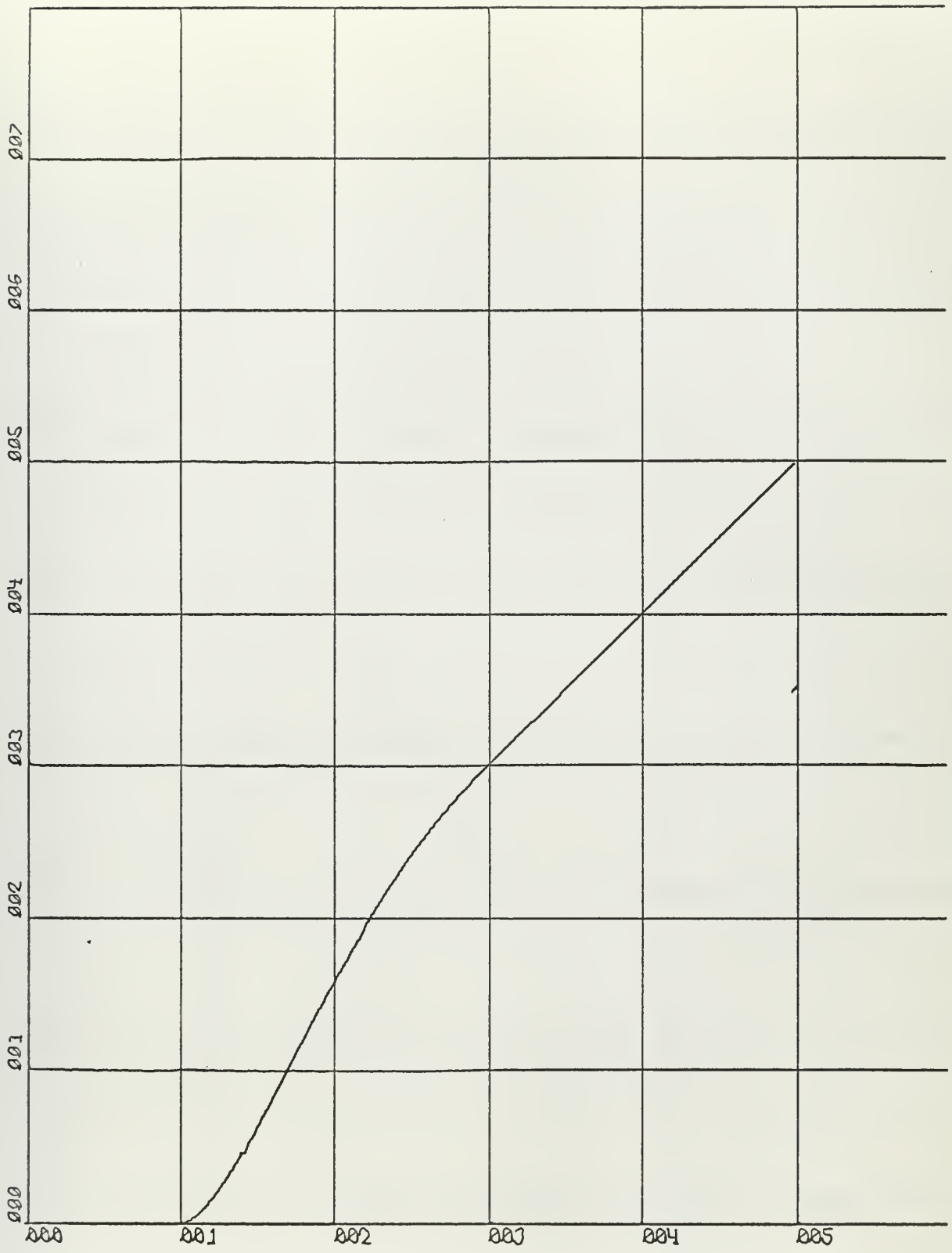
Figure 30



X-SCALE = 2.00E-01 UNITS/INCH.
Y-SCALE = 2.00E-01 UNITS/INCH.

BROWNE DIGIT 2
RUN 1 EDOT VS. ERROR

Figure 31

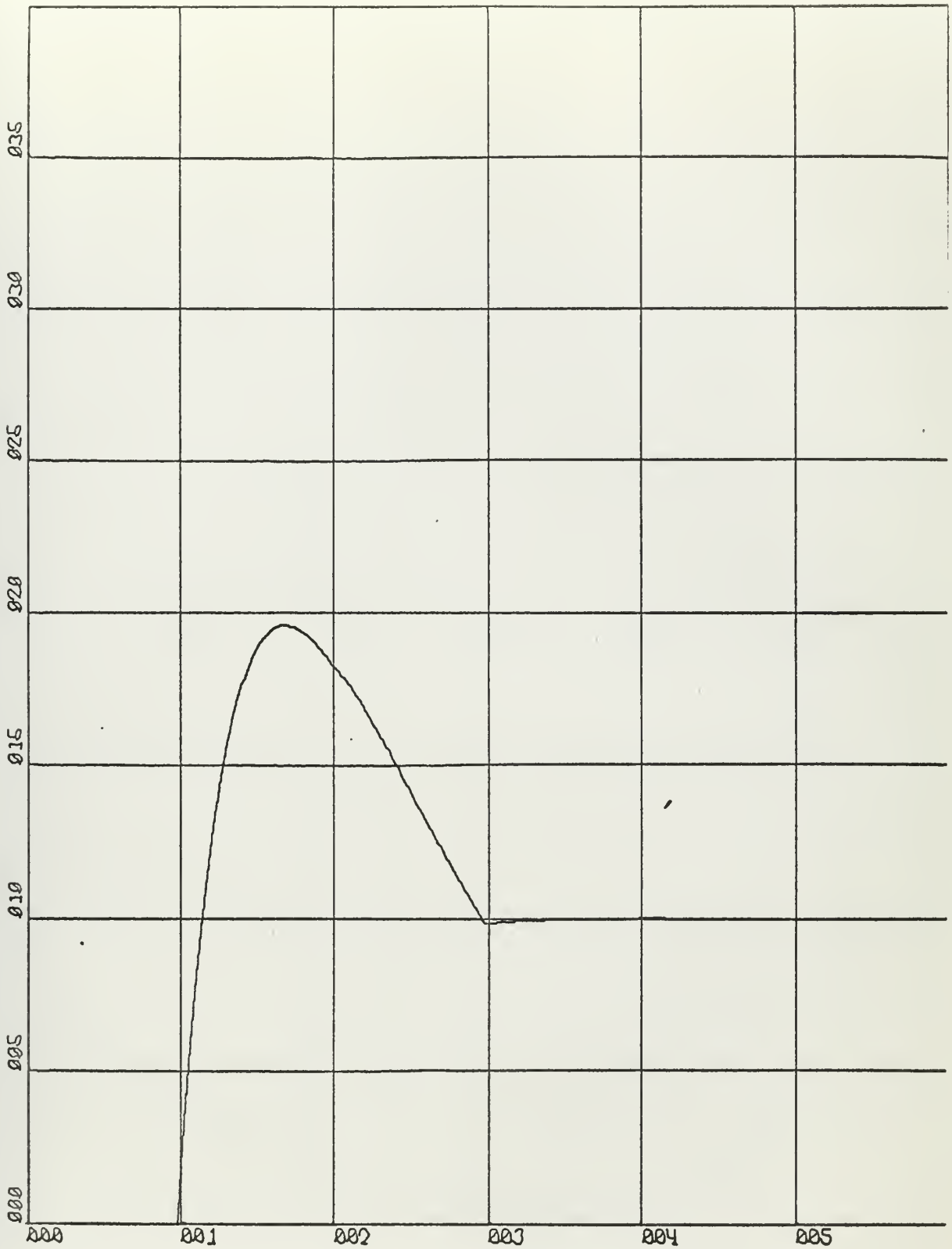


X-SCALE = 1.00E+00 UNITS/INCH.
Y-SCALE = 1.00E+00 UNITS/INCH.

BROWNE DIGIT 2R $G(S) = \frac{1}{(s+1)(s+2)}$ $r(kT) = k$

RUN 1 OUTPUT VS. TIME

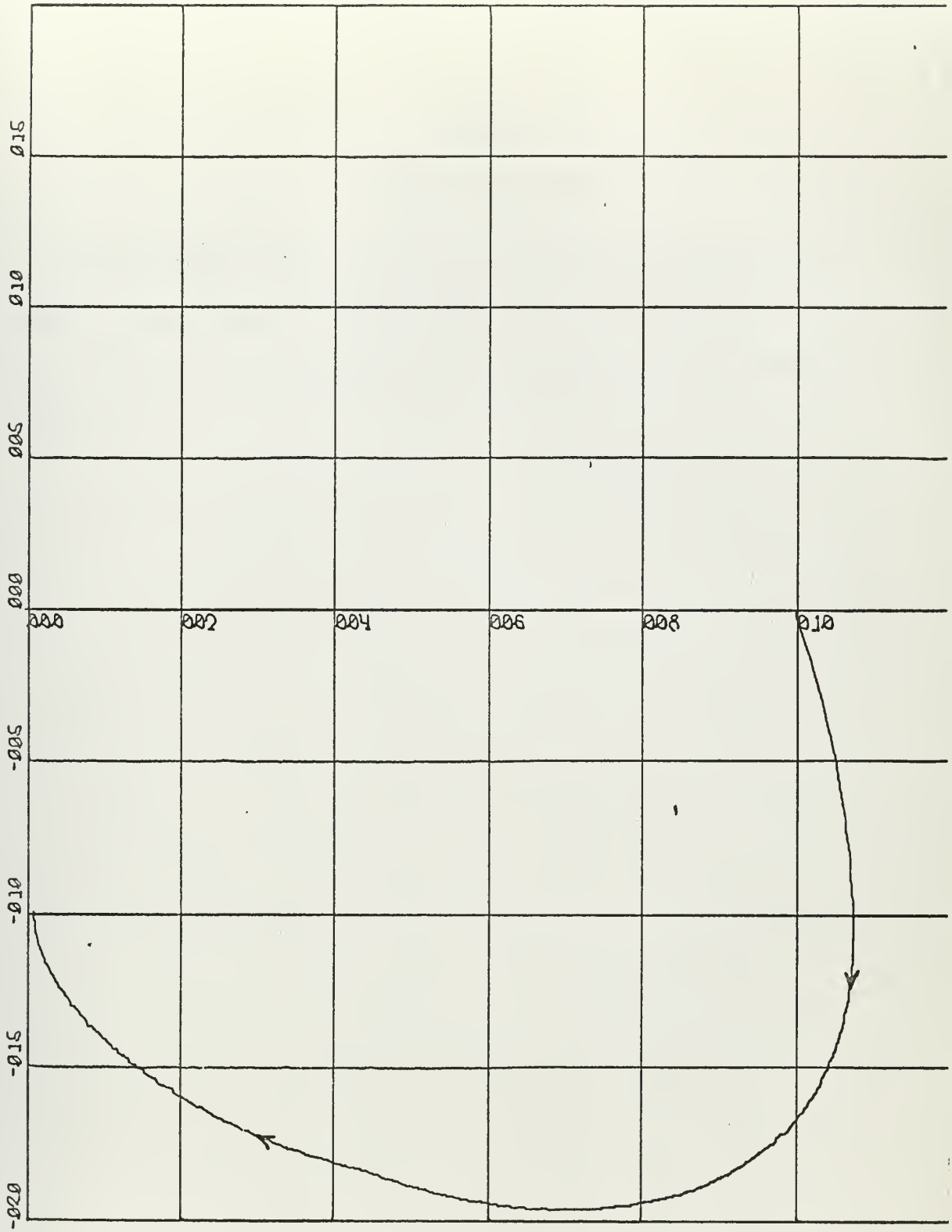
Figure 32



X-SCALE = 1.00E+00 UNITS/INCH.
Y-SCALE = 5.00E-01 UNITS/INCH.

BROWNE DIGIT 2R
RUN 1 VELOCITY VS. TIME

Figure 33



X-SCALE = 2.00E-01 UNITS/INCH.
Y-SCALE = 5.00E-01 UNITS/INCH.

BROWNE DIGIT 2R
RUN 1 EDOT VS. ERROR

Figure 34

APPENDIX II

SIMULATION PROGRAM

The following pages contain the simulation program used in verifying the digital controller design for a "deadbeat response" to a unit step input. The programs for simulating a unit ramp response are similar.

```
..JOB127F, BROWNE,E.R.
      PROGRAM DIGIT 1
      DIMENSION X(30),XDOT(30),C(15)
      C(10)=1.0
    1 CALL INTEG1 (T,X,XDOT,C)
      INPUT=1.0
      IF (T-1.*C(1)) 10,11,12
   10 HOLD= C(2)
      GO TO 14
   11 HOLD = C(3)
      GO TO 14
   12 IF (T-2.*C(1)) 11,13,13
   13 HOLD = C(4)
   14 XDOT(2)=HOLD-X(2)
      XDOT(1)=X(2)
      ERROR=1.0-X(1)
      EDOT=-X(2)
      X(3)=ERROR
      X(4)=EDOT
      C(11)=50.0
      GO TO 1
      END
      END
```

```
                                        BROWNE DIGIT 1
                                        ONE RUN IS CALLED FOR

INPUT DATA RECORD

ORDER OF EQUATIONS =  2
INITIAL TIME       =  .0000E+00
FINAL TIME         =  .5000E+00
STEP SIZE          =  .2000E-03

THE NON-ZERO CONSTANTS, C(I), ARE
              C( 1) =  .1000E+00
              C( 2) =  .1056E+03
              C( 3) = -.9580E+02

ALL THE INITIAL CONDITIONS ARE ZERO


THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

          TIME      X( 0)
        OUTPUT      X( 1)
      VELOCITY      X( 2)
         ERROR      X( 3)


THE INDIVIDUAL GRAPH TITLES AND THE CORRESPONDING VARIABLES ARE

      OUTPUT VS. TIME      X( 1) VS. X( 0)
      VELOCITY VS.TIME     X( 2) VS. X( 0)
       EDOT VS. ERROR      X( 4) VS. X( 3)
```

BROWNE DIGIT 1

TIME	OUTPUT	VELOCITY	ERROR
.00000E+00	.00000E+00	.00000E+00	.10000E+01
.10000E-01	.52624E-02	.10507E+01	.99474E+00
.20000E-01	.20980E-01	.20910E+01	.97902E+00
.30000E-01	.47048E-01	.31210E+01	.95295E+00
.40000E-01	.83365E-01	.41406E+01	.91664E+00
.50000E-01	.12983E+00	.51502E+01	.87017E+00
.60000E-01	.18633E+00	.61497E+01	.81367E+00
.70000E-01	.25279E+00	.71392E+01	.74721E+00
.80000E-01	.32909E+00	.81189E+01	.67091E+00
.90000E-01	.41513E+00	.90889E+01	.58487E+00
.10000E+00	.51083E+00	.10042E+02	.48917E+00
.11000E+00	.60598E+00	.89893E+01	.39402E+00
.12000E+00	.69065E+00	.79466E+01	.30935E+00
.13000E+00	.76495E+00	.69143E+01	.23505E+00
.14000E+00	.82897E+00	.58923E+01	.17103E+00
.15000E+00	.88283E+00	.48805E+01	.11717E+00
.16000E+00	.92662E+00	.38787E+01	.73384E-01
.17000E+00	.96044E+00	.28869E+01	.39565E-01
.18000E+00	.98439E+00	.19049E+01	.15614E-01
.19000E+00	.99857E+00	.93272E+00	.14340E-02
.20000E+00	.10031E+01	-.29786E-01	-.30727E-02
.21000E+00	.10027E+01	-.32651E-01	-.27445E-02
.22000E+00	.10024E+01	-.32326E-01	-.24196E-02
.23000E+00	.10021E+01	-.32005E-01	-.20980E-02
.24000E+00	.10018E+01	-.31686E-01	-.17795E-02
.25000E+00	.10015E+01	-.31371E-01	-.14643E-02
.26000E+00	.10012E+01	-.31059E-01	-.11521E-02
.27000E+00	.10008E+01	-.30750E-01	-.84307E-03
.28000E+00	.10005E+01	-.30444E-01	-.53710E-03
.29000E+00	.10002E+01	-.30141E-01	-.23418E-03
.30000E+00	.99993E+00	-.29841E-01	.65725E-04
.31000E+00	.99964E+00	-.29544E-01	.36265E-03
.32000E+00	.99934E+00	-.29250E-01	.65662E-03
.33000E+00	.99905E+00	-.28959E-01	.94766E-03
.34000E+00	.99876E+00	-.28671E-01	.12358E-02
.35000E+00	.99848E+00	-.28386E-01	.15211E-02
.36000E+00	.99820E+00	-.28103E-01	.18035E-02
.37000E+00	.99792E+00	-.27824E-01	.20832E-02
.38000E+00	.99764E+00	-.27547E-01	.23600E-02
.39000E+00	.99737E+00	-.27273E-01	.26341E-02
.40000E+00	.99709E+00	-.27001E-01	.29055E-02
.41000E+00	.99683E+00	-.26733E-01	.31741E-02
.42000E+00	.99656E+00	-.26467E-01	.34401E-02
.43000E+00	.99630E+00	-.26203E-01	.37035E-02
.44000E+00	.99604E+00	-.25942E-01	.39642E-02
.45000E+00	.99578E+00	-.25684E-01	.42223E-02
.46000E+00	.99552E+00	-.25429E-01	.44779E-02
.47000E+00	.99527E+00	-.25176E-01	.47309E-02
.48000E+00	.99502E+00	-.24925E-01	.49814E-02
.49000E+00	.99477E+00	-.24677E-01	.52294E-02

BROWNE DIGIT 1

TIME	OUTPUT	VELOCITY	ERROR
.50000E+00	.99453E+00	-.24432E-01	.54750E-02

NORMAL STOP AT FINAL TIME

GRAPH TITLED . . BROWNE DIGIT 1
RUN 1
OUTPUT VS. TIME

GRAPH TITLED . . BROWNE DIGIT 1
RUN 1
VELOCITY VS.TIME

GRAPH TITLED . . BROWNE DIGIT 1
RUN 1
EDOT VS. ERROR

THE ONE RUN CALLED FOR HAS BEEN COMPLETED.

STOP
TIME, 1 MINUTES AND 14 SECONDS

APPENDIX III

CONTINUOUS PLANT

The continuous plant which was to be controlled in a sampled-data manner consisted of two operational a.c. amplifiers, a d.c. power amplifier, and a d.c. torque motor manufactured by the Inland Motor Corporation of Pearl River, New York.

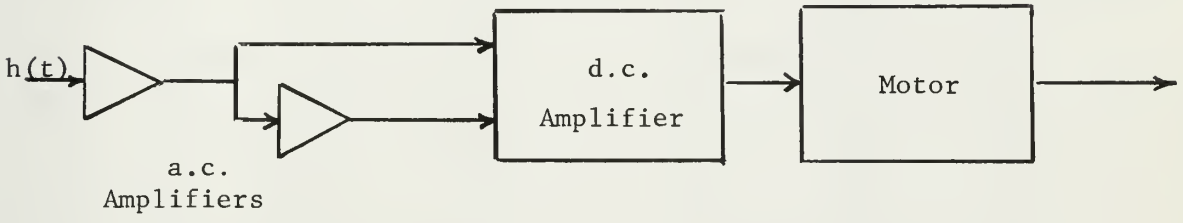
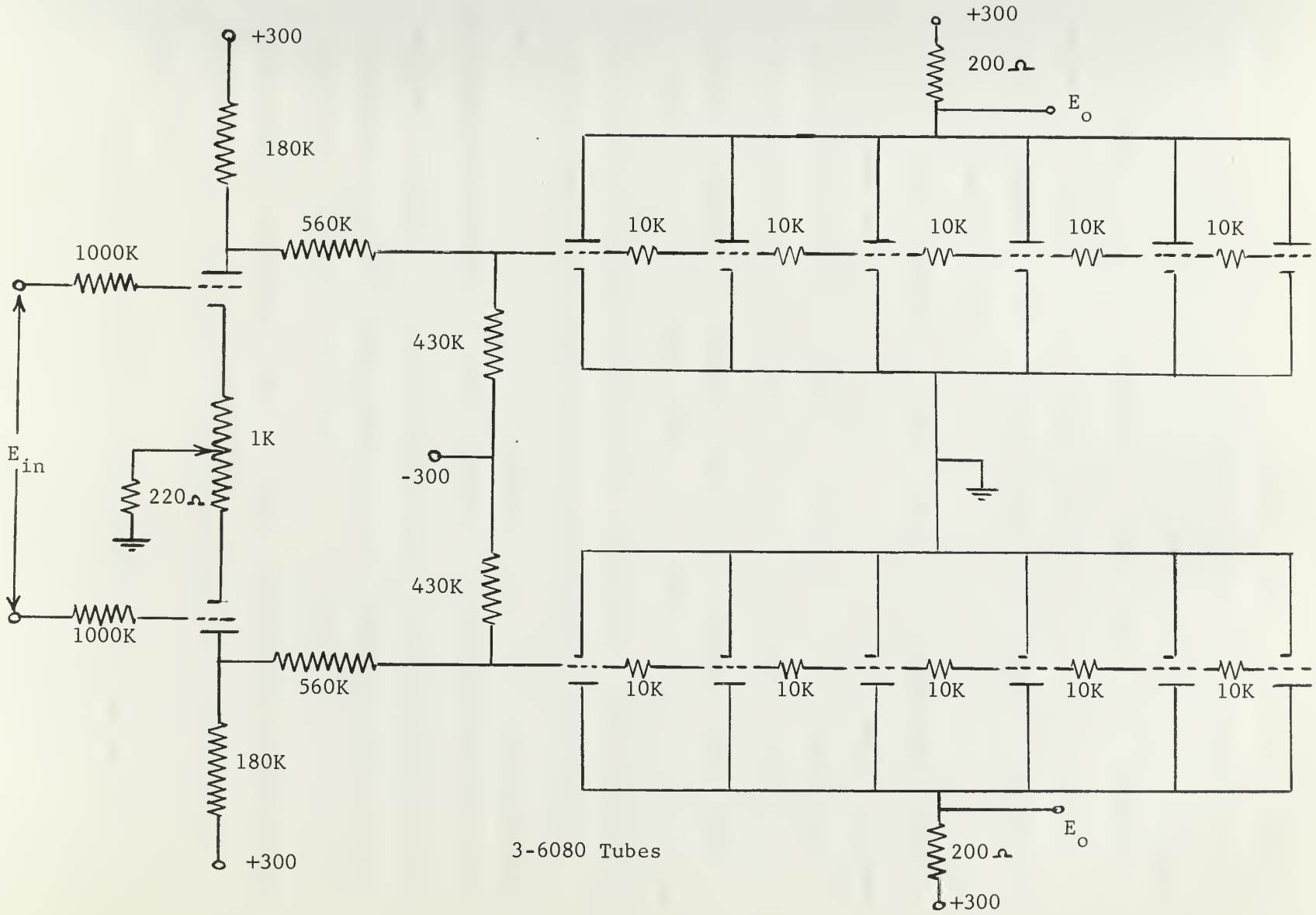


Figure 35

As is seen in the figure, the control signal to the plant was received from the zero-order hold and fed into the first operational amplifier. The second operational amplifier transmitted the signal to the d.c. power amplifier where the signal size was increased further. From the d.c. power amplifier the control signal was fed to the armature of the motor.

The Inland motor has a permanent magnet d.c. field and is armature controlled. An inertia disc was attached to the shaft to slow down the response of the system. Also on the motor shaft were a tachometer for velocity pick-off and a potentiometer which was used for position feedback. The motor was rated at 0.47 ampere and 90 volts. [5]



D.C. Power Amplifier

Figure 36

To provide the needed power to the motor an amplifier whose capabilities were greater than the common commercial type was needed. The d.c. amplifier used was the same one that had been constructed for use on a previous thesis using the same motor. [5] As seen in Figure 35 the input to the d.c. amplifier was double ended. The amplifier operation is push-pull which reduces third harmonic distortion and provides a balanced output. (See Figure 36 for a circuit diagram of the d.c. power amplifier.)

To insure that the d.c. power amplifier was operating correctly two tests were performed. The first test checked the frequency response characteristics of the a.c. and d.c. combination. The second test plotted the output vs. the input of the d.c. amplifier alone thus checking the saturation characteristic of the amplifier. Test data and curves are shown on pages 82-84. It should be mentioned that some difficulty was encountered with the d.c. amplifier. The amplifier needs a supply voltage of 300 volts with a corresponding current of one ampere. Since this power was not available directly in the room where the equipment was set up, it was decided to place four portable power supplies in parallel. This arrangement proved fairly satisfactory with the exception that the power supplied was not regulated. This caused difficulty in balancing the amplifiers and once they were balanced they did not remain so for any length of time. This balancing problem was bothersome but in no way invalidated the results or conclusions of this report.

To determine the transfer function of the amplifier motor

combination a standard frequency response test was conducted on the closed loop system. The input sinusoid was supplied by a Hewlet Packard Low-Frequency Generator Model 202A. Unity feedback was employed and a Mark II Brush Recorder was used to measure the input and output. (See Figure 37)

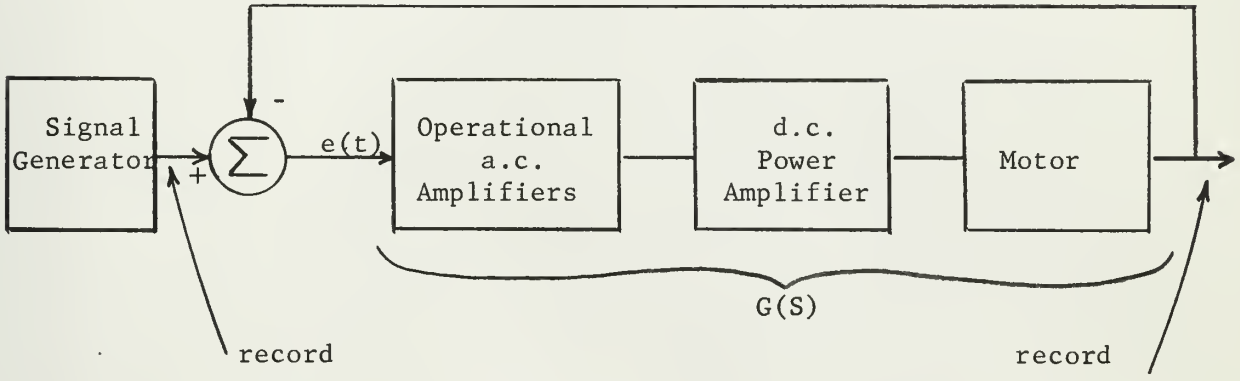


Figure 37

The recorded closed loop data was first plotted as a Bode diagram. (Magnitude and Phase) The closed loop data was then converted to an open loop transfer function by using a Nichols Chart and a second Bode plot. The open loop transfer function was determined to be: $G(S) = \frac{A}{s(s + 3)}$ with A set at 40. (See pages 85-89 for test data, Bode diagrams, and Nichols Chart.)

Frequency Response of Amplifiers in Cascade

frequency (cps)	ω (radians)	input (volts)	output (volts)	gain
0.1	0.628	5.4	55.0	10.2
0.2	1.255	5.4	55.0	10.2
0.3	1.885	5.3	55.0	10.4
0.4	2.510	5.3	55.0	10.4
0.5	3.140	5.2	55.0	10.6
0.6	3.770	5.2	55.0	10.6
0.7	4.390	5.2	55.0	10.6
0.8	5.020	5.2	55.0	10.6
0.9	5.650	5.2	55.0	10.6
1.0	6.280	5.2	55.0	10.6
1.1	6.900	5.2	55.0	10.6
1.2	7.550	5.2	55.0	10.6
2.0	12.550	5.2	55.0	10.6
4.0	25.100	5.2	55.0	10.6
6.0	37.700	5.1	55.0	10.8
8.0	50.200	5.1	55.0	10.8
10.0	62.800	5.0	54.0	10.8
12.0	75.500	5.0	53.0	10.6
15.0	94.500	5.0	53.0	10.6
20.0	125.500	5.0	52.0	10.4
40.0	251.000	4.6	48.0	10.4
50.0	314.000	4.6	42.0	9.2
70.0	439.000	4.2	34.0	8.6

Saturation Characteristic of d.c. Amplifier

frequency= 0.7 cps; ω = 4.39

input (volts)	output (volts)
0.3	1.0
1.3	4.4
3.1	10.0
5.6	17.5
7.6	24.0
10.0	33.0
13.0	43.0
15.5	50.0
18.0	58.0
20.0	64.0
22.0	71.0
25.0	77.0
27.0	82.5
30.0	85.0
32.0	90.0
34.5	95.0
38.0	100.0
42.0	102.5
46.0	105.0
50.0	107.5
53.0	110.0

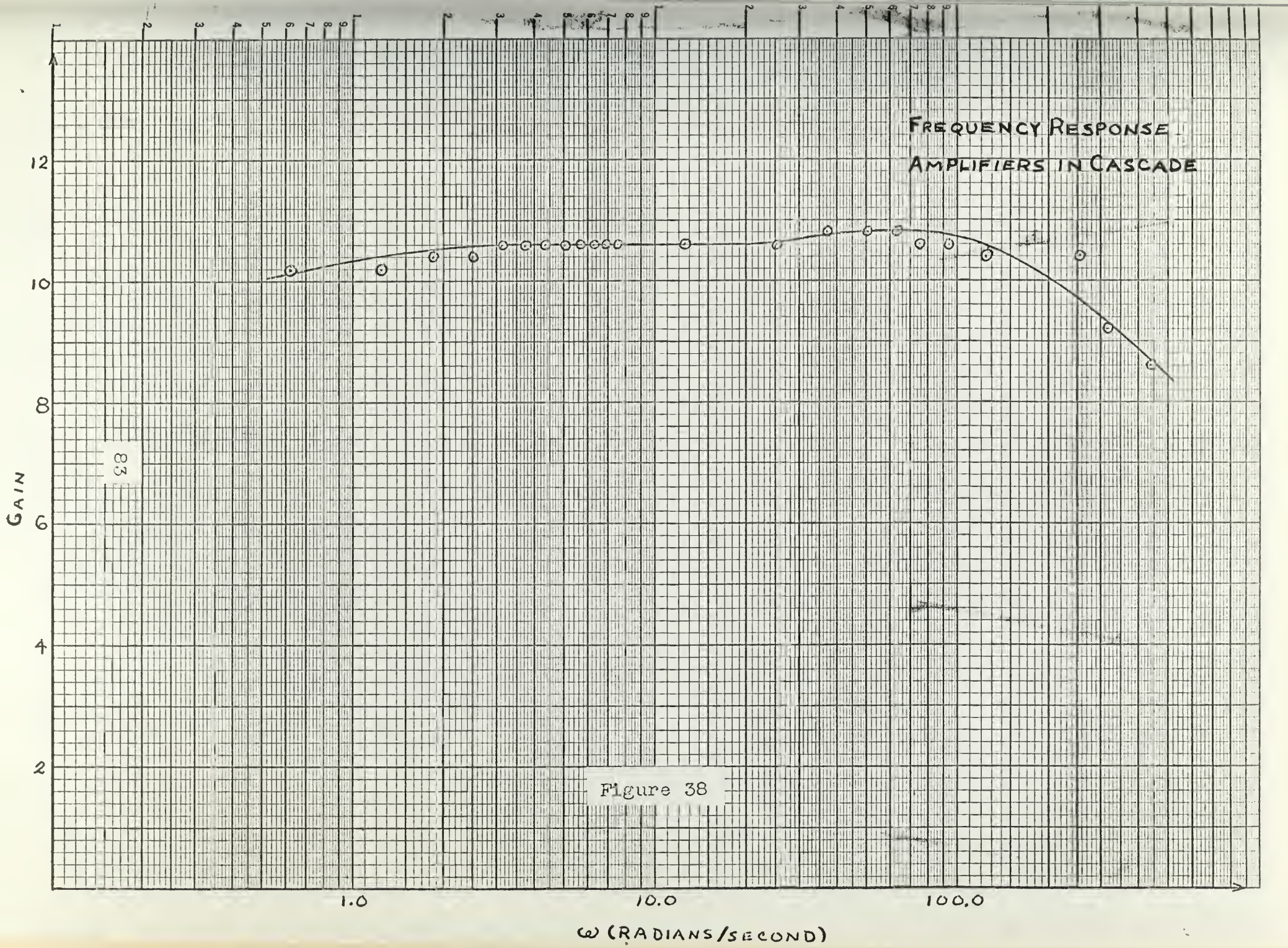

FREQUENCY RESPONSE
AMPLIFIERS IN CASCADE
GAIN
12
10
8
6
4
2
1.0
10.0
100.0
ω (RADIANS/SECOND)

Figure 38

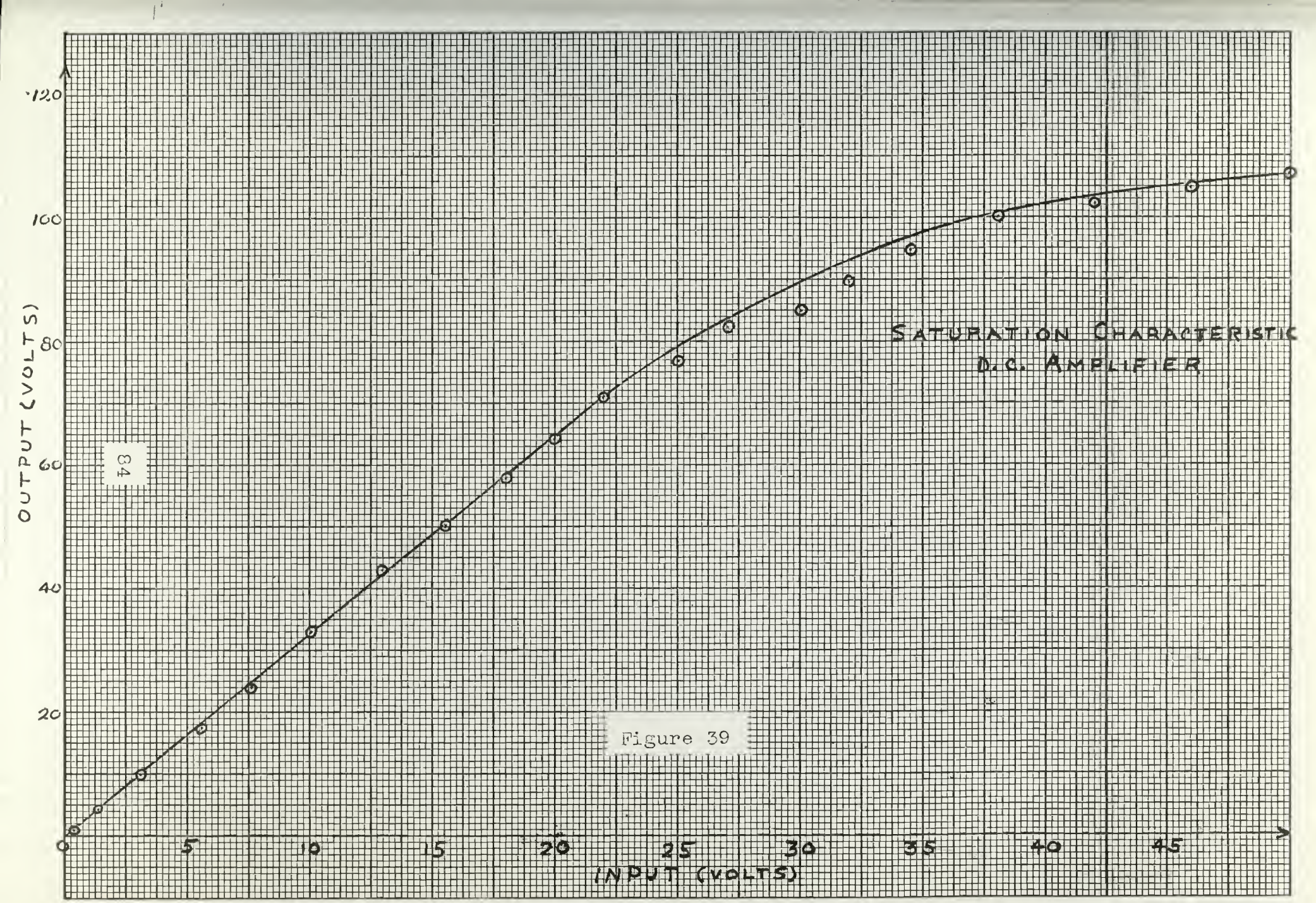

SATURATION CHARACTERISTIC
D.C. AMPLIFIER
OUTPUT (VOLTS)
INPUT (VOLTS)
120
100
80
60
40
20
0
5
10
15
20
25
30
35
40
45

Figure 39

Frequency Response Data

frequency (cps)	ω (radians)	magnitude (db.)	phase angle (degrees)
0.1	0.628	0.0	- 0
0.2	1.25	0.7	- 11
0.25	1.57	1.3	- 16
0.3	1.88	1.6	- 11
0.35	2.20	2.6	- 22
0.4	2.55	2.4	- 25
0.45	2.82	3.18	- 23
0.5	3.14	3.5	- 33
0.55	3.45	3.7	- 29
0.6	3.78	3.75	- 31
0.625	3.93	3.75	- 37
0.65	4.08	4.3	- 37
0.675	4.25	3.75	- 39
0.7	4.40	5.2	- 41
0.725	4.56	5.2	- 37
0.75	4.71	5.2	- 43
0.775	4.87	3.75	- 45
0.8	5.03	4.3	- 47
0.825	5.19	5.75	- 48
0.85	5.34	6.85	- 86
0.875	5.50	6.7	- 89
0.9	5.65	6.3	- 92
0.95	5.96	5.3	-102
1.0	6.28	4.25	-106
1.05	6.59	3.25	-112
1.1	6.91	2.6	-117
1.5	9.45	-2.9	-124

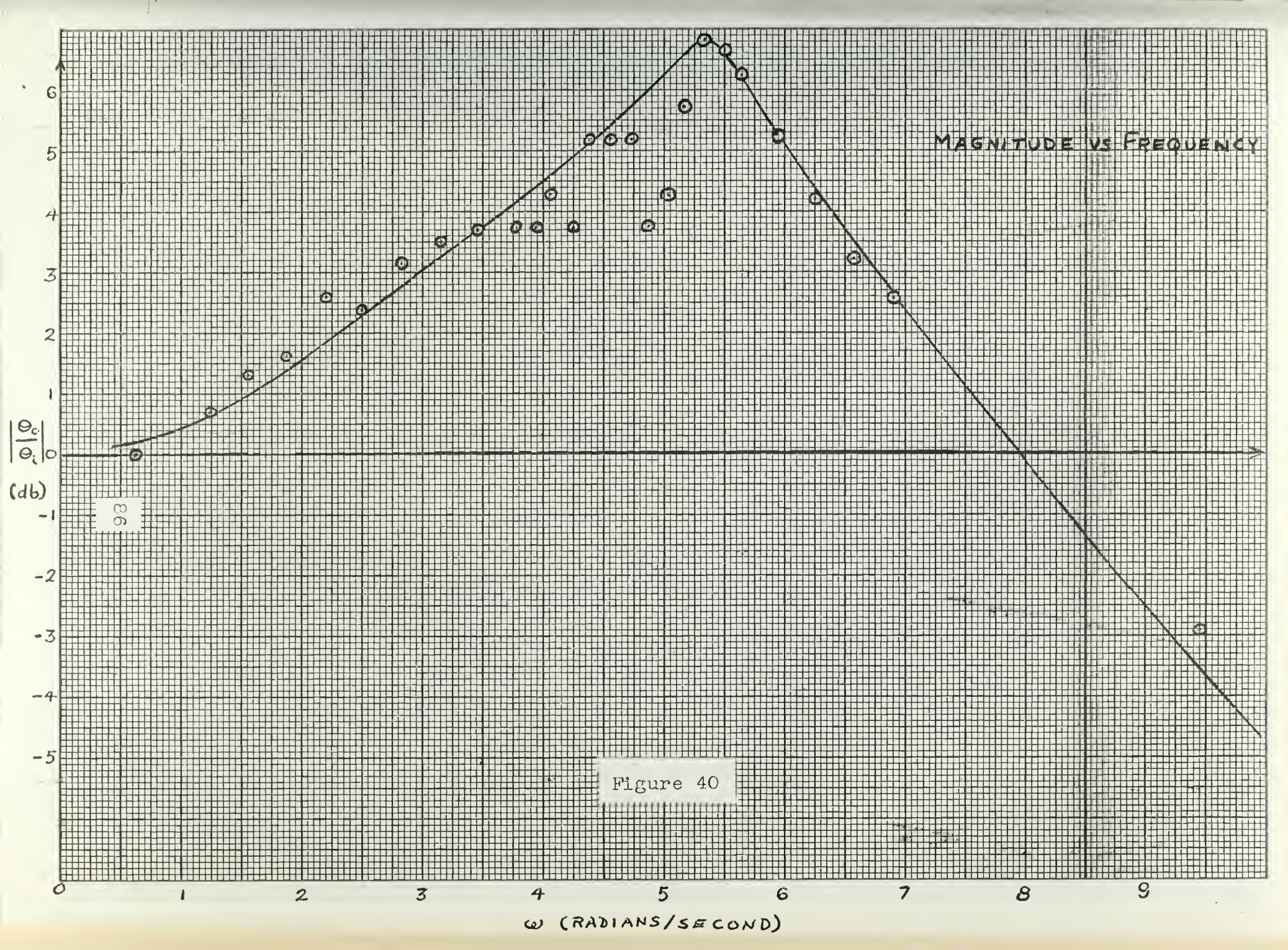

MAGNITUDE VS FREQUENCY
$\left|\frac{e_o}{e_i}\right|$ (db)
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
1
2
3
4
5
6
7
8
9
ω (RADIANS/SECOND)

Figure 40

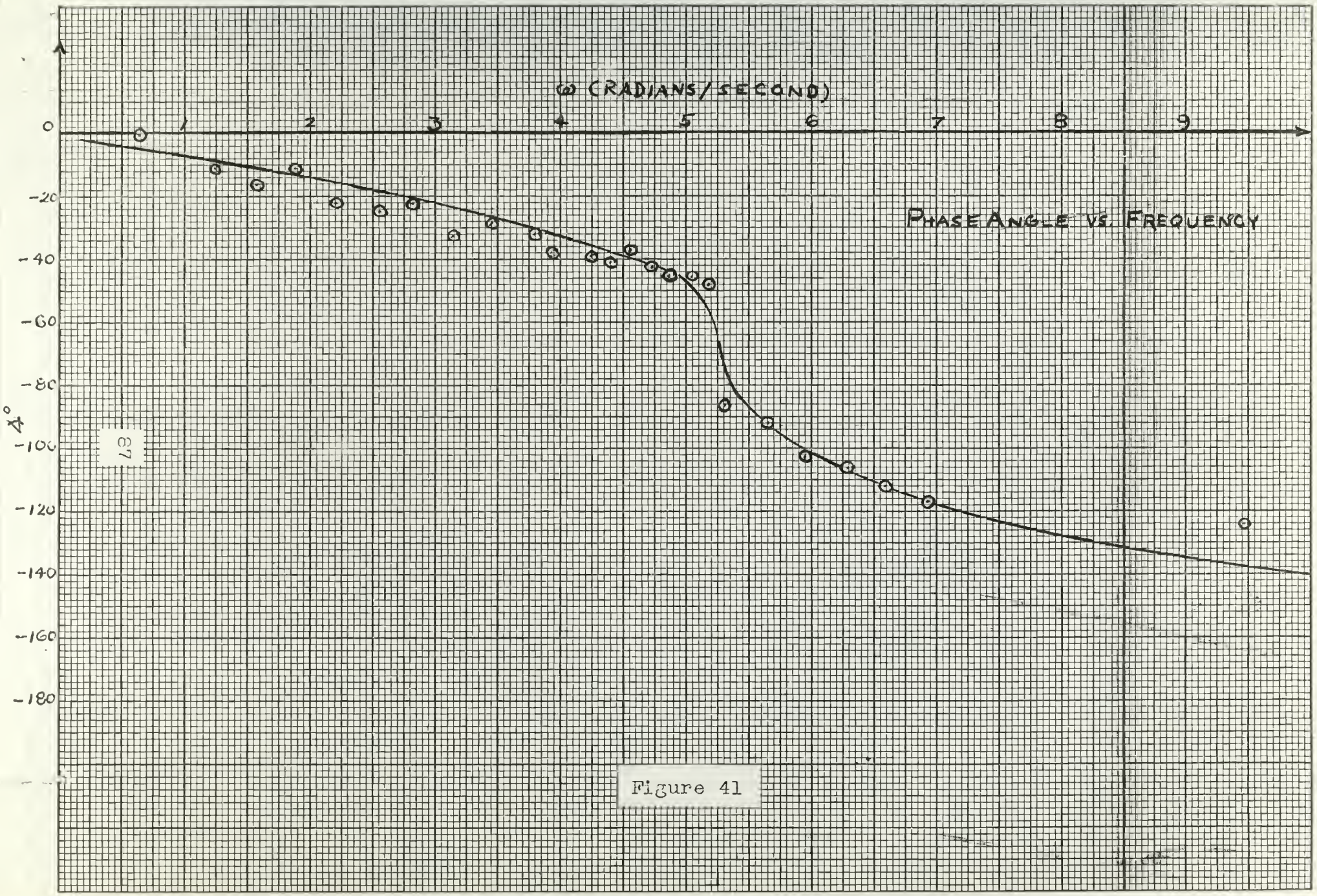

ω (RADIANS/SECOND)
0
1
2
3
4
5
6
7
8
9
-20
-40
-60
-80
-100
-120
-140
-160
-180
Δ°
PHASE ANGLE VS. FREQUENCY

Figure 41

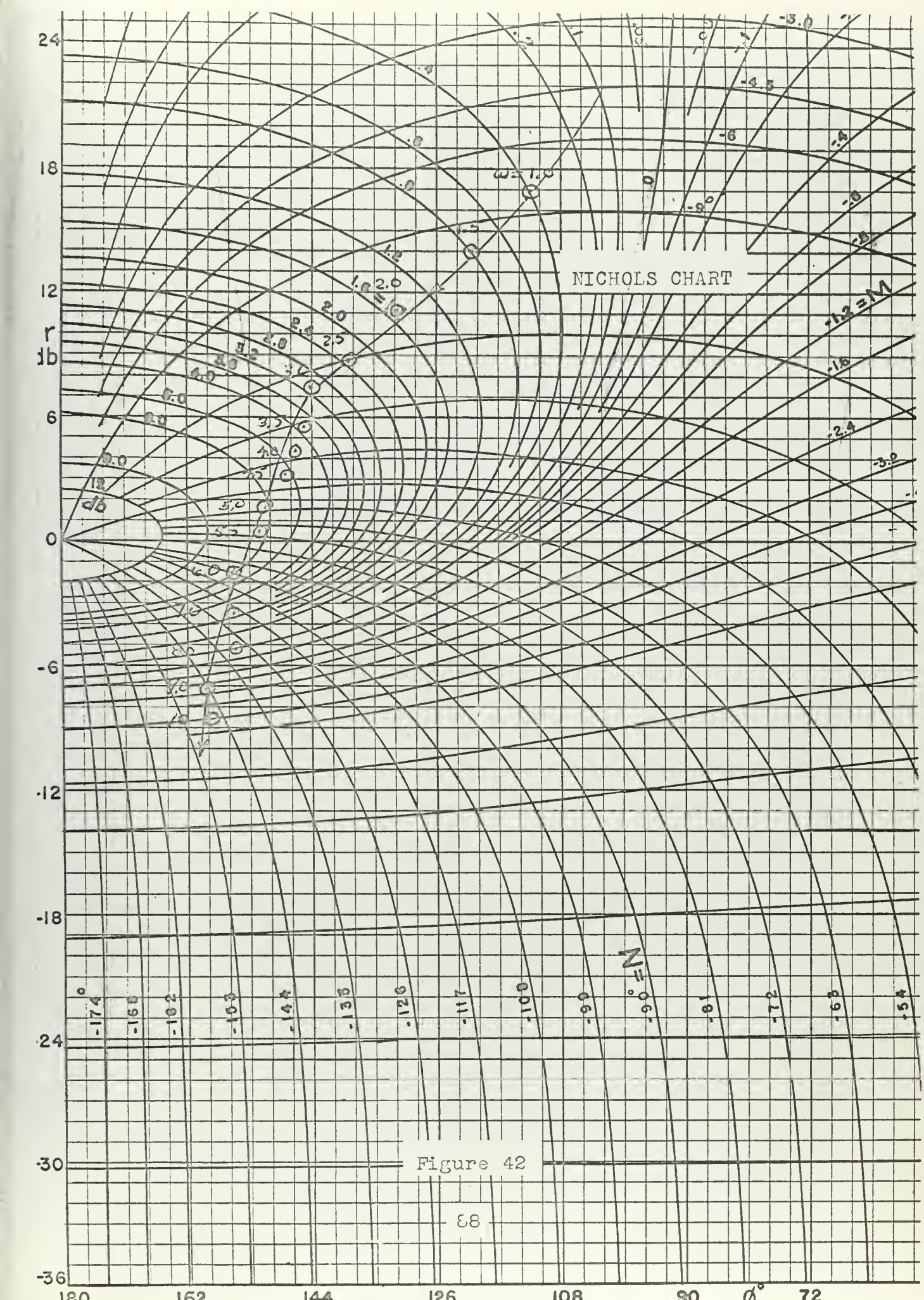

NICHOLS CHART
24
18
12
r
db
6
0
-6
-12
-18
-24
-30
-36
180
162
144
126
108
90
72
φ°
ω=1.0
M=1.2
M=.06
-174°
-168
-162
-153
-144
-138
-126
-117
-108
-99
-81
-72
-63
-54

Figure 42

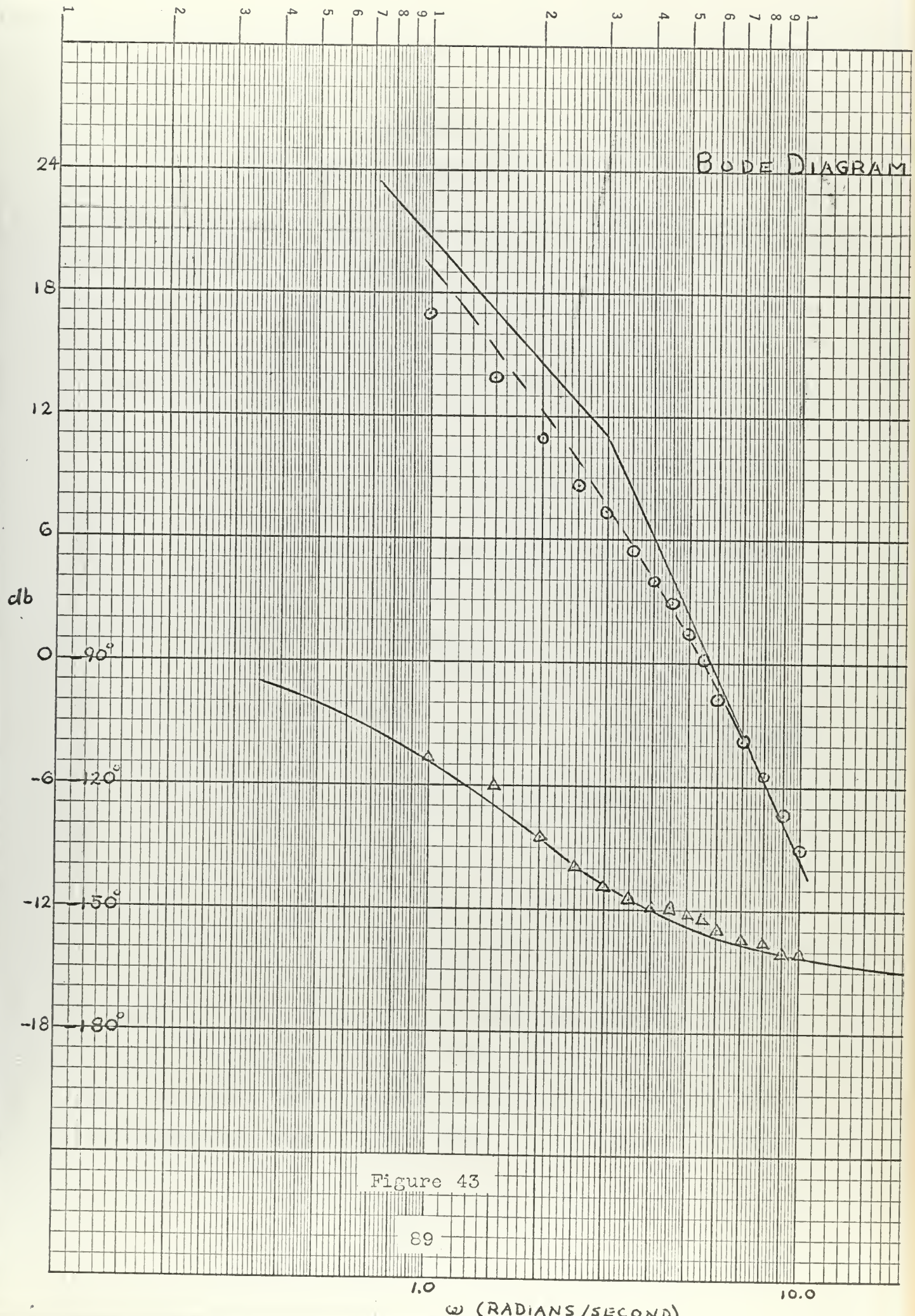

BODE DIAGRAM
24
18
12
6
db
0
-6
-12
-18
-90°
-120°
-150°
-180°
1.0
10.0
ω (RADIANS/SECOND)

Figure 43

APPENDIX IV

MEASUREMENT OF QUANTIZATION ERRORS

This appendix will cover the determination of quantization errors in the sampled-data control system. It will only deal with the real plant since the main thesis effort was directed to the control of this plant. The theory used was described by John B. Slaughter of the Navy Electronics Laboratory in the January 1964 IEEE Proceedings on Automatic Control. [6]

Quantization, or round off, errors occur because of the inevitable conversion operations required when analog and digital devices are connected together in a sampled-data closed loop system. Quantization is the process of converting a signal in analog form to its digital approximation. This is best illustrated by Figure 44.

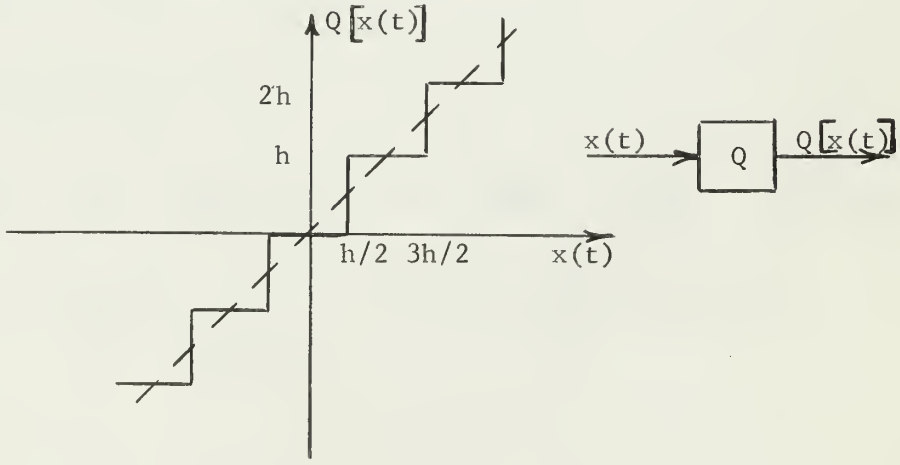


Figure 44

The dashed line in the figure is the desired linear response, while the staircase represents the actual output of the quantizer. The range of input magnitudes is divided into disjoint intervals (h_i) which are not necessarily equal. All magnitudes falling within an interval

are equated to a single value within the interval. This results in the digital approximation to the analog input. It can be seen that the maximum round off error of the quantization operation is h/2. (h is called the quantizing level.)

Quantization is non-linear in the sense that a discrete set of amplitude levels is produced for some continuous range of input. Because of this an analytical expression for the effects of this error is difficult to obtain. However, by assuming that the quantizing levels are kept small the concepts of probability may be used. The quantizer is treated as a summing point with uniformly distributed noise introduced at that point.

In continuing, we will refer to the real plant which was used in the sampled-data control problem.

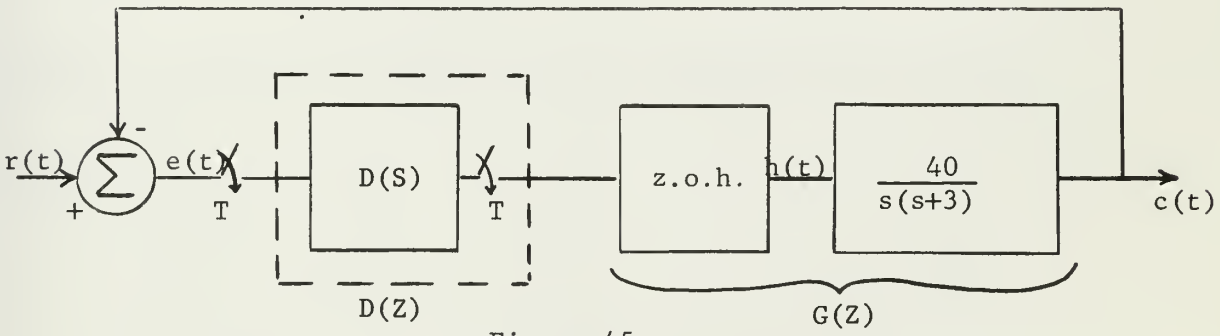


Figure 45

In Figure 45 we have the system that was used in this study. The sampling period is 0.3 seconds, and the digital controller is designed to give a "deadbeat response" to a step input. From z-transform theory we have:

$$G(Z) = \frac{.368Z^{-1}+.264Z^{-2}}{1 - 1.368Z^{-1}+.368Z^{-2}} \qquad (1)$$

$$D(Z) = \frac{1.582 - .582Z^{-1}}{1 + .418Z^{-1}} \qquad (2)$$

The equations describing the system may also be written as:

$$\underline{x}(k+1) = A\underline{x}(k) + Dr(k) \tag{3}$$

$$c(k) = B\underline{x}(k) \tag{4}$$

Now if we decompose our system into its canonical form equations 2 and 3 may be written by inspection.

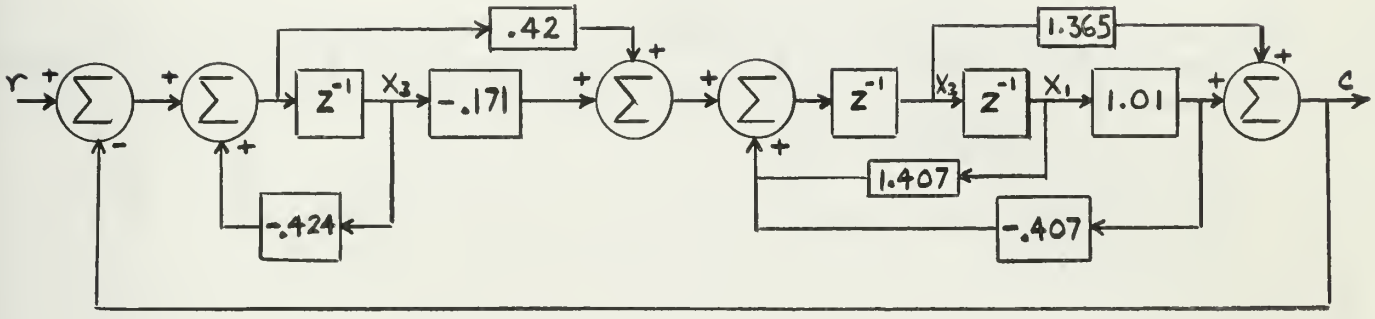


Figure 46

Canonical form of sampled-data control system

Thus, if we define our states as shown in Figure 46 we arrive at:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -.831 & .833 & -.349 \\ -1.01 & -1.365 & -.424 \end{bmatrix} \tag{5}$$

$$D = \begin{bmatrix} 0 \\ .42 \\ 1 \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} 1.01 & 1.365 & 0 \end{bmatrix} \tag{7}$$

Before the input, r(k), or the output, c(k), are introduced into the digital controller they pass through the A/D converter and are changed from analog signals to digital signals. Also, since the

digital computer must operate with finite word lengths, round off occurs for each word the computer processes or outputs. Therefore quantization operations occur and should be accounted for wherever these processes happen. This may be done by introducing quantizers in the canonical form of the system as shown in Figure 47.

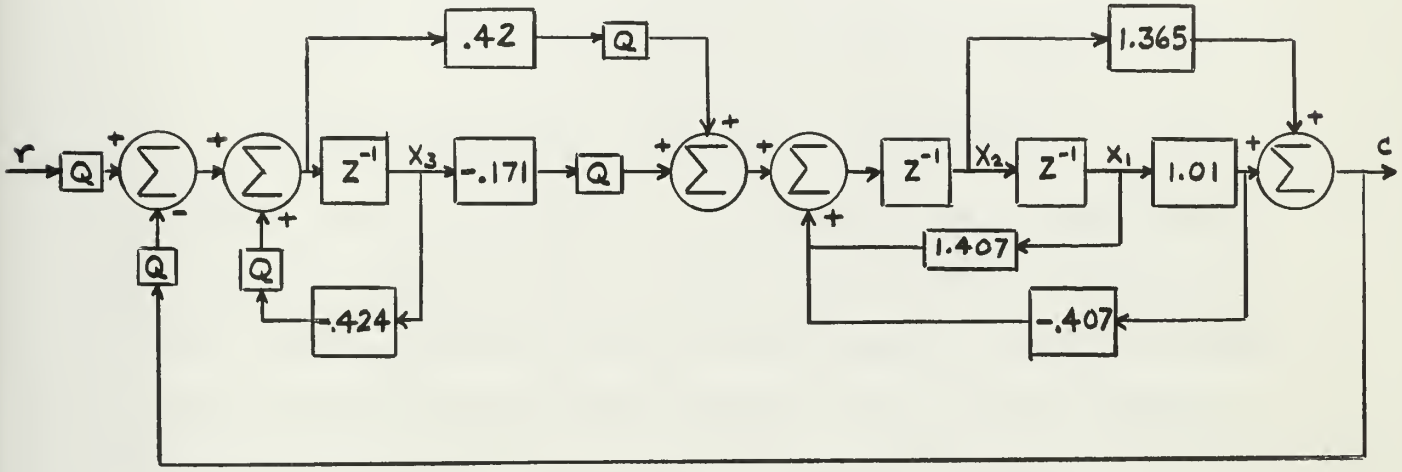


Figure 47

We now introduce a new vector $\underline{R}(k)$ which is the difference between the quantized and un-quantized terms. The quantized system may now be described by:

$$\underline{x}_q(k+1) = A\underline{x}_q(k) + \underline{R}(k) + Dr(k) \qquad (8)$$

where the subscript q indicates the quantized system. It may be seen by inspecting the above equation that if the un-quantized system is stable then the quantized system is also stable.

To obtain the difference between the quantized and un-quantized systems we define:

$$\underline{e}(k) = \underline{x}_q(k) - \underline{x}(k) \qquad (9)$$

from which:

$$\underline{e}(k+1) = A\underline{e}(k) + \underline{R}(k) \tag{10}$$

$$c_q(k) - c(k) = B\underline{e}(k) \tag{11}$$

Thus in our problem:

$$e_1(k+1) = e_2(k) \tag{12}$$

$$e_2(k+1) = -.831e_1(k) + .833e_2(k) - .349e_3(k) + R_2(k) \tag{13}$$

$$e_3(k+1) = -1.01e_1(k) - 1.365e_2(k) - .424e_3(k) + R_3(k) \tag{14}$$

We must now find an upper bound on R and hence on e. From this we may obtain the maximum output error due to quantization. Since $R_i(k)$ is the difference between the state vector component, $x_i(k+1)$, obtained in the quantized and un-quantized systems, an upper bound on $R_i(k)$ is the greatest error due to quantizing in one sampling period.

In our system we have:

$$R_1(k) = 0 \tag{15}$$

$$R_2(k) \leq \left| .42h_1 + .21h_2 + h_2h_3 \right| \triangleq H_2 \tag{16}$$

$$R_3(k) \leq \left| h_1 + .5h_2 \right| \triangleq H_3 \tag{17}$$

where $h_1/2$ is the A/D quantizing error, $h_2/2$ is the computer round off error, and $h_3/2$ is the D/A quantizing error

To find an upper bound on e(k) we must convert equations 12, 13, and 14 in their z-transform equivalents.

Thus:

$$e_1(z) = z^{-1}e_2(z) \tag{18}$$

$$e_2(z) = -.831z^{-1}e_1(z) + .833z^{-1}e_2(z) - .349z^{-1}e_3(z) + \frac{H_2 z^{-1}}{1-z^{-1}} \qquad (19)$$

$$e_3(z) = -1.01z^{-1}e_1(z) - 1.365z^{-1}e_2(z) - .424z^{-1}e_3(z) + \frac{H_3 z^{-1}}{1-z^{-1}} \qquad (20)$$

Referring to equations 7 and 11 it is seen that to find the overall system error due to quantization we only have to solve for $e_1(z)$ and $e_2(z)$. Also, the steady state error e_1 equals the steady state error e_2. That is: $\lim_{z\to 1} (1-z^{-1})e_1(z) = \lim_{z\to 1} (1-z^{-1})e_2(z)$

Therefore after solving equations 18, 19 and 20 simultaneously and applying the final value theorem we arrive at:

$$e_1(\infty) = e_2(\infty) = \frac{1.424H_2 - .349H_3}{.595} \qquad (21)$$

and since in the hardware used h_1 = 2.44 millivolts, h_2 = .0244 millivolts, and h_3 = 2.44 millivolts

then

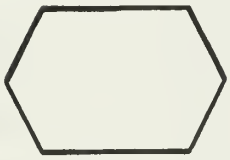
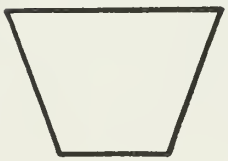
$$e_1(\infty) = e_2(\infty) = 15.3 \text{ millivolts} \qquad (22)$$

Therefore $c_q(k) - c(k) = 36.35$ millivolts

This difference between the quantized and unquantized systems was less than 1% of the final value of the system for a five volt step which was ordinarily used. Therefore no problems were encountered with quantization errors in the control of the real system.

APPENDIX V

PROGRAM FLOWCHART SYMBOLS

Symbol	Represents
	Processing
	External Function
	Terminal or Connector
	Decision
	Indexing Operation
	Predefined Process
	Jump

APPENDIX VI

PROGRAMS FOR THE CDC-160 COMPUTER

PROGRAM 1

Location	Contents	Comments
1000	7500	Call A/D
1001	1401	
1002	7600	Read into A
1003	4000	Store in 0000
1004	7500	Call D/A
1005	2401	
1006	7304	Output (0000)
1007	0001	
1010	6510	Jump to 1000
1011	6411	
1012	0000	

.

Program 2

Location	Contents	Comments
1000	7500	Call A/D
1001	1401	
1002	7600	Read into A
1003	4062	Store 'New value' in 0062
1004	3460	Subtract 'Old Value'
1005	4065	Store 'Diff' in 0065
1006	3063	Add Increment
1007	6315	If sum negative, jump to 1024
1010	2065	Load 'Diff' into A
1011	3463	Subtract Increment
1012	6211	If difference positive, jump to 1023
1013	2062	Load 'New Value'
1014	4060	Store as 'Old Value' in 0060
1015	7500	Call D/A
1016	2401	
1017	7306	Output (0060)
1020	0061	
1021	6521	Jump to 1000
1022	6422	
1023	7071	Jump to Negative Step Solution (0071)
1024	7072	Jump to Positive Step Solution (0072)
1025	0060	

PROGRAM 2 (CONTINUED)

LOW CORE ALLOCATION

Location	Contents	Comments
0060	----	'Old Value'
0062	----	'New Value'
0063	0010	Increment
0065	----	'Diff'
0071	7700	Dummy start of Negative Step Solution
0072	7700	Dummy start of Positive Step Solution

PROGRAM 3

Location	Contents	Comments
1030	7500	Call D/A
1031	2401	
1032	7317	Output (0020)
1033	0021	
1034	2050	Zeroize counter
1035	3450	
1036	4050	
1037	5450	Add one to counter
1040	2050	Subtract counter total from preset total
1041	3451	
1042	6002	If difference = 0, jump to 1044
1043	6504	If difference not = 0, jump to 1037
1044	7500	Call D/A
1045	2401	
1046	7304	Output (0022)
1047	0023	
1050	7700	Halt
1051	0020	
1052	0022	

.

PROGRAM 3 (CONTINUED)

LOW CORE ALLOCATION

Location	Contents	Comments
0020	0631	Converted as -4 volts
0022	7146	Converted as -6 volts
0050	----	Counter
0051	3403	Preset total for 0.1 second delay

PROGRAM 4

Location	Contents	Comments
1000	7500	
1001	1401	Sense Loop
1002	7600	(See PROGRAM 2)
1003	4062	
1004	3460	
1005	4065	
1006	3063	
1007	6315	
1010	2065	
1011	3463	
1012	6211	
1013	2062	
1014	4060	
1015	7500	
1016	2401	
1017	7306	
1020	0061	
1021	6521	
1022	6422	
1023	7071	
1024	7072	
1025	0060	

PROGRAM 4 (CONTINUED)

Location	Contents	Comments
		NEGATIVE STEP SOLUTION
1026	2060	Load 'Old Value'
1027	3014	Add H^0
1030	4020	Store adjusted H^0 in 0020
1031	2060	Load 'Old Value'
1032	3015	Add H^1
1033	4022	Store adjusted H^1 in 0022
1034	7500	Call D/A
1035	2401	
1036	7327	Output (0020)
1037	0021	
1040	2052	Start Delay
1041	3452	(See PROGRAM 3)
1042	4052	
1043	2050	
1044	3450	
1045	4050	
1046	5450	
1047	2050	
1050	3451	
1051	6002	
1052	6504	

PROGRAM 4 (CONTINUED)

Location	Contents	Comments
1053	5452	
1054	2052	
1055	3453	
1056	6002	
1057	6514	
1060	7500	Call D/A
1061	2401	
1062	7304	Output(0022)
1063	0023	
1064	7073	Jump to 1130
1065	0020	
1066	0022	
1067	2060	POSITIVE STEP SOLUTION Load 'Old Value'
1070	3016	Add H^0
1071	4024	Store adjusted H^0 in 0024
1072	2060	Load 'Old Value'
1073	3017	Add H^1
1074	4026	Store adjusted H^1 in 0026
1075	7500	Call D/A
1076	2401	
1077	7327	Output(0024)

PROGRAM 4 (CONTINUED)

Location	Contents	Comments
1100	0025	
1101	2052	Start Delay
1102	3452	(See PROGRAM 3)
1103	4052	
1104	2050	
1105	3450	
1106	4050	
1107	5450	
1110	2050	
1111	3451	
1112	6002	
1113	6504	
1114	5452	
1115	2052	
1116	3453	
1117	6002	
1120	6514	
1121	7500	Call D/A
1122	2401	
1123	7304	Output (0026)
1124	0027	

PROGRAM 4 (CONTINUED)

Location	Contents	Comments
1125	7074	Jump to 1130
1126	0024	
1127	0026	
1130	2052	Start Delay
1131	3452	(See PROGRAM 3)
1132	4052	
1133	2050	
1134	3450	
1135	4050	
1136	5450	
1137	2050	
1140	3451	
1141	6002	
1142	6504	
1143	5452	
1144	2052	
1145	3453	
1146	6002	
1147	6514	
		BOOTSTRAP
1150	7500	

PROGRAM 4 (CONTINUED)

Location	Contents	Comments
1151	1401	
1152	7600	Read into A
1153	4060	Store as 'Old Value'
1154	7070	Jump to 1000

LOW CORE ALLOCATION

0014	----	H^0 (-)
0015	----	H^1 (-)
0016	----	H^0 (+)
0017	----	H^1 (+)
0020	----	Adjusted H^0 (-)
0022	----	Adjusted H^1 (-)
0024	----	Adjusted H^0 (+)
0026	----	Adjusted H^1 (+)
0051	3403	Preset total for 0.1 second delay
0053	0003	Number of passes through 0.1 second delay
0060	----	'Old Value'
0062	----	'New Value'
0063	----	Increment
0065	----	'Diff'
0070	1000	
0071	1026	
0072	1067	
0073-0074	1130	

PROGRAM 5

Location	Contents	Comments
1000	7500	Sense Loop
1001	1401	
1002	7600	
1003	4062	
1004	3460	
1005	4065	
1006	3063	
1007	6315	
1010	2065	
1011	3463	
1012	6211	
1013	2062	
1014	4060	
1015	7500	
1016	2401	
1017	7306	
1020	0061	
1021	6521	
1022	6422	
1023	7011	Jump to Negative Step Solution
1024	7012	Jump to Positive Step Solution
1025	0060	

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1026	7500	Negative Step Solution
1027	1401	Call A/D
1030	7600	Read into A
1031	3460	Subtract 'Old Value'
1032	4024	Store E^{O}
1033	0101	Multiply E^{O} by 100 K^{O}
1034	7000	
1035	0031	
1036	0020	
1037	0024	
1040	0030	
1041	0101	Divide product by 100
1042	7000	
1043	0041	
1044	0030	
1045	0002	
1046	0030	
1047	2030	Load H^{O}
1050	3060	Add 'Old Value'
1051	4030	Store adjusted H^{O}

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1052	7500	Call D/A
1053	2401	
1054	7353	Output adjusted H°
1055	0031	
1056	2052	Delay
1057	3452	
1060	4052	
1061	2050	
1062	3450	
1063	4050	
1064	5450	
1065	2050	
1066	3451	
1067	6002	
1070	6504	
1071	5452	
1072	2052	
1073	3453	
1074	6002	

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1075	6514	
1076	7500	Call A/D
1077	1401	
1100	7600	Read into A
1101	3460	Subtract 'Old Value'
1102	4024	Store E^1
1103	0101	Multiply E^1 by $100K^1$
1104	7000	
1105	0031	
1106	0022	
1107	0024	
1110	0030	
1111	0101	Divide product by 100
1112	7000	
1113	0041	
1114	0030	
1115	0002	
1116	0030	
1117	2030	Load H^1
1120	3060	Add 'Old Value'

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1121	4030	Store adjusted H^1
1122	7500	Call D/A
1123	2401	
1124	7303	Output adjusted H^1
1125	0031	
1126	7013	Jump to second delay
1127	0030	
		Positive Step Solution
1130	7500	Call A/D
1131	1401	
1132	7600	Read into A
1133	3460	Subtract 'Old Value'
1134	4024	Store E^o
1135	2424	Complement E^o
1136	4024	
1137	0101	Multiply complemented E^o by 100 K^o
1140	7000	
1141	0031	
1142	0020	
1143	0024	
1144	0030	

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1145	0101	Divide product by 100
1146	7000	
1147	0041	
1150	0030	
1151	0002	
1152	0030	
1153	2430	Complement dividend
1154	4030	Store H^{O}
1155	2030	Load H^{O}
1156	3060	Add 'Old Value'
1157	4030	Store adjusted H^{O}
1160	7500	Call D/A
1161	2401	
1162	7357	Output adjusted H^{O}
1163	0031	
1164	2052	Delay
1165	3452	
1166	4052	
1167	2050	
1170	3450	

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1171	4050	
1172	5450	
1173	2050	
1174	3451	
1175	6002	
1176	6504	
1177	5452	
1200	2502	
1201	3453	
1202	6002	
1203	6514	
1204	7500	Call A/D
1205	1401	
1206	7600	Read into A
1207	3460	Subtract 'Old Value'
1210	4024	Store E^1
1211	2424	Complement E^1
1212	4024	
1213	0101	Multiply complemented E^1 by $100K^1$
1214	7000	
1215	0031	

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1216	0022	
1217	0024	
1220	0030	Divide product by 100
1221	0101	
1222	7000	
1223	0041	
1224	0030	
1225	0002	
1226	0030	
1227	2430	Complement dividend
1230	4030	
1231	2030	Load H^1
1232	3060	Add 'Old Value'
1233	4030	Store adjusted H^1
1234	7500	Call D/A
1235	2401	
1236	7303	Output adjusted H^1
1237	0031	
1240	7013	Jump to second delay

PROGRAM 5 (CONTINUED)

Location	Contents	Comments
1241	0030	
1242	2052	Second Delay
1243	3452	
1244	4052	
1245	2050	
1246	3450	
1247	4050	
1250	5450	
1251	2050	
1252	3451	
1253	6002	
1254	6504	
1255	5452	
1256	2052	
1257	3453	
1260	6002	
1261	6514	
1262	7500	Bootstrap
1263	1401	
1264	7600	
1265	4060	
1266	7010	

PROGRAM 5 (CONTINUED)

LOW CORE ALLOCATION

Location	Contents	Comments
0000	----	Address of Subroutine ARITH
0002	0144	
0010	1000	Address of start of program
0011	1026	Address of negative step solution
0012	1130	Address of positive step solution
0013	1164	Address of second delay
0020-21	----	K^0
0022-23	----	K^1
0024-25	----	Current error
0030-31	----	Current H
0051	3403	Preset total for 0.1 second delay
0053	0003	Number of passes through 0.1 second delay
0060	----	'Old Value'
0062	----	'New Value'
0063	----	Increment
0065	----	'Diff'

PROGRAM 6

Due to the basic similarity of this program to the preceding programs, the comments presented here will be general in nature.

Location	Contents	Comments
1000	7500	SENSE LOOP
1001	1401	
1002	7600	
1003	4064	
1004	7500	
1005	1402	
1006	7600	
1007	4065	
1010	2064	Sense error in
1011	3461	System 1
1012	4067	
1013	3063	
1014	6351	If error, set
1015	2067	E^1 Flag
1016	3463	
1017	6250	
1020	2064	
1021	4061	
1022	6101	
1023	4066	
1024	7500	If no error, D/A
1025	2401	System 1
1026	7353	
1027	0067	
1030	2065	Sense error in
1031	3462	System 2
1032	4067	
1033	3063	
1034	6337	If error, set
1035	2067	E^2 Flag and
1036	3463	jump to solution
1037	6236	routine
1040	2065	
1041	4062	
1042	6101	
1043	4066	
1044	7500	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1045	2402	If no error,
1046	7333	D/A System 2
1047	0067	
1050	2040	If E^1 Flag set,
1051	6123	jump to Solution
1052	6122	Routine
1053	2050	
1054	3450	Basic Delay
1055	4050	
1056	5450	
1057	2050	
1060	3451	
1061	6002	
1062	6504	
1063	6463	Return to start
1064	6564	of Sense Loop
1065	5440	
1066	6536	
1067	2040	
1070	3401	
1071	4040	
1072	6542	
1073	5441	
1074	7010	
1075	2041	
1076	3401	
1077	4041	
1100	7010	
1101	0066	
1102	0000	
1103	0000	
1104	0000	
1105	2067	
1106	3467	
1107	4067	
		SOLUTION ROUTINE
1110	2040	E^1 Flag test
1111	6003	
1112	6304	If = 0, jump to
1113	6205	E^2 Flag test

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1114	7101	If = -1, jump to
1115	1370	Negative Step
1116	7101	Solution for System 1
1117	1250	
		Positive Step
1120	2042	Solution for
1121	6002	System 1
1122	6137	If J $\neq$ 0, jump to 1161
1123	7500	
1124	1401	
1125	7600	Compute and output
1126	3461	$H^{o}(+)$ for System 1
1127	4020	
1130	2420	
1131	4020	
1132	0101	
1133	7000	
1134	0031	
1135	0020	
1136	0022	
1137	0016	
1140	0101	
1141	7000	
1142	0041	
1143	0016	
1144	0004	
1145	0016	
1146	2416	
1147	3061	
1150	4016	
1151	7500	
1152	2401	
1153	7305	
1154	0017	
1155	5442	J = J + 1
1156	7101	
1157	1370	Jump to E^2 Flag test
1160	0016	
1161	5443	I = I + 1
1162	2043	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1163	3403	If I ≠ (0003),
1164	6506	jump to E^2 Flag test
1165	2042	
1166	3401	If J ≠ 1, jump to
1167	6003	E^1R Flag test
1170	7101	
1171	1760	
1172	7500	Compute and output
1173	1401	$H^1(+)$ for System 1
1174	7600	
1175	3461	
1176	4066	
1177	2466	
1200	4066	
1201	3420	
1202	6303	
1203	7101	
1204	1205	
1205	0101	
1206	7000	
1207	0031	
1210	0066	
1211	0024	
1212	0016	
1213	0101	
1214	7000	
1215	0041	
1216	0016	
1217	0004	
1220	0016	
1221	2416	
1222	3061	
1223	4016	
1224	7500	
1225	2401	
1226	7310	
1227	0017	
1230	5442	J = J + 1
1231	2043	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1232	3443	I = 0
1233	4043	
1234	7101	Jump to E^2 Flag test
1235	1370	
1236	0016	
1237	0000	
1240	0000	
1241	0000	
1242	0000	
1243	0000	
1244	0000	
1245	0000	
1246	0000	
1247	0000	
1250	2042	Negative Step Solution for System 1
1251	6002	
1252	6135	
1253	7500	
1254	1401	
1255	7600	
1256	3461	(This solution is handled exactly as is its positive counterpart. Thus, no comments will be made.)
1257	4020	
1260	0101	
1261	7000	
1262	0031	
1263	0020	
1264	0022	
1265	0016	
1266	0101	
1267	7000	
1270	0041	
1271	0016	
1272	0004	
1273	0016	
1274	2016	
1275	3061	
1276	4016	
1277	7500	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1300	2401	
1301	7305	
1302	0017	
1303	5442	
1304	7101	
1305	1370	
1306	0016	
1307	5443	
1310	2043	
1311	3403	
1312	6506	
1313	2042	
1314	3401	
1315	6003	
1316	7101	
1317	1760	
1320	7500	
1321	1401	
1322	7600	
1323	4066	
1324	3420	
1325	6303	
1326	7101	
1327	1330	
1330	0101	
1331	7000	
1332	0031	
1333	0066	
1334	0024	
1335	0016	
1336	0101	
1337	7000	
1340	0041	
1341	0016	
1342	0004	
1343	0016	
1344	2016	
1345	3061	
1346	4016	
1347	7500	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1350	2401	
1351	7310	
1352	0017	
1353	5442	
1354	2043	
1355	3443	
1356	4043	
1357	7101	
1360	1370	
1361	0016	
1362	0000	
1363	0000	
1364	0000	
1365	0000	
1366	0000	
1367	0000	
1370	2041	E^2 Flag test
1371	6003	
1372	6304	If = 0, jump to
1373	6205	delay
1374	7101	
1375	1650	If = -1, jump to
1376	7101	Negative Step
1377	1525	Solution for System 2
1400	2044	
1401	6002	Positive Step
1402	6137	Solution for
1403	7500	System 2
1404	1402	If L $\neq$ 0, jump to
1405	7600	1441
1406	3462	
1407	4030	
1410	2430	Compute and output $H^{O}(+)$ for System 2
1411	4030	
1412	0101	
1413	7000	
1414	0031	
1415	0030	
1416	0032	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1417	0016	
1420	0101	
1421	7000	
1422	0041	
1423	0016	
1424	0004	
1425	0016	
1426	2416	
1427	3062	
1430	4016	
1431	7500	
1432	2402	
1433	7305	
1434	0017	
1435	5444	L = L + 1
1436	7101	
1437	1650	Jump to delay
1440	0016	
1441	5445	K = K + 1
1442	2045	
1443	3402	If K ≠ (0002),
1444	6506	jump to delay
1445	2044	
1446	3401	If L ≠ 1, jump to
1447	6003	E^2R Flag test
1450	7101	
1451	2040	
1452	7500	
1453	1402	Compute and output
1454	7600	$H^1(+)$ for System 2
1455	3462	
1456	4066	
1457	2466	
1460	3430	
1461	6303	
1462	7101	
1463	1464	
1464	0101	
1465	7000	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1466	0031	
1467	0066	
1470	0034	
1471	0016	
1472	0101	
1473	7000	
1474	0041	
1475	0016	
1476	0004	
1477	0016	
1500	2416	
1501	3062	
1502	4016	
1503	7500	
1504	2402	
1505	7310	
1506	0017	
1507	5444	L = L + 1
1510	2045	
1511	3445	K = 0
1512	4045	
1513	7101	Jump to delay
1514	1650	
1515	0016	
1516	0000	
1517	0000	
1520	0000	
1521	0000	
1522	0000	
1523	0000	
1524	0000	
1525	2044	Negative Step
1526	6002	Solution for
1527	6135	System 2
1530	7500	(As before, no
1531	1402	comments will be made
1532	7600	for the negative
1533	3462	step solution.)
1534	4030	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1535	0101	
1536	7000	
1537	0031	
1540	0030	
1541	0032	
1542	0016	
1543	0101	
1544	7000	
1545	0041	
1546	0016	
1547	0004	
1550	0016	
1551	2016	
1552	3062	
1553	4016	
1554	7500	
1555	2402	
1556	7305	
1557	0017	
1560	5444	
1561	7101	
1562	1650	
1563	0016	
1564	5445	
1565	2045	
1566	3402	
1567	6506	
1570	2044	
1571	3401	
1572	6003	
1573	7101	
1574	2040	
1575	7500	
1576	1402	
1577	7600	
1600	3462	
1601	4066	
1602	3430	
1603	6303	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1604	7101	
1605	1606	
1606	0101	
1607	7000	
1610	0031	
1611	0066	
1612	0034	
1613	0016	
1614	0101	
1615	7000	
1616	0041	
1617	0016	
1620	0004	
1621	0016	
1622	2016	
1623	3062	
1624	4016	
1625	7500	
1626	2402	
1627	7310	
1630	0017	
1631	5444	
1632	2045	
1633	3445	
1634	4045	
1635	7101	
1636	1650	
1637	0016	
1640	0000	
1641	0000	
1642	0000	
1643	0000	
1644	0000	
1645	0000	
1646	0000	
1647	0000	
1650	2050	Delay
1651	3450	
1652	4050	
1653	5450	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1654	2050	
1655	3451	
1656	6002	
1657	6504	
1660	2040	If E^1 Flag $\neq$ 1,
1661	6133	sense error for
1662	7500	System 1
1663	1401	
1664	7600	
1665	4064	
1666	3461	
1667	4067	
1670	3063	
1671	6316	
1672	2067	
1673	3463	
1674	6215	
1675	2064	
1676	4061	
1677	6101	
1700	4066	
1701	7500	
1702	2401	
1703	7303	
1704	0067	
1705	6107	
1706	0066	
1707	5440	
1710	6104	
1711	2040	
1712	3401	
1713	4040	
1714	2041	If E^2 Flag = 0,
1715	6133	sense error for
1716	7500	System 2
1717	1402	
1720	7600	
1721	4065	
1722	3462	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1723	4067	
1724	3063	
1725	6316	
1726	2067	
1727	3463	
1730	6215	
1731	2065	
1732	4062	
1733	6101	
1734	4066	
1735	7500	
1736	2402	
1737	7303	
1740	0067	
1741	7110	
1742	0066	
1743	5441	
1744	7105	
1745	2041	
1746	3401	
1747	4041	
1750	7010	Jump to start of
1751	1105	Solution Routine
1752	0000	
1753	0000	
1754	0000	
1755	0000	
1756	0000	
1757	0000	
		RETURN ROUTINE
1760	2046	If E^1R Flag $\neq 0$,
1761	6132	jump to ramp solution
1762	2042	routine (not included)
1763	3442	
1764	4042	J = 0
1765	2043	
1766	3443	I = 0
1767	4043	
1770	2040	E^1 Flag = 0
1771	3440	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
1772	4040	
1773	2041	If E^2 Flag $\neq 0$,
1774	6112	bootstrap System 1
1775	7500	and jump to Solution
1776	1401	Routine
1777	7600	
2000	4061	
2001	7500	Bootstrap Systems
2002	1402	1 and 2, and jump
2003	7600	to Sense Loop
2004	4062	
2005	7011	
2006	7500	
2007	1401	
2010	7600	
2011	4061	
2012	7010	
2013	7101	
2014	1105	
2015	0000	
2016	0000	
2017	0000	
2020	0000	
2021	0000	
2022	0000	
2023	0000	
2024	0000	
2025	0000	
2026	0000	
2027	0000	
2030	0000	
2031	0000	
2032	0000	
2033	0000	
2034	0000	
2035	0000	
2036	0000	
2037	0000	
2040	2047	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
2041	6132	If E^2R Flag $\neq 0$,
2042	2044	jump to ramp
2043	3444	solution routine
2044	4044	
2045	2045	L = 0
2046	3445	
2047	4045	K = 0
2050	2041	E^2 Flag = 0
2051	3441	
2052	4041	
2053	2040	If E^1 Flag $\neq 0$,
2054	6112	bootstrap System 2
2055	7500	and jump to
2056	1401	Solution Routine
2057	7600	
2060	4061	
2061	7500	Bootstrap Systems
2062	1402	1 and 2, and jump
2063	7600	to Sense Loop
2064	4062	
2065	7011	
2066	7500	
2067	1402	
2070	7600	
2071	4062	
2072	7010	
2073	7101	
2074	4000	

LOW CORE ALLOCATION

0000	----	Address of Sub-routine ARITH
0001	0001	
0002	----	Q
0003	----	P
0004	0144	
0010	1105	
0011	1000	

PROGRAM 6 (CONTINUED)

Location	Contents	Comments
0016-17	----	H's for output
0020-21	----	E(1)
0022-23	----	$100K^{0}(1)$
0024-25	----	$100K^{1}(1)$
0030-31	----	E(2)
0032-33	----	$100K^{0}(2)$
0034-35	----	$100K^{1}(2)$
0040	----	E^{1} Flag
0041	----	E^{2} Flag
0042	----	J
0043	----	I
0044	----	L
0045	----	K
0046	----	$E^{1}R$ Flag
0047	----	$E^{2}R$ Flag
0050	----	Delay counter
0051	0675	Preset total for 25 milli-second delay
0060	----	Old Value (1)
0061	----	Old Value (2)
0063	0100	Increment
0064	----	New Value (1)
0065	----	New Value (2)
0066	----	Temp Storage

Note: Cells 0070 through 0077 and cell 0007 are reserved for Subroutine ARITH.