
Retrospective Theses and Dissertations

Winter 1972

Analog/Hybrid Computer Simulation Applied to Sampled-Data Control Systems

Sam Joseph Monte
University of Central Florida

 Part of the [Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/rtd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Monte, Sam Joseph, "Analog/Hybrid Computer Simulation Applied to Sampled-Data Control Systems" (1972). *Retrospective Theses and Dissertations*. 24.
<https://stars.library.ucf.edu/rtd/24>

ANALOG/HYBRID COMPUTER SIMULATION APPLIED
TO SAMPLED-DATA CONTROL SYSTEMS

BY

SAM JOSEPH MONTE
B.S., Auburn University, 1960

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program of
Florida Technological University, 1972

Orlando, Florida

ACKNOWLEDGMENTS

The author wishes to express his sincere gratitude to his advisor and committee chairman, Dr. Fred O. Simons, for suggesting the project which served as the subject matter for this thesis and for his untiring patience and guidance. He is also thankful to Mrs. Marilyn Ross for the many hours she spent in typing this thesis.

This thesis is dedicated to my wonderful wife whose encouragement and patience were unending.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iii
LIST OF TABLES	v
LIST OF ILLUSTRATIONS	vi
INTRODUCTION	1
Chapter	
I. BASIC ANALOG COMPUTER SIMULATION	6
EAI 580 Logic Program	
Impulse Simulation	
EAI 680 Logic Program	
A/D-5 Logic Program	
II. SECOND ORDER SYSTEM SIMULATION	23
EAI 580 Analog Programs	
EAI 680 Analog Programs	
A/D-5 Analog Programs	
III. REAL TIME SIMULATION OF SAMPLED-DATA SYSTEMS .	48
IV. ADAPTIVE AND MULTIRATE SAMPLING	58
V. CONCLUSIONS AND RECOMMENDATIONS	100
APPENDIXES	103
REFERENCES	120

LIST OF TABLES

Table	Page
1. Performance Data of the Control Law With Various Order Holds	72
2. Performance Data of the Control Law With a True First-Order Hold	81

LIST OF ILLUSTRATIONS

Figure		Page
1.	Sampled-Data Control System Block Diagram . . .	4
2.	Typical Adaptive Sampled-Data Control System .	4
3.	Sampling Process	8
4.	EAI 580 Logic Program	12
5.	Simple Sampled-Data System	14
6.	Simulation of System in Figure 5	14
7.	Output of System in Figure 5 When Input Equal Step Function	15
8.	Logic Variables for Up-Data Period (EAI 680) .	17
9.	Logic Program for Up-Data Period (EAI 680) . .	18
10.	A/D-5 Logic Program for Up-Data Period	20
11.	Logic Variables for Up-Data Period (A/D-5) . .	21
12.	Second Order System With No Sampler	25
13.	Second Order System With Sampler But No Hold Circuit (EAI 580)	26
14.	Variables of the Simulation in Figure 13 . . .	27
15.	Comparison of Output Responses of Continuous System and No Hold System	28
16.	Simulation of Second-Order System With Zero- Order Hold (EAI 580)	29
17.	Zero-Order Hold Output and System Output Responses for Figure 16	31
18.	Two More Variables From the Simulation Shown in Figure 16	32

Figure		Page
19.	Second Order System With First-Order Hold . . .	33
20.	Additional Logic Control Variables Needed for First-Order Hold Microprogram	34
21.	System Variables for the Simulation Shown in Figure 19	36
22.	Output Comparison for Four Different Systems .	37
23.	Second-Order System With Second-Order Hold . .	38
24.	Three System Variables of Simulation Shown in Figure 23	40
25.	Comparison of Output Responses of a Second- Order System With Various Order Hold Devices	42
26.	EAI 680 Time Base Generator and Second Order Continuous System	44
27.	Second-Order System With Sampler But No Hold Device (EAI 680)	45
28.	Second-Order System With Zero-Order Hold (A/D-5)	47
29.	"Real Time" Simulation of a Second-Order System With No Hold Device	50
30.	"Real Time" Simulation of a Second-Order System With Zero-Order Hold	51
31.	"Real Time" Simulation of a Second-Order System With a First-Order Hold	52
32.	Output Responses for Step Input	56
33.	Voltage Control Oscillator	59
34.	"Real Time" Simulation of a Second Order System With Multimode Hold Circuit	61
35.	Bekey and Tomovic's Adaptive Sampling Control Law	63
36.	System Variables for Simulation in Figure 34 When Hold is Zero-Order and $T = T_{\max}$	64

Figure	Page
37. System Variables for Simulation in Figure 34 When Hold is First-Order and $T = T_{\max}$	65
38. ISE as a Function of the Error Fractional Order Hold Parameter	67
39. System Variables When $k_e = k_{e(\text{opt})}$	68
40. System Variables When $k_e = k_c = 0$, $N_s = 20$ and Minimum ISE	70
41. System Variables When $k_e = k_c = 0$ and Minimum N_s	71
42. System Variables When $k_e = 0$, $k_c = 1$, $N_s = 20$ and Minimum ISE	73
43. System Variables When $k_e = 0$, $k_c = 1$ and Minimum N_s	74
44. System Variables When $k_e = 1$, $k_c = 0$, $N_s = 16$ and Minimum ISE	76
45. System Variables When $k_e = 1$, $k_c = 0$ and Minimum N_s	77
46. System Variables When $k_e = 1$, $k_c = 1$, $N_s = 15$ and Minimum ISE	78
47. System Variables When $k_e = 1$, $k_c = 1$ and Minimum N_s	79
48. Microprogram to Produce $1/T_n$ Function	80
49. ISE Versus B	83
50. Normalized ISE Versus Number of Samples for $k_e = 1$, $k_c = 1$ and T_{\max}	84
51. Normalized ISE Versus Number of Samples for $k_e = k_c = 1$ and $B_1 = 0.1$	85
52. Normalized ISE Versus Number of Samples for $k_e = 0$, $k_c = 1$	86
53. Normalized ISE Versus Number of Samples for $k_e = 0$, $k_c = 0$	87
54. Normalized ISE Versus Number of Samples for $k_e = 1$, $k_c = 0$	88

Figure		Page
55.	ISE Versus Number of Samples for Control Law in Equation 28	90
56.	Equation 28 Control Law System Variables When $K_1 = 0$	91
57.	Equation 28 Control Law System Variables When $K_1 = 10$	92
58.	ISE Versus Number of Samples for Control Law in Equation 30	93
59.	Equation 30 Control Law System Variables When $K_1 = K_2 = 10$	94
60.	More Equation 30 Control Law System Variables When $K_1 = K_2 = 10$	95
61.	Open-Loop Block Diagram of Multirate Sampling System	97
62.	Multirate Sampling Addition to Figure 33 . . .	99
63.	Impulse Response of a Zero-Order Hold	109
64.	Decomposition of the Impulse Response of a Zero-Order Hold	109
65.	Impulse Response of a First-Order Hold	111
66.	Impulse Response of a Second-Order Hold . . .	111
67.	Corresponding Regions in Z and S Planes . . .	114
68.	Second-Order Sampled-Data System With No Hold	114
69.	Stability Region for Second Order System With No Hold	117
70.	Stability Region for Second Order System With Zero-Order Hold	118

INTRODUCTION

Sampled-data systems are dynamic systems in which one or more of the system variables can change only at discrete time instants. Such systems are also called discrete time systems and differ from ordinary continuous systems in that the signals are in a sampled or pulse data form. Pulsed radar units, multichannel time division data link systems, and especially digital computer control systems are a few important examples of this type of system.

In reality, a sampled-data system is nothing more than a continuous system with a sampling operation being performed on some of the continuous variables within the system. The output of any sampler is not continuous with respect to time but is in the form of a train of pulses occurring at discrete instants of time. These distinct time instants are periodic and short in duration and they occur whenever the sampler closes. Thus, data on the output of the sampler is a train of amplitude-modulated pulses of width equal to the sampler closure time occurring at the sampling instances. A filtering system generally follows the sampler.

When sampled-data control systems are investigated the problem of modeling the sampling device arises. This

modeling is generally accomplished by an ideal sampler followed by a smoothing or filtering circuit which is commonly referred to as a hold or clamp circuit. An ideal sampler is a device which produces an output train of impulse functions whose amplitudes are equal to the function being sampled at the sample instances. The hold circuit has the important function to reproduce approximately the form of the input signal of the sampler by an interpolation or extrapolation process of the pulse train. The hold circuit is also generally followed by the usual components of the control feedback loop, shown in Figure 1 as Blocks H and G, comprising amplifiers, shaping networks, and the controlled member.

Fixed frequency sampling is the more common sampling technique used in most sampled-data control systems. Multi-rate sampling is a different form of fixed frequency sampling. This form of sampling technique is used in a system that contains two or more samplers. One or more of the samplers operate at some multiple of the basic sampling rate.

In recent years another sampling technique has been investigated [1, 2, 3, 4, and 5] and that technique is referred to as adaptive sampling. In adaptive sampling, the sampling rate of the sampling device is continuously varied, according to some control law, as the system operates. The

maintaining of the overall system performance while reducing the total number of samples over a given period of time is the primary objective of adaptive sampling. As the total number of samples is reduced with a satisfactory performance of the system, the so-called "sampling efficiency" of the system is increased. Figure 2 is a general representation of an adaptive control system when the input to the control law is generally a function of either the continuous error, $e(t)$, the derivative of the continuous error, $\dot{e}(t)$, or the output of the system, $c(t)$.

The purpose of this study is to demonstrate how the analog computer can be utilized in a manner that would capitalize on its high speed dynamic capabilities in the study, analysis and synthesis of all types of sampled-data control systems. In the techniques that are developed in this study, all of the system variables are simulated in their natural forms. This means that continuous variables are valid at all points in time, whereas the system discrete variables are obtained as discrete values at each instant. This approach is certainly a plausible type of simulation since in actuality a sampled-data control system is nothing more than a continuous system with samples.

With this basic concept in mind, it is easy to conclude that continuous portions of a sampled-data system can be simulated on standard analog hardware but some type(s) of

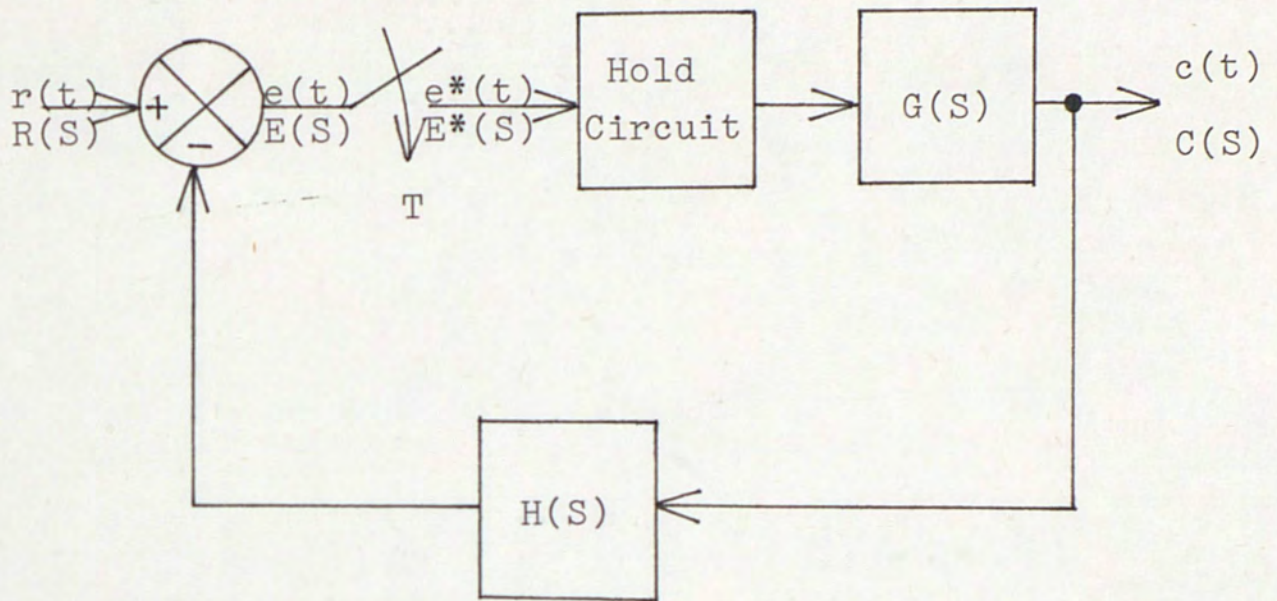


Figure 1. Sampled-Data Control System Block Diagram

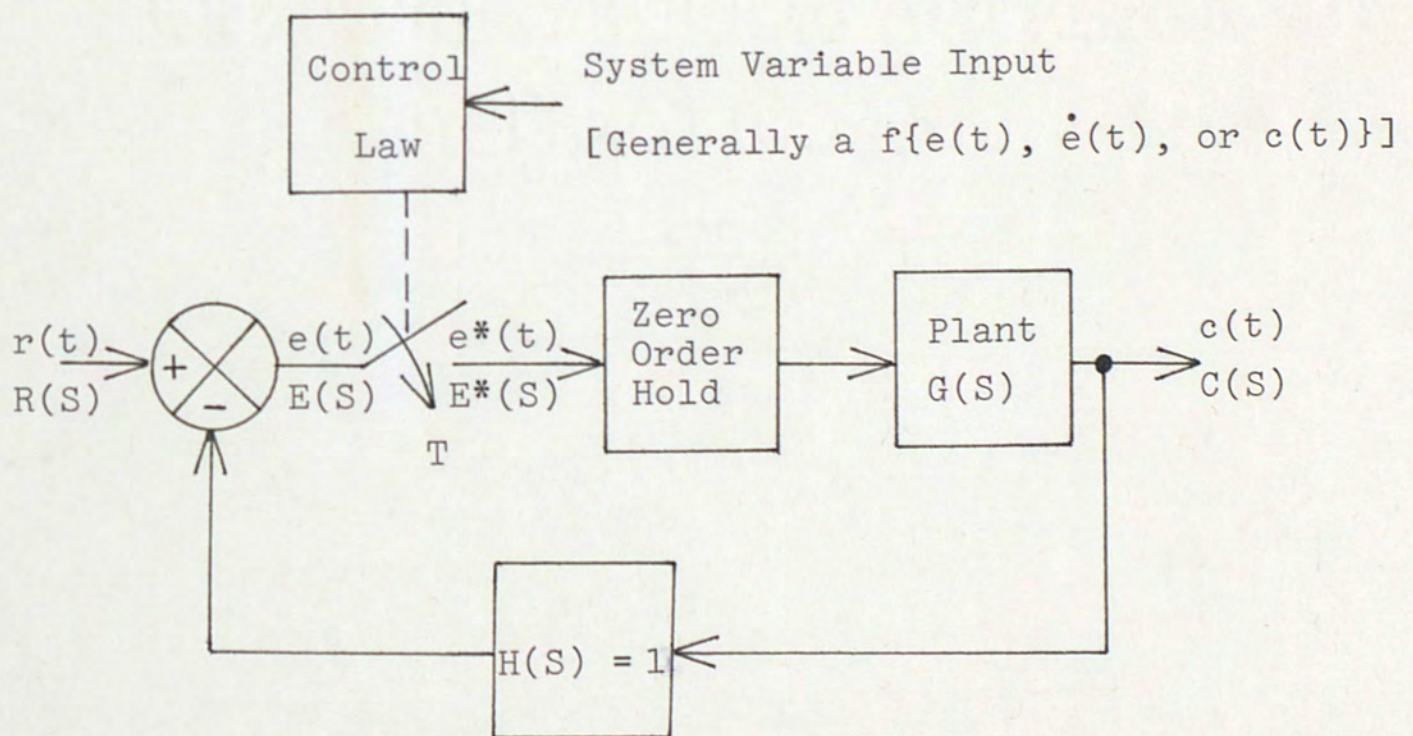


Figure 2. Typical Adaptive Sampled-Data Control System

digital and/or switching devices must be used in conjunction to accomplish the simulation of the sampler devices.

On most analog computers, there exists an option that is in the form of "digital logic." This particular option, as explained by Electronic Associates [6], has been generated for the primary purpose of augmenting the computing capabilities of the basic analog computer. The computer is therefore able to represent digital-controlled software quite adequately using this logic option and thus make it suitable for simulating the complete control system, both continuous and discrete. Any analog computer containing the previously mentioned "digital logic" is generally referred to as an analog/hybrid computer.

CHAPTER I

BASIC ANALOG COMPUTER SIMULATION

As described before a sampled-data system has some sampling operations being performed, which means that discrete signals are involved in the simulation of such a system. Since digital control logic now will be required to exercise control over these discrete signals, the particular logic complement of the analog/hybrid computer to be used will be of interest.

Three different digital logic programs were utilized in the sampled-data simulation study primarily due to the fact that three different analog/hybrid computers were employed in the complete investigation. The three computers were the Electronic Associates, Incorporated (EAI) 580, the EAI 680, and the Applied Dynamics (A/D)-5. The 680 is really a more sophisticated version of the 580, while the A/D-5 is similar to the 680.

The Repetitive-Operation (Rep-Op) interval timer, which is an integral part of the logic complement of the EAI 680 and the A/D-5, but is not part of the EAI 580 logic complement, is the principal reason for the differences in the generation technique of the timing logic for the three

computers. Another difference is that the two EAI computers use AND gates, while the A/D-5 computer uses only OR gates. The integrators of the EAI 680 and A/D-5 are of the three electronic mode control (EMC) variety, while those contained in the EAI 580 are two EMC.

It was stated that a sampled-data system was in reality a continuous system with some of the continuous variables being subjected to a sampling operation. The symbolic representation for synchronous sampler with a fixed sample period of T is shown in Figure 3. If $e(t)$ and $e^*(t)$ are used to denote the input and output, respectively, of the sampler, the output can be given by the following relation:

$$e^*(t) = e(t)\delta_T(t) \quad (1)$$

where $\delta_T(t)$ denotes a train of unit impulses (delta functions);

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT) \quad (2)$$

where the term $\delta(t-nT)$ indicates a train of unit impulses occurring at time $t = nT$.

Substitution of Equation (2) into Equation (1) produces a mathematical expression for

$$e^*(t) = \sum_{n=0}^{\infty} e(nT)\delta(t-nT) \quad (3)$$

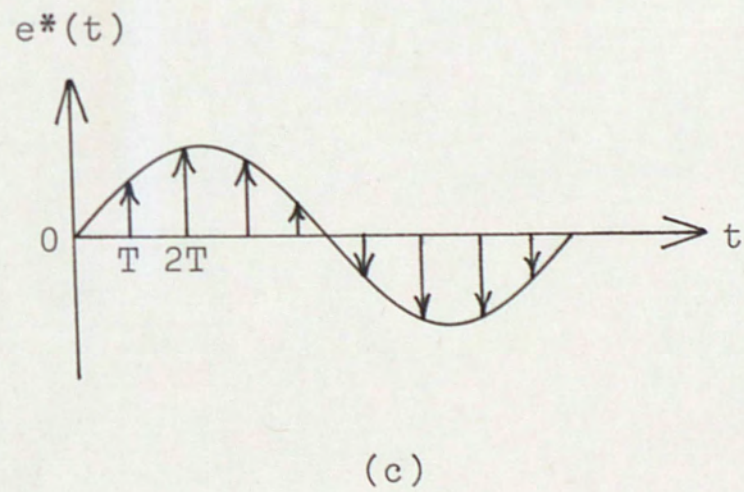
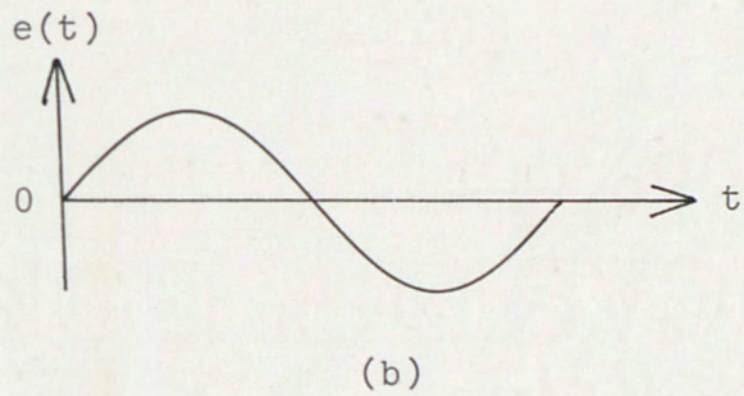
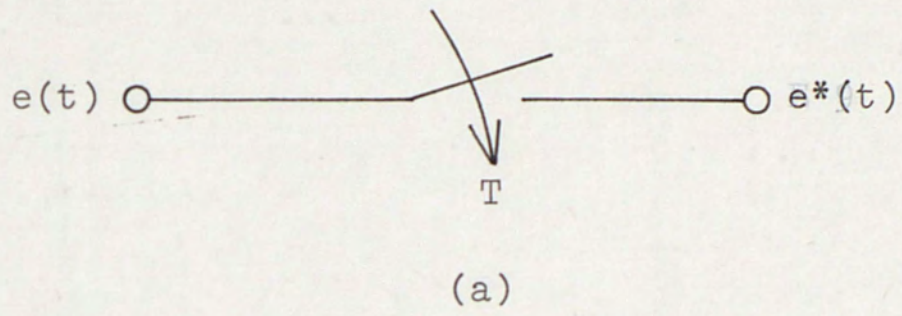


Figure 3. Sampling Process

where the negative values of n are absent by assuming that $e(t)$ does not exist for negative values of time. The output pulses then are the continuous variable $e(t)$ converted into a train of impulses with weightings $e(nT)$.

From the above discussion it becomes immediately apparent that the impulses, $e(nT)\delta(t-nT)$, nor their derivatives are readily simulated on an analog computer. Therefore, in simulating any sampled-data system, this problem must first be overcome. Ogata [7] states that an impulse to a differential equation can be shown to be equivalent to a jump in the appropriate initial conditions.

Consider the system of Figure 1 with the hold circuit removed and the feedback block $H(S)$ equal to unity. The output $c(t)$ and the error signal which has been sampled, $e^*(t)$, are expressed by the differential equation below:

$$\ddot{c}(t) + \dot{c}(t) = e^*(t) \quad (4)$$

where $e^*(t)$ is the output of an ideal sampler given by Equation (1). Using the notation of p being defined as the d/dt operator, Equation (4) can be rewritten as

$$c(t) = -\frac{1}{p}c(t) + \frac{1}{p^2}e^*(t) \quad (5)$$

Now, if the $e^*(t)$ impulses were available on the analog computer the simulation of this equation could be accomplished fairly easy. One method would be to feed the impulse function into an integrator with a gain of one.

This integrator has its output tied back to a second gain one input and the output is also fed into another gain one integrator. The output would then jump by the amount of the weightings $e(nT)$ of the impulse function. Therefore, if the initial condition of the integrator that is normally driven by the impulse train is updated, it would be equivalent to driving an integrator with an impulse train.

EAI 580 Logic Program

A logic program was generated on the EAI 580 to accomplish this updating operation. The PP button on both the logic panel and the analog panel are depressed to begin the operation. This places the logic panel in a patch panel mode and the normal machine logic signals IC(A) and OP(\bar{A}) are used to drive the logic mode controls of "clear" and "run," respectively. There was no need for analog and digital synchronization since this particular EAI 580 has been modified so that the frequency divider circuits were controlled by these same modes. Counters 0 and 1 are used to control the length of the sample period (T_s) and also the update (T_u) period. The sample period is defined by the following expression

$$T_s = N_0 \left(\frac{1}{10} k \right)$$

while the update period equation is thus

$$T_u = N_1 \left(\frac{1}{10} n \right)$$

Figure 4(a) shows the timing variables and Figure 4(b) indicates how the variables were generated. The logic variable A_1 places the simulation in a series of operate-update periods. All of these appear within each machine operate period which is controlled by the variable A of the simulation. Time is considered "frozen" or being held constant during these update periods. During this time span the output (IC) of the integrators that would have been driven by the impulse train are updated at each nT by the weightings $e(nT)$. As shown in Figures 4(a) and (b) other logic variables are generated and are utilized as the sampled-data system complexity increases.

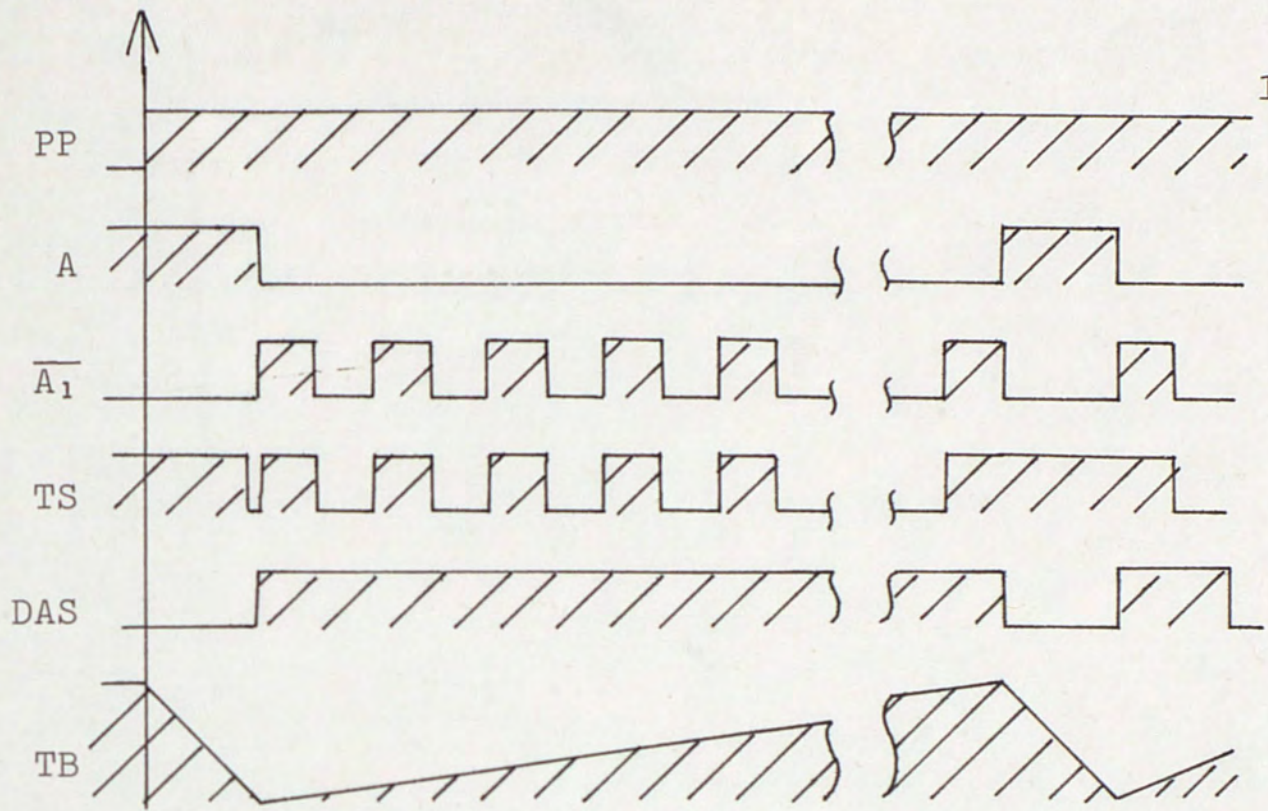
Impulse Simulation

One of the more common examples of a simple sampled-data system is shown in Figure 5. This example will be used to explain the analog portion of simulating the important impulse function. The system behavior is described by the following differential equation.

$$\frac{e^*-v}{R} = C \frac{dv}{dt} \quad (6)$$

If the voltage (v) variation with respect to time is defined as \dot{v} then

$$\dot{v} = \frac{1}{RC}(e^*-v) \quad (7)$$



TB--Analog Time Base

PP--Analog Rep-Op Logic Signal

(a)

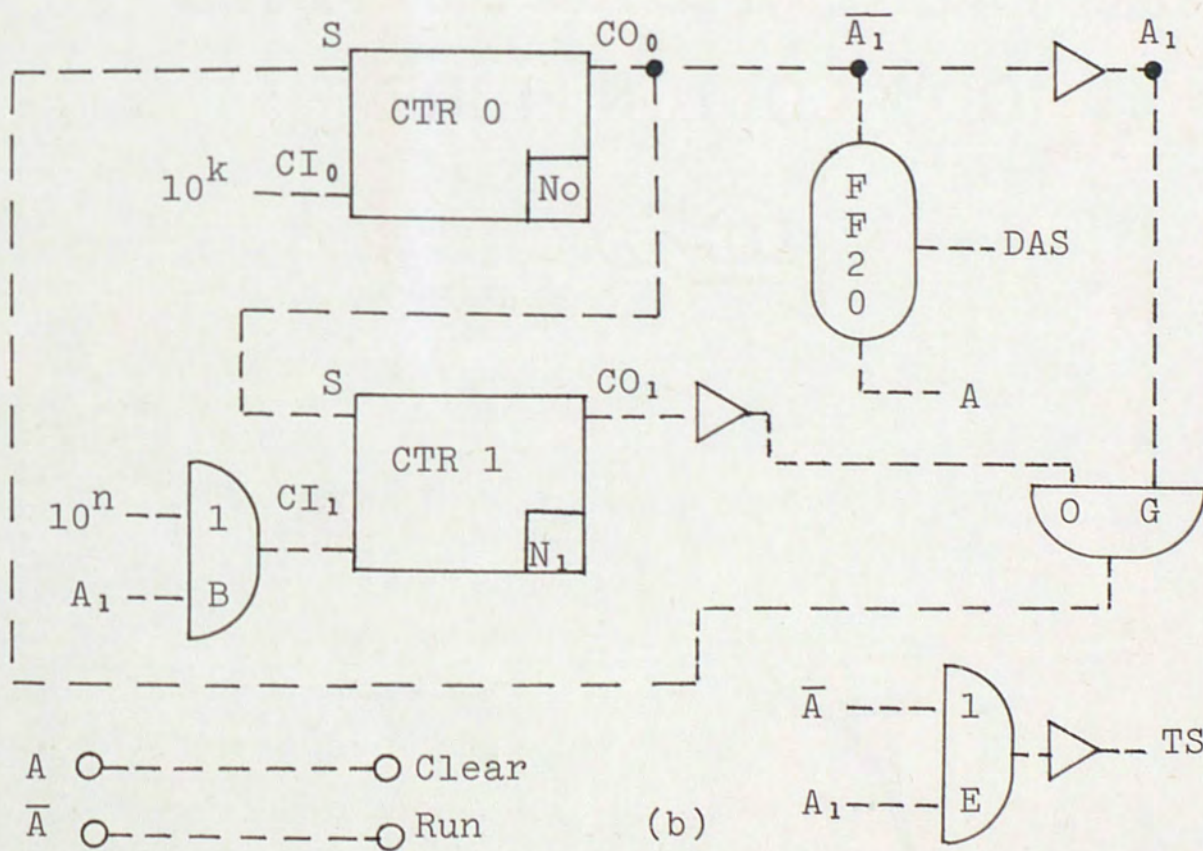


Figure 4. EAI 580 Logic Program

By also defining a new time scale variable (τ) as thus

$$\tau = \frac{1}{RC}(t) \quad (8)$$

and substituting this new time scale variable into Equation (7) the following evolves

$$\dot{v}(\tau) = e^*-v(\tau) \quad (9)$$

Therefore, the system in Figure 5, which is described by Equation (9), was simulated using the program in Figure 6. The problem time base generator (Integrator 11) is used to make the system go through an operate/hold manner. This is nothing more than the equivalent of making the integrators of the EAI 580 computer from a 2-mode EMC variety to a 3-mode variety. The logic variable DAS, which is the output of flip-flop 20, is provided to open the track/store and integrator initial condition loop during the IC period of the simulation. The signal TS is utilized to alternately put the track/store unit in either a track or store mode of operation during the simulation. The unit is in a track mode of operation during the time either A_1 or A is in a logic one state. During any other time, which is really the update period, the unit is in a store mode.

When the conventional unit step function occurs at $\tau = 0$, the output is shown in Figure 7. This is the situation when the input $e(t)$ is equal to a step function $u(t)$. By applying the machine reference through a scaling

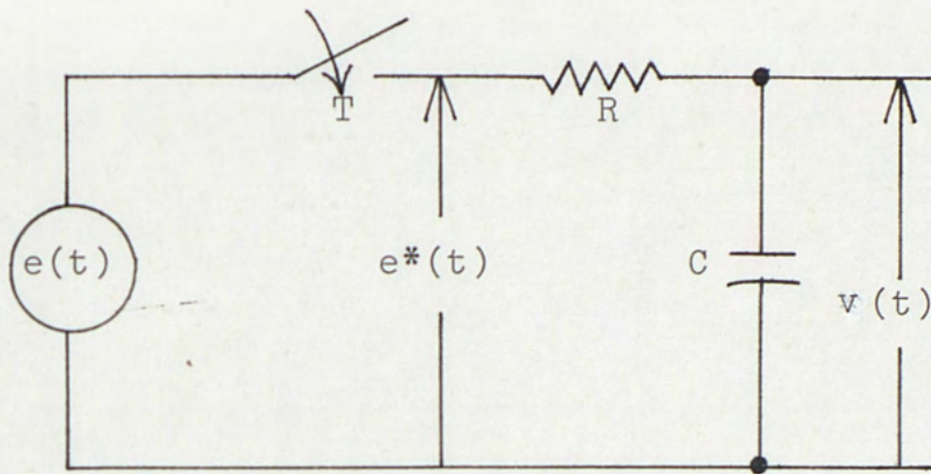


Figure 5. Simple Sampled-Data System

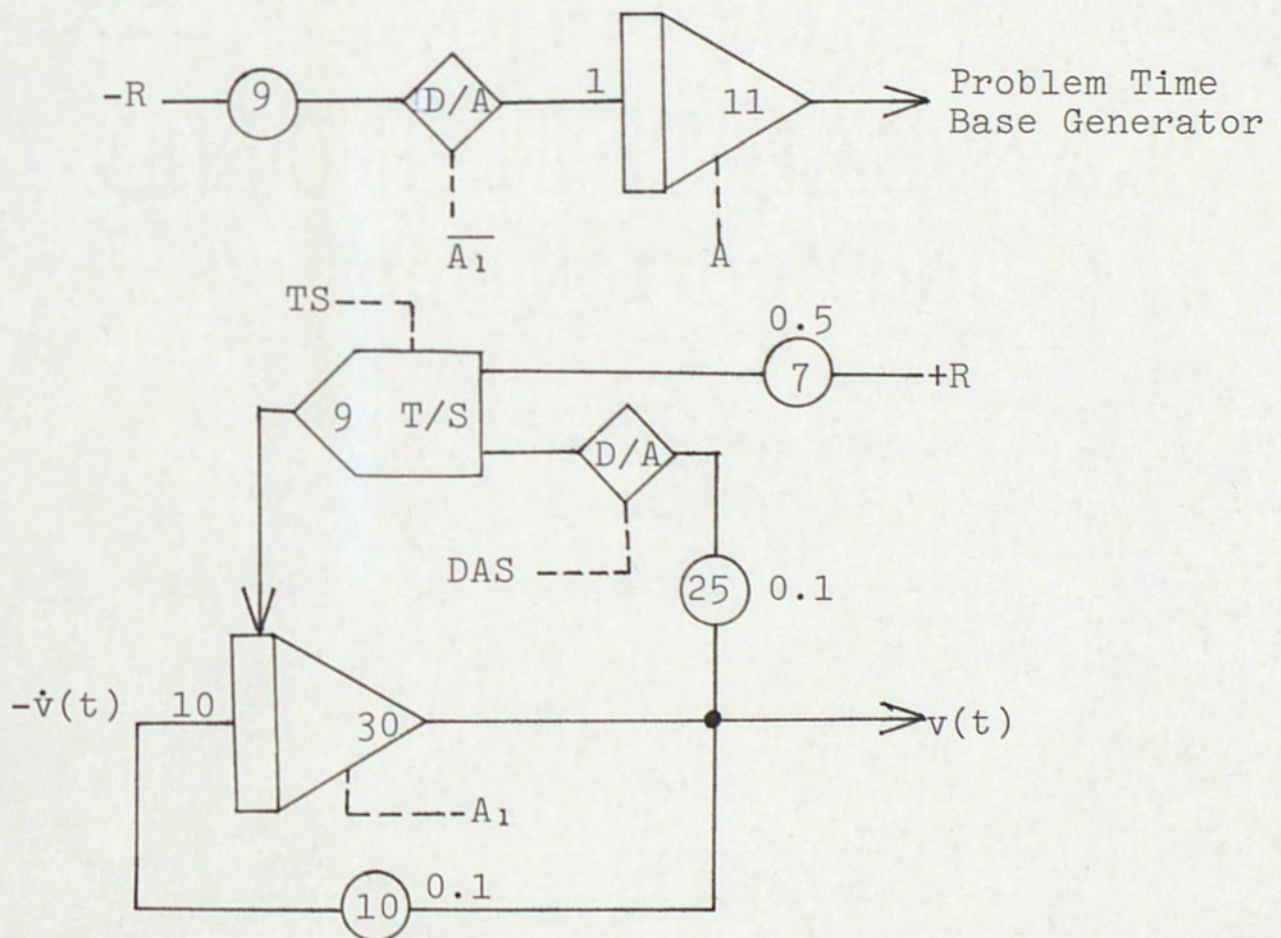


Figure 6. Simulation of System in Figure 5

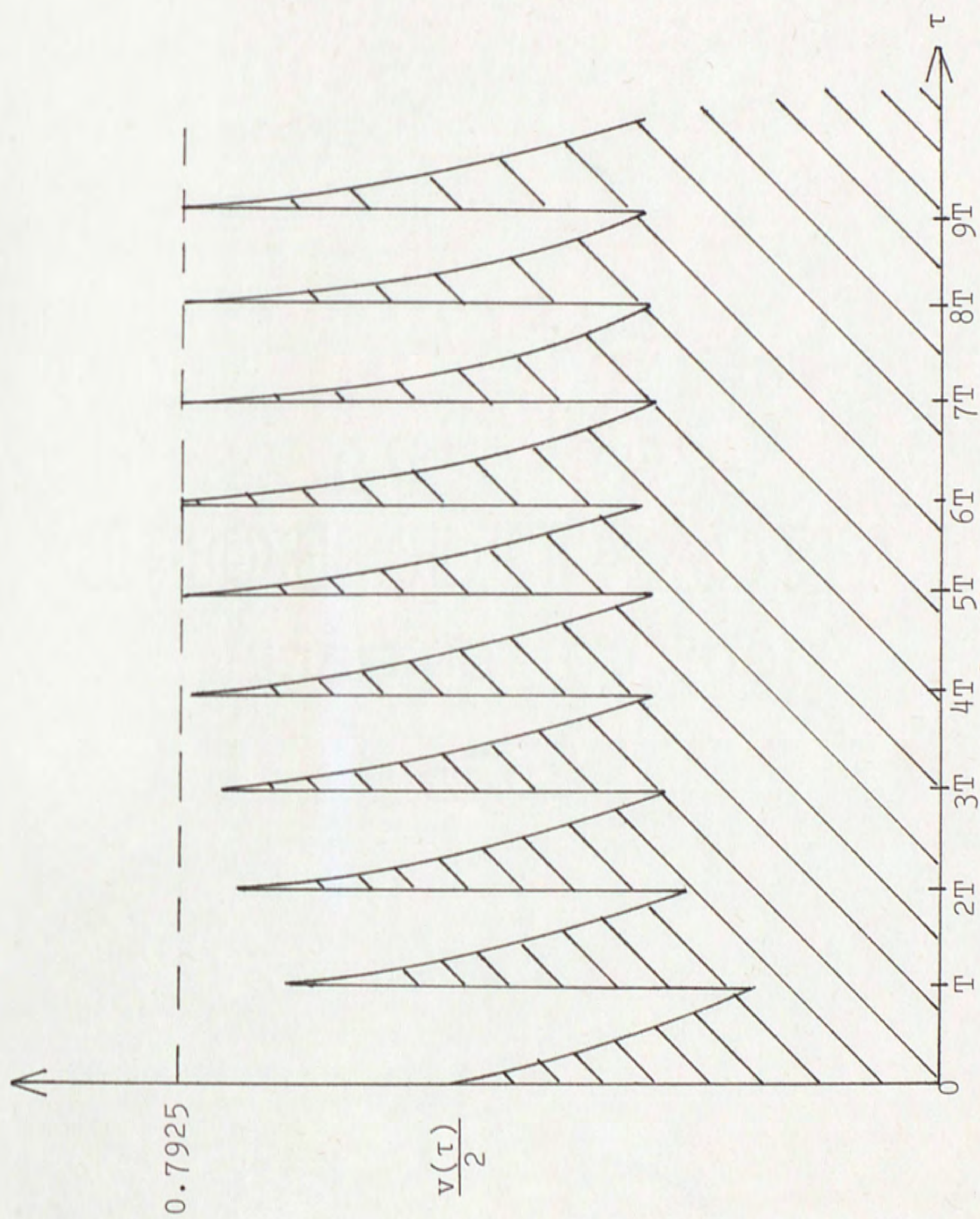


Figure 7. Output of System in Figure 5 When Input Equal Step Function

potentiometer, the scaled variable of $\frac{e(\tau)}{2}$ was generated. The steady value of the results turned out to be 0.7925 which agrees with the results in Kuo's text [8].

EAI 680 Logic Program

A logic microprogram was also implemented on the EAI 680 computer to generate the updating operation. The logic variable and the logic program are shown in Figures 8 and 9. Like the 580 program, this program was started by depressing the PP analog button. Once this button is depressed the interval timer, an integral part of the 680 logic complement, starts to proceed through an A_T (IC), B_T (OPERATE), and C_T (HOLD) mode of operation. Simultaneously, counter zero (CTR 0) is preset (PRE) by way of AND gate 1C. Now, as the timer goes through the A_T , B_T , and C_T states, differentiator zero (DIFF 0) sends a pulse by way of AND gate 2A to CTR 0 every time B_T (OPERATE) goes from a logic "1" to a logic "0". At the same time a pulse is transmitted to the skip input of the interval timer to insure the C_T (HOLD) state is skipped. The conditions described above continue to take place until CTR 0 has received No-1 input pulses from AND gate 2A. No is the value that is loaded into the counter by way of its thumbwheel. When this No-1 count is reached on CTR 0, the output of the counter (Co) goes to a logic one and this stops AND gate 2A from passing the next pulse from the differentiator (DIFF 0). Since the output of AND gate

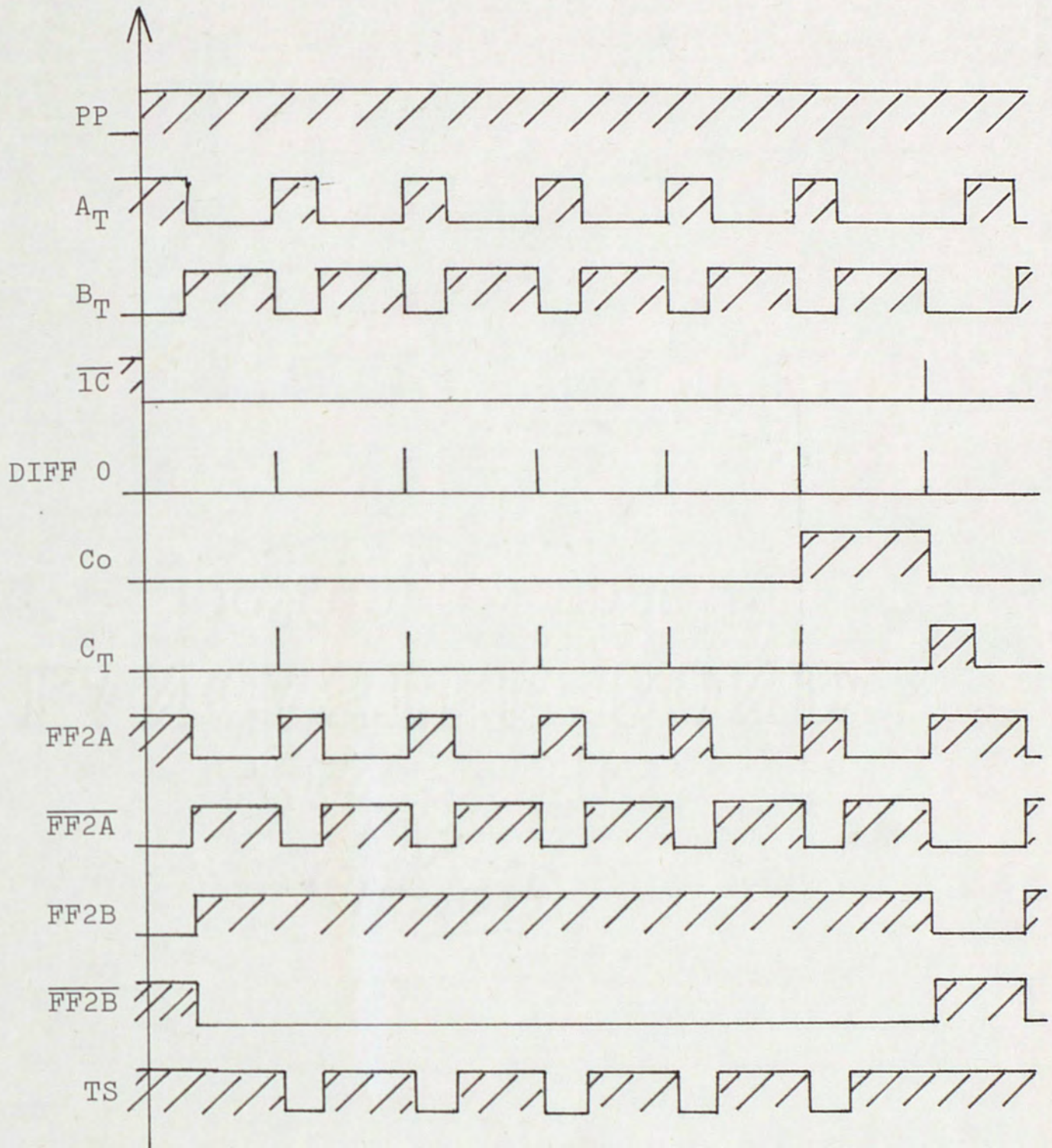


Figure 8. Logic Variables for Up-Date Period (EAI 680)

FF2B O-----OIC Buss

FF2A O-----OOP Buss

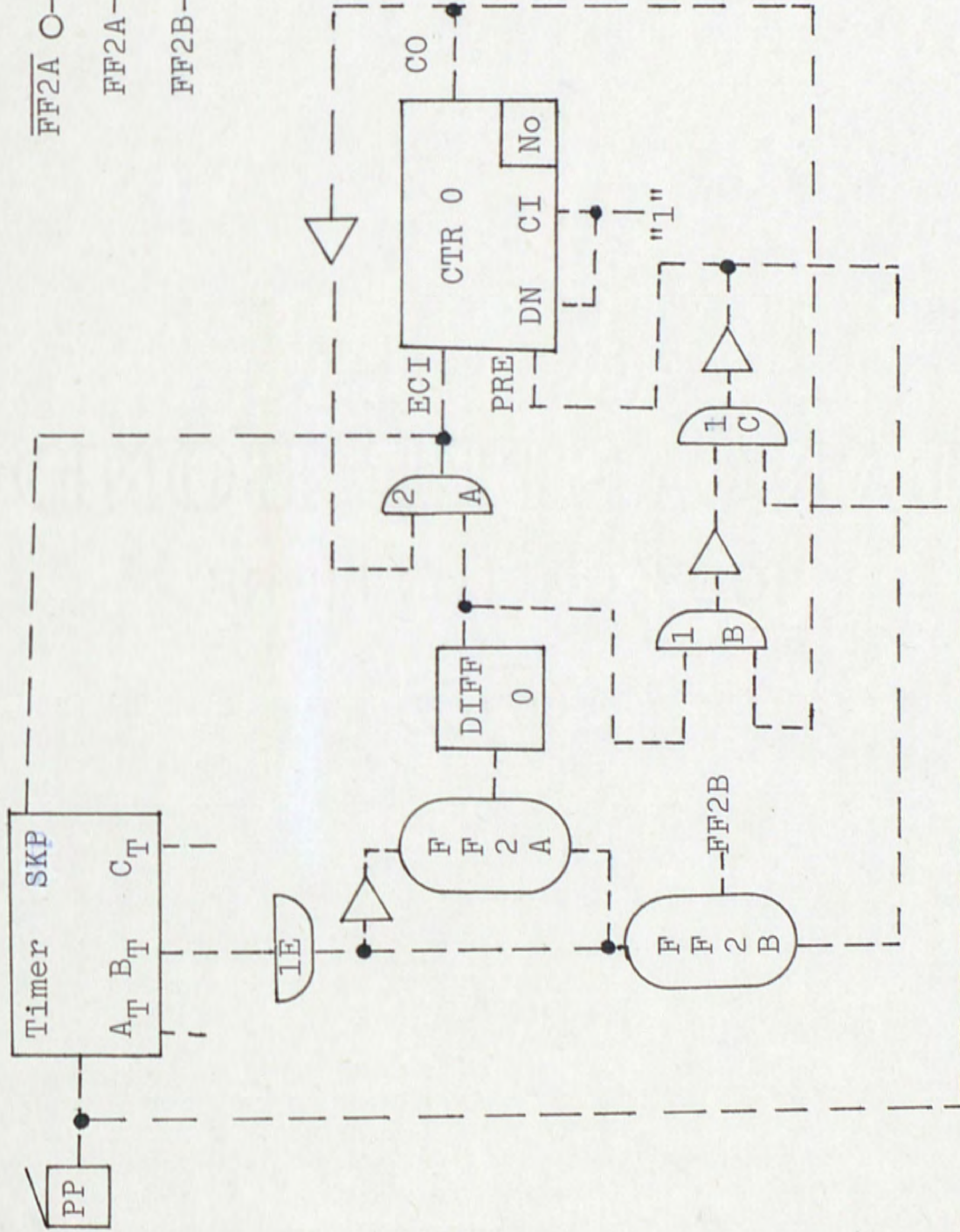


Figure 9. Logic Program for Up-Date Period (EAI 680)

2A is a logic zero and no skip pulse is received by the timer, the C_T state of the computer is not skipped. C_T then recycles the complete operation and the next time DIFF 0 puts out a pulse, CTR 0 is reset through the path of AND gates 1B, 1C, and the preset input of the counter. Now, a complete new pulse count sequence is begun by CTR 0 and the procedure is repeated.

A/D-5 Logic Program

The A/D-5 logic program for implementing the updating process is similar to that of the EAI 680 program in the sense it is built around the interval timer and a counter. Figure 10 shows the generation of the logic valuables while Figure 11 represents the variables themselves. This program is begun using the operation of the RUN button on the control panel. When the computer is put in the RUN mode, clock pulses are automatically delivered to all logic elements and the logic program proceeds. In this mode of operation, the interval timer counts the clock signals and cycles through the A_T , B_T , and C_T states, staying in each one for the number of clock signals set by the interval timer thumbwheels on the control panel. The interval timer states A_T , B_T , and C_T are like the EAI 680 and represents the mode's initial condition, operate, and hold, respectively.

Once the timer starts operation the counter, CTR 50, is cleared (CL) by way of the OR gate 28 and 27 path. As

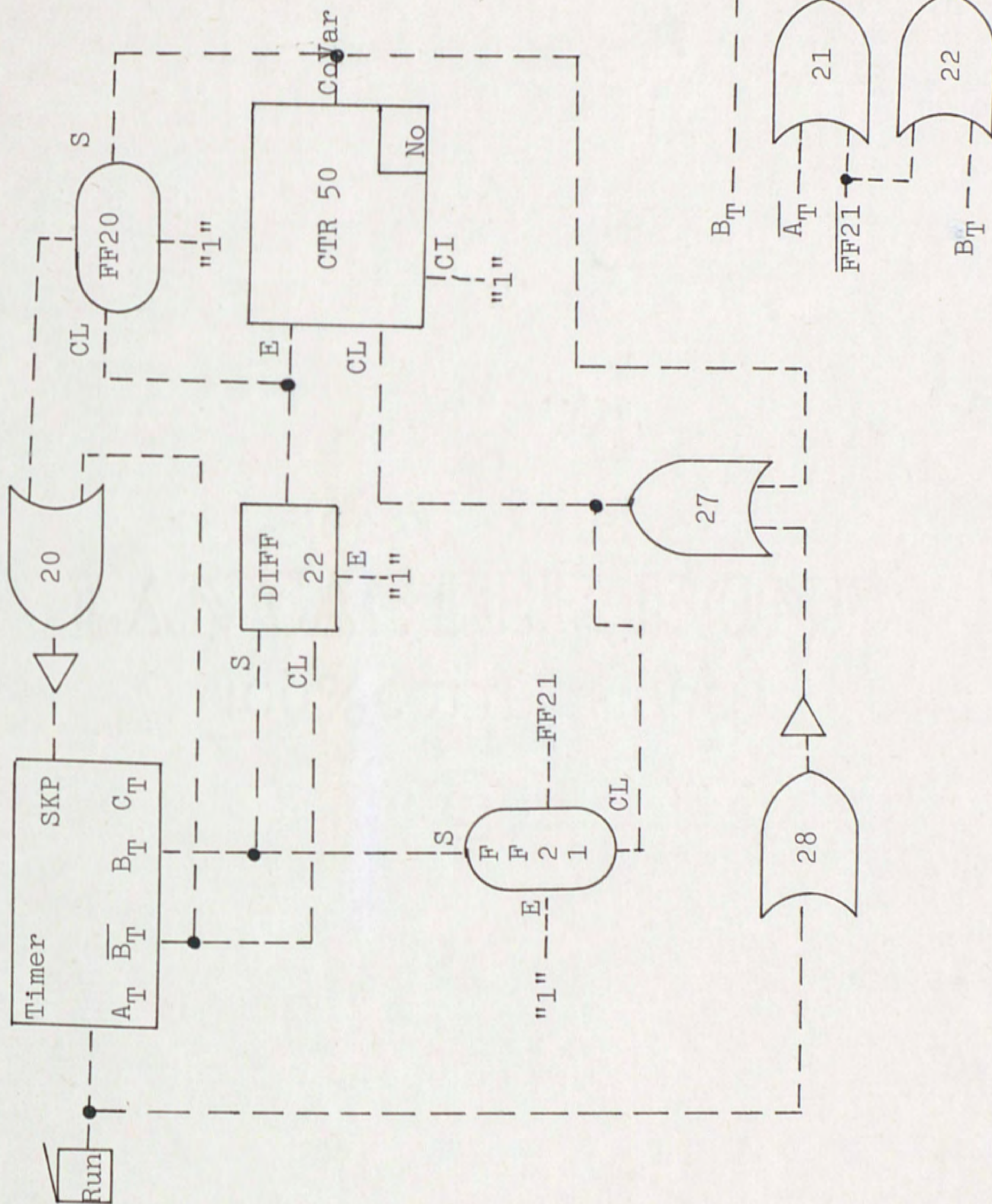


Figure 10. A/D-5 Logic Program for Up-Date Period

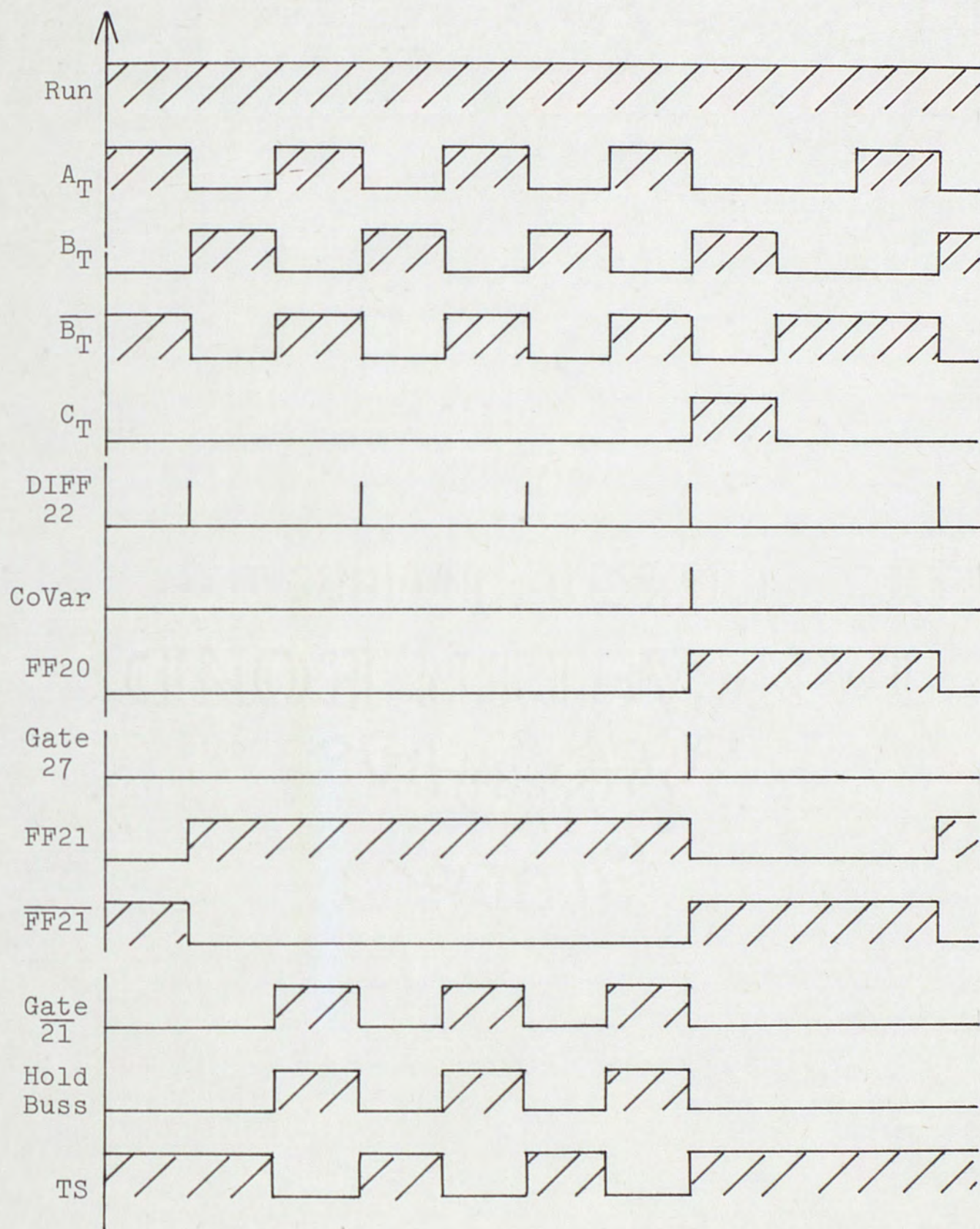


Figure 11. Logic Variables for Up-Date Period (A/D-5)

the timer starts to cycle through its A_T , B_T , and C_T states, differentiator 22 (DIFF 22) sends a pulse to the enable (E) input of CTR 50 every time B_T goes to a logic one. The first time B_T goes to a logic one it also sets (S) flip-flop 21 (FF-21) which will be used to create a hold buss. The complement of B_T ($\overline{B_T}$), flip-flop 20 (FF-20), and gate 20 are used to cause the timer to skip (SKP) the C_T state. This skipping of the C_T state will continue until FF-20 gets a set signal from the output of the CTR 50 (CoVar). CoVar does not go to a logic one until CTR 50 has received No-1 pulses at the enable (E) input from DIFF-22. Here No is a number preset by thumbwheels. When this does occur (CoVar = "1"), FF-20 is set by a pulse out of CTR-50 which is a logic "1" for a microsecond. Once FF-20 is set, it will remain in this state until its CL input is pulsed which occurs when B_T sends a pulse to DIFF 22. The skip input is deactivated from gate 20 once FF-20 goes high. This action allows the appearance of the C_T state. CTR-50 is also cleared once CoVar goes high by way of gate 27. Then after the timer goes back through an IC condition the count process will start once again.

CHAPTER II

SECOND ORDER SYSTEM SIMULATION

Now that the big problem of generating an impulse function has been overcome, a standard second-order Type I system, as explained in Appendix A, will become the basic model for the investigation. Five different versions of this basic system will be simulated. These are: a continuous system, a sampled-data system with no hold device, a sampled-data system with a zero-order hold, a sampled-data system with a first-order hold, and a sampled-data system with a second-order hold. The last two configurations were not implemented on the EAI 680 or the A/D-5 due to a hardware limitation.

Appendices B and C in conjunction give some insight to the stability problems that can be encountered in the sampler but no hold and also the zero-order hold systems. Appendix B shows how the transfer function of each hold device may be generated and Appendix C indicates how the information can lead to a stability region determination.

EAI 580 Analog Programs

Since most of the logic variables are explained in the section on the EAI 580 Logic Program and shown in

Figures 4(a) and (b), no detailed explanation will be given here. If the sampler and hold circuit are removed in Figure 1 to make a continuous system, the simulation program is shown in Figure 12. The impulse function simulation microprogram of Figure 6 is included, but is not mandatory. Here it is included from an ease of hardware implementation point of view. With a step function as the system input, the output response is the typical damped sine wave of a second-order system. This output response will be shown later in a comparison of different system outputs.

The next version of the system was a sampled-data system, but with no hold circuit included. The analog simulation of this system is shown in Figure 13. Maximum sample period of this system and still remain stable as shown in Appendix C is 4.32 seconds. Signals TS and DAS have the purposes, as described in the section on impulse function simulation. Figure 14 shows the response of the system to a step function, $r(t) = u(t)$. Also shown in Figure 14 are two of the other variables of the system, $\frac{e(\tau)}{2}$, the output of AMP 34 and $-\frac{\dot{e}(\tau)}{2}$, the output from INT 30. A comparison of the continuous system output response and that of the sampled-data system with no hold is shown in Figure 15.

Now a zero-order hold is inserted in the forward path of the system. The analog simulation program of a sampled-data system with a zero-order hold is shown in Figure 16.

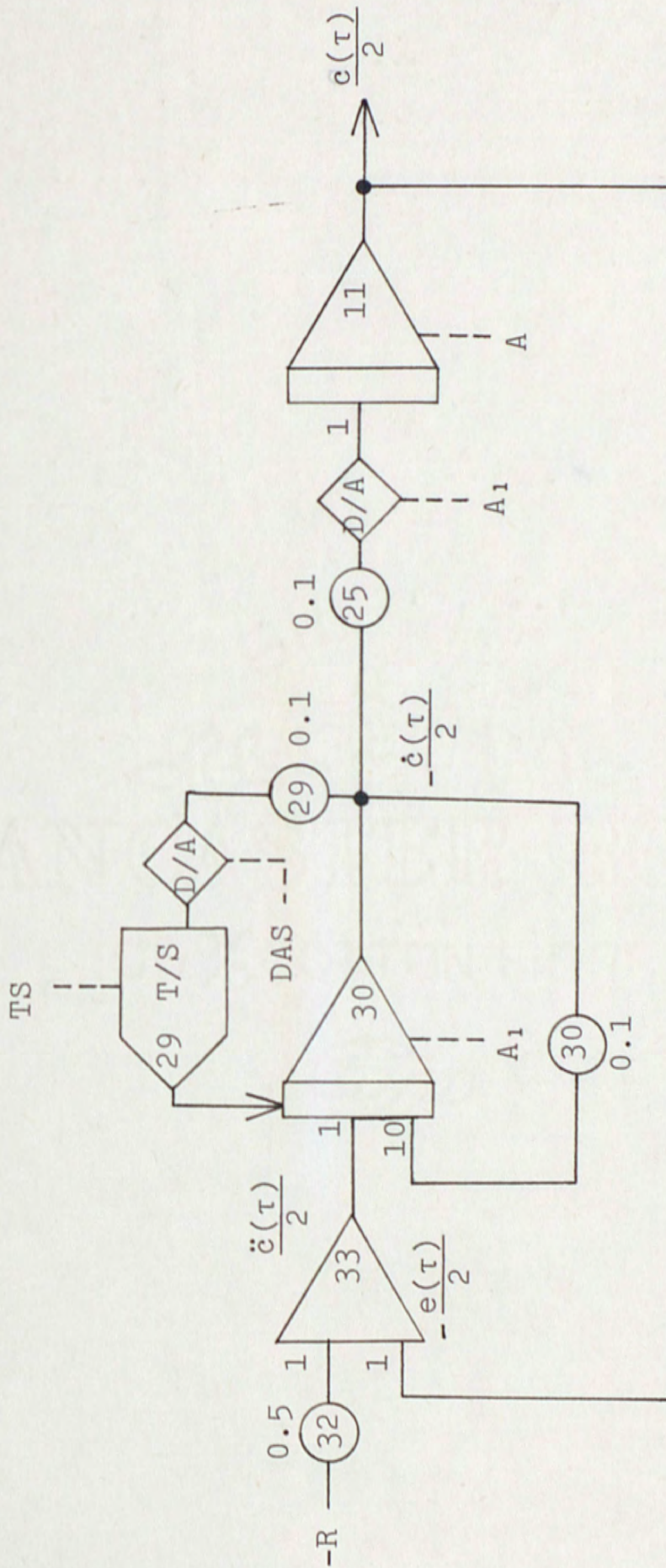


Figure 12. Second Order System With No Sampler

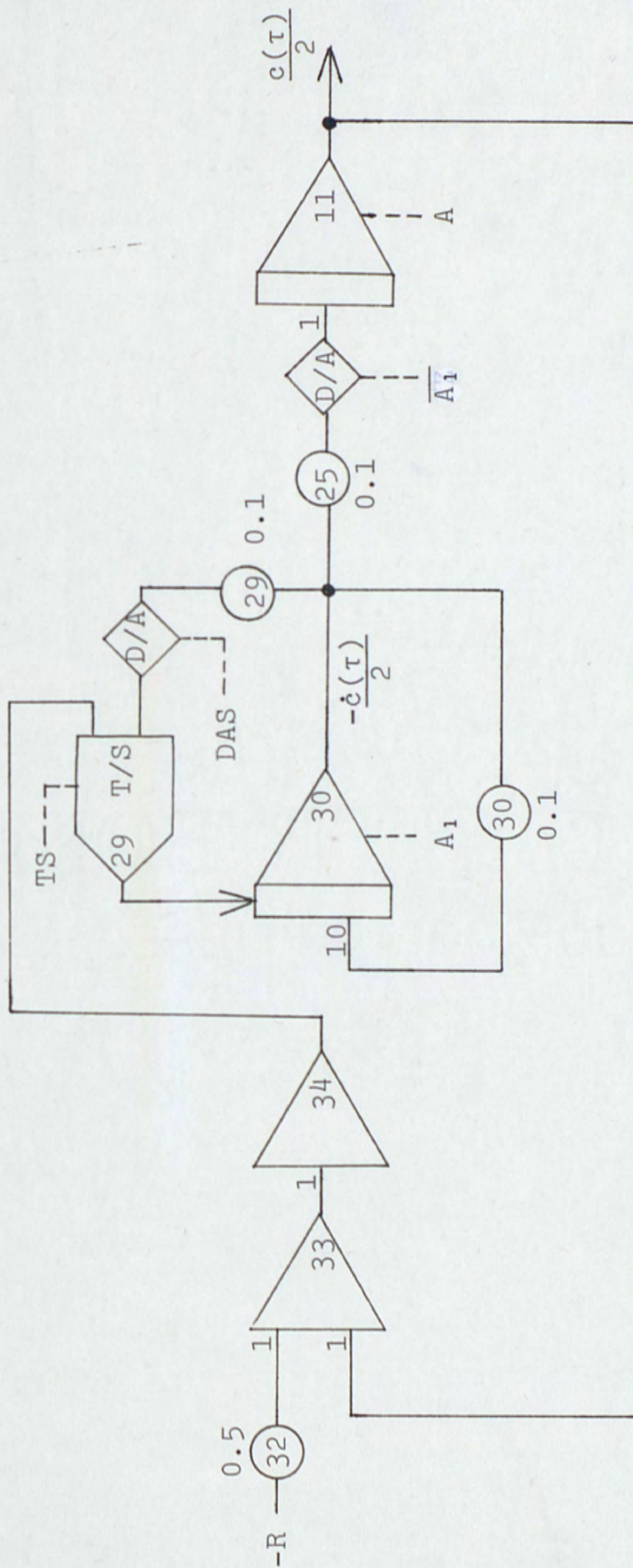


Figure 13. Second-Order System With Sampler But No Hold Circuit (EAI 580)

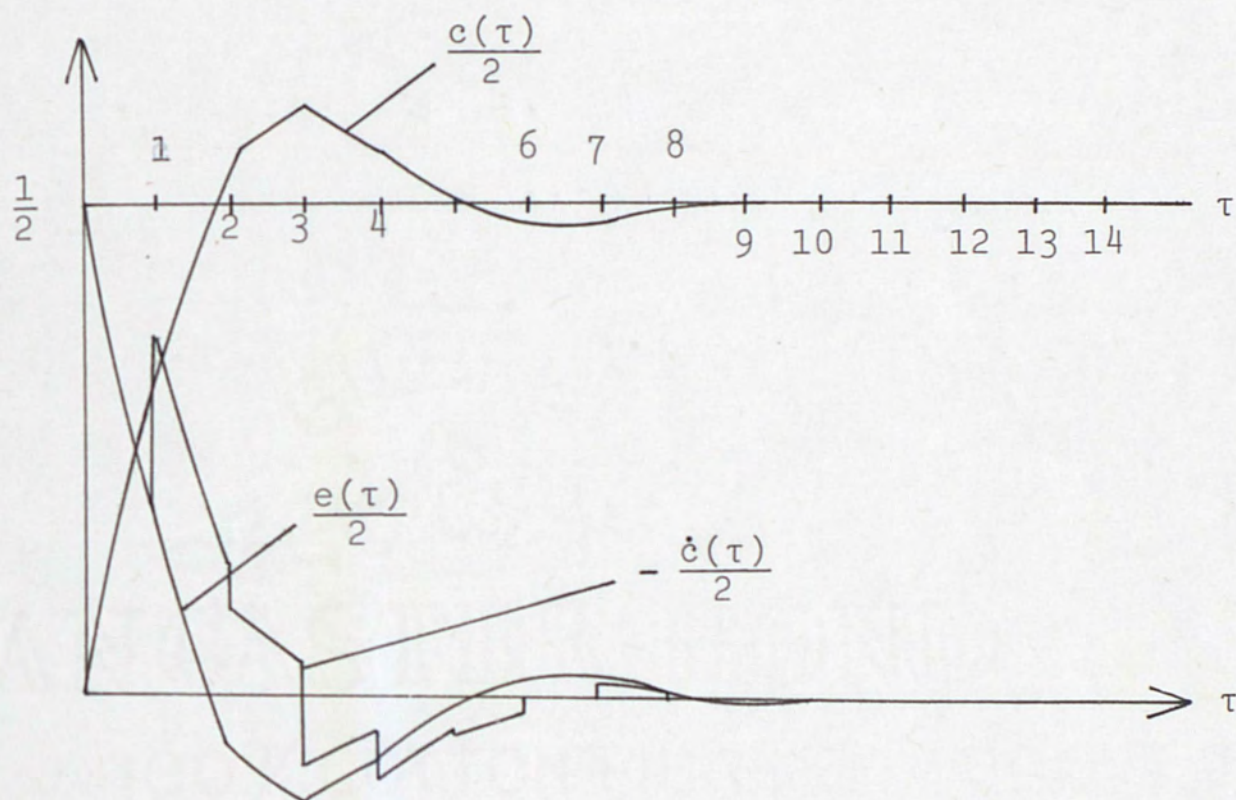


Figure 14. Variables of the Simulation in Figure 13

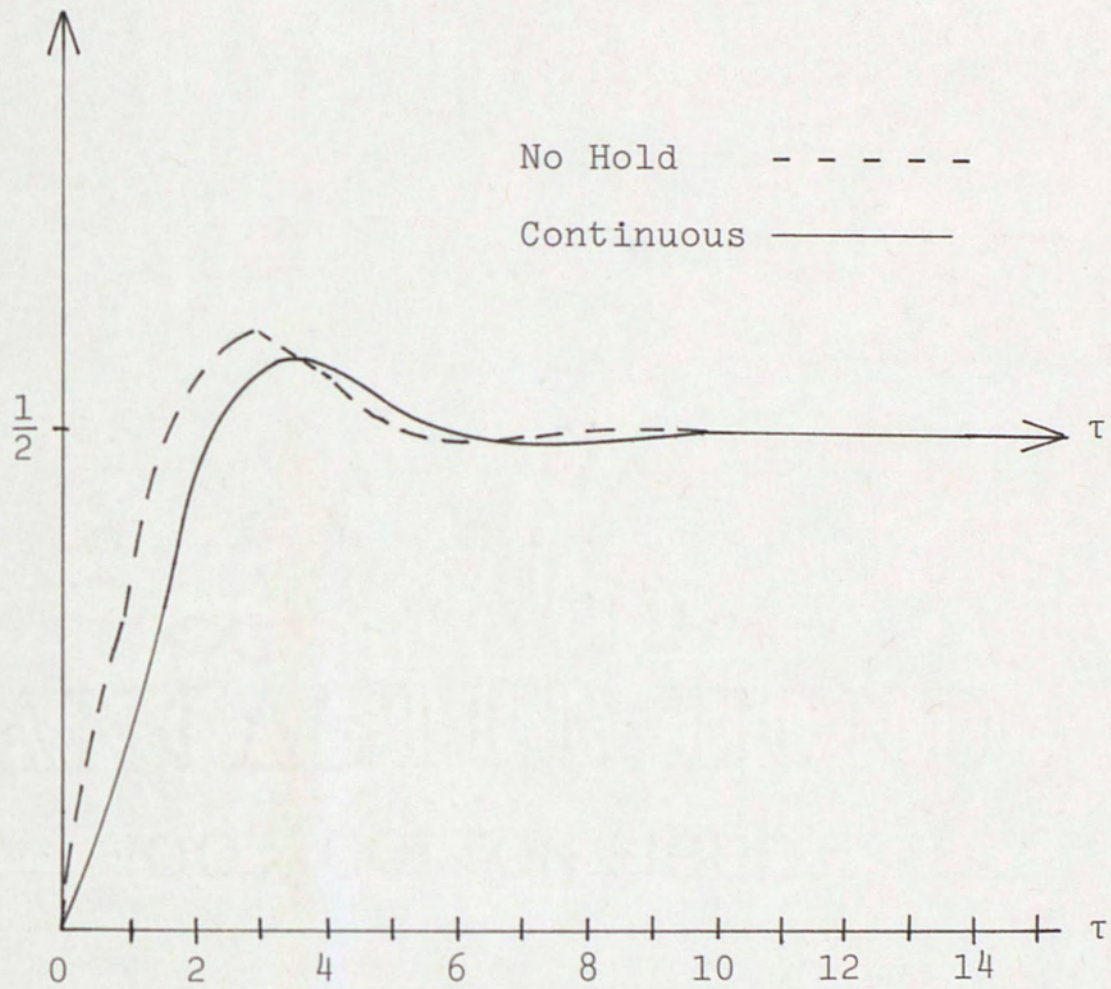


Figure 15. Comparison of Output Responses of Continuous System and No Hold System.

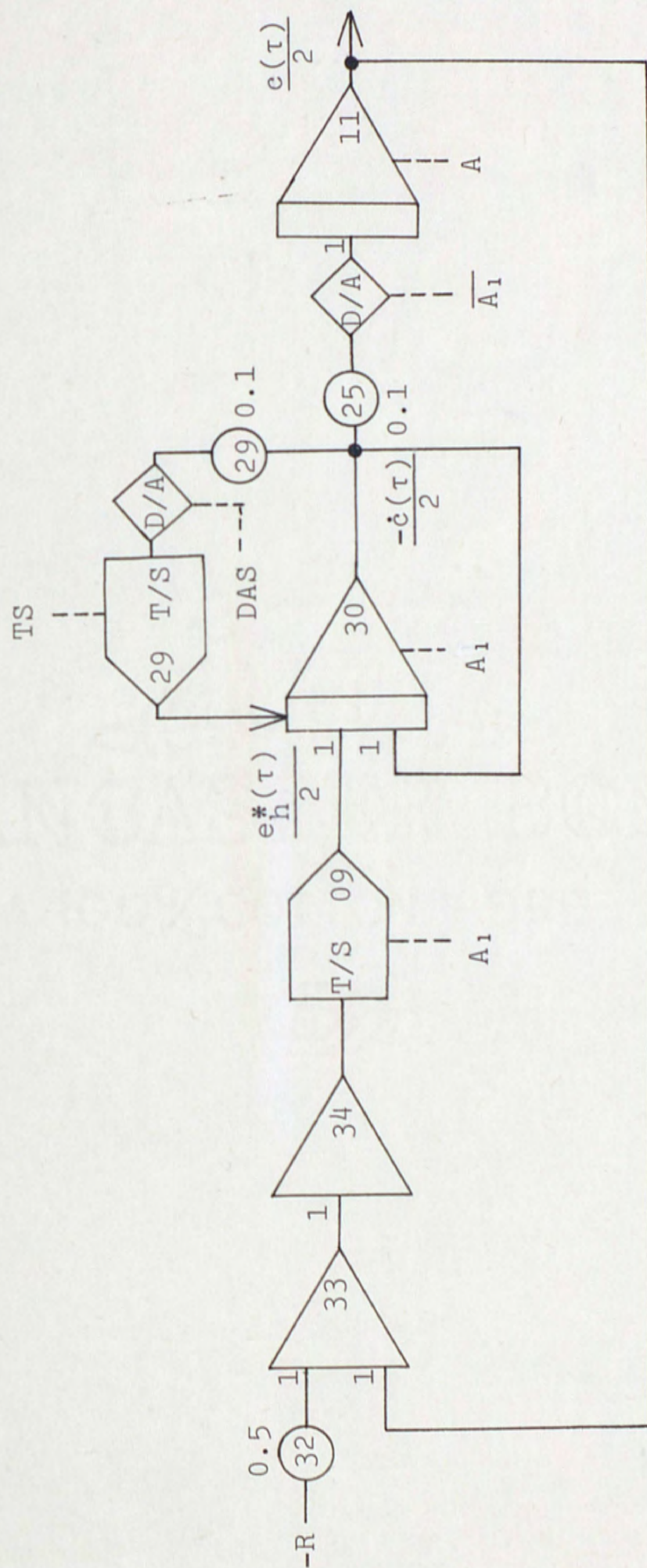


Figure 16. Simulation of Second-Order System With Zero-Order Hold (EAI 580)

Here track/store 09 represents the zero-order hold device. Its purpose is to sample (track) the continuous error signal and then hold (store) it until the next sample period when it will sample a new value of the error signal. Track/store 09 tracks the error signal when the simulation is in operate and stores during the update period.

This operation is shown in Figure 17 which shows the output of T/S 09, zero-order hold output, and the output response, c. The error response for a step function input and the derivative of the output are represented in Figure 18.

Figure 19 shows the simulation program when the hold circuit of Figure 1 has been increased to a first-order hold. It must be noted that the two logic variables of TSA and TSB must be generated in addition to those of Figure 4(a). These signals are used to perform the necessary update operations to provide the first-order hold output function and their generations are shown in Figure 20. Equation (9) from Appendix B is shown below

$$e_k(t) = e(kT) + K \left[\frac{e(kT) - e(k-1)T}{T} \right] (t - kT)$$

which indicates that the first-order hold output function is really a function of the previous values of the error signal, e, at the sample instances. The parameter K, which is represented by potentiometer 33 in the simulation, controls the order of the hold. If it is one, the hold is of

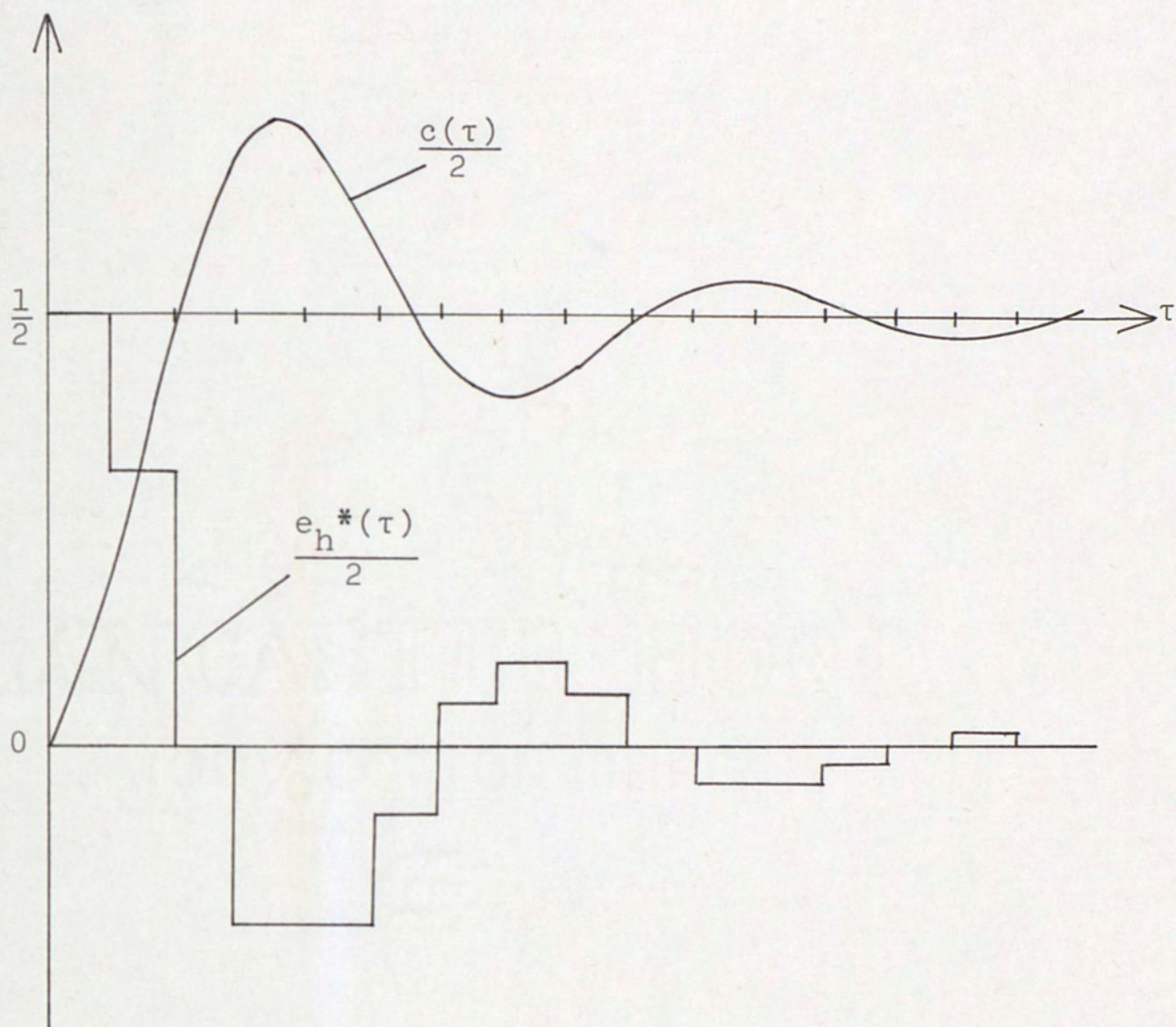


Figure 17. Zero-Order Hold Output and System Output Responses for Figure 16.

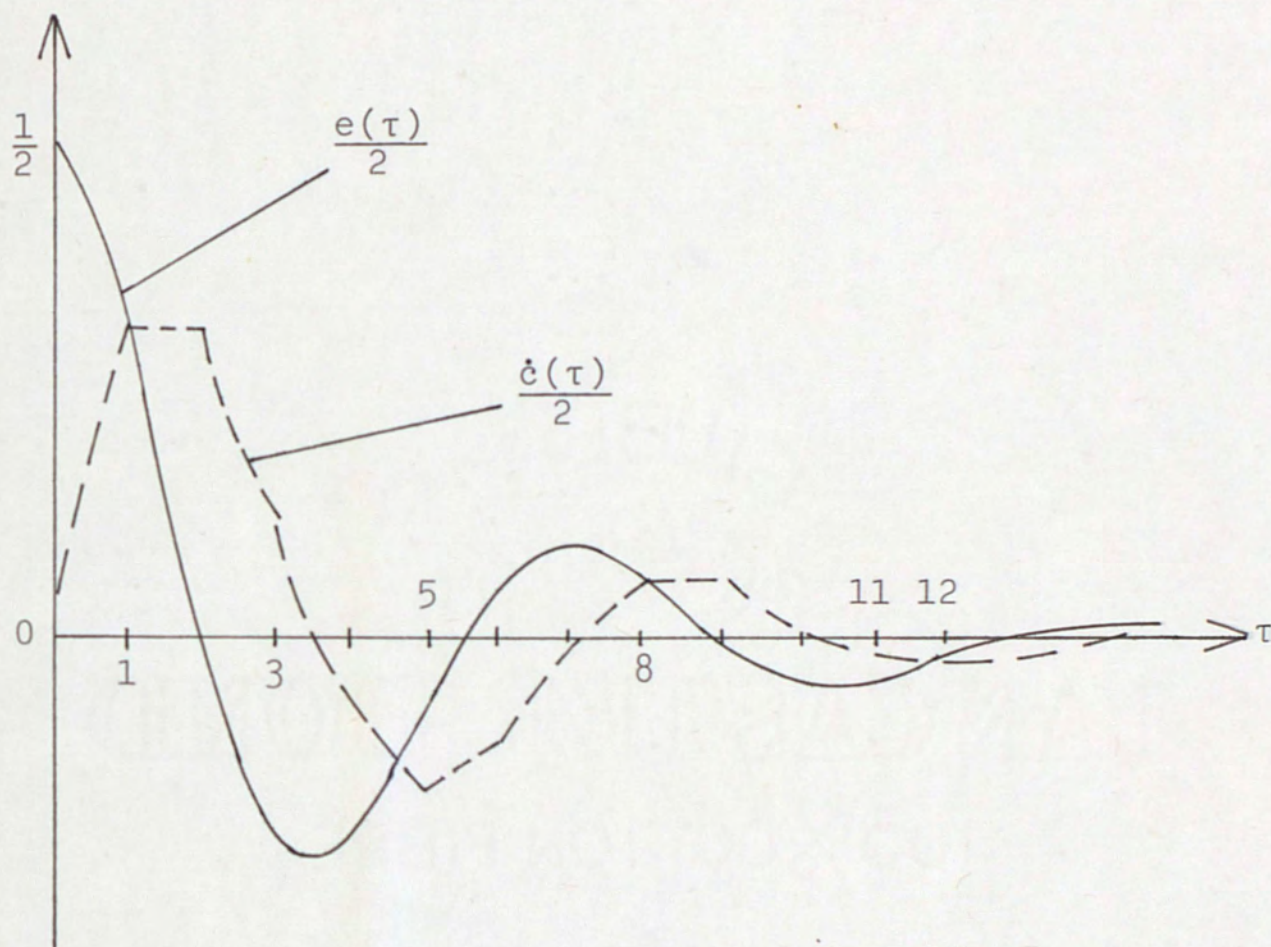


Figure 18. Two More Variables From the Simulation Shown in Figure 16.

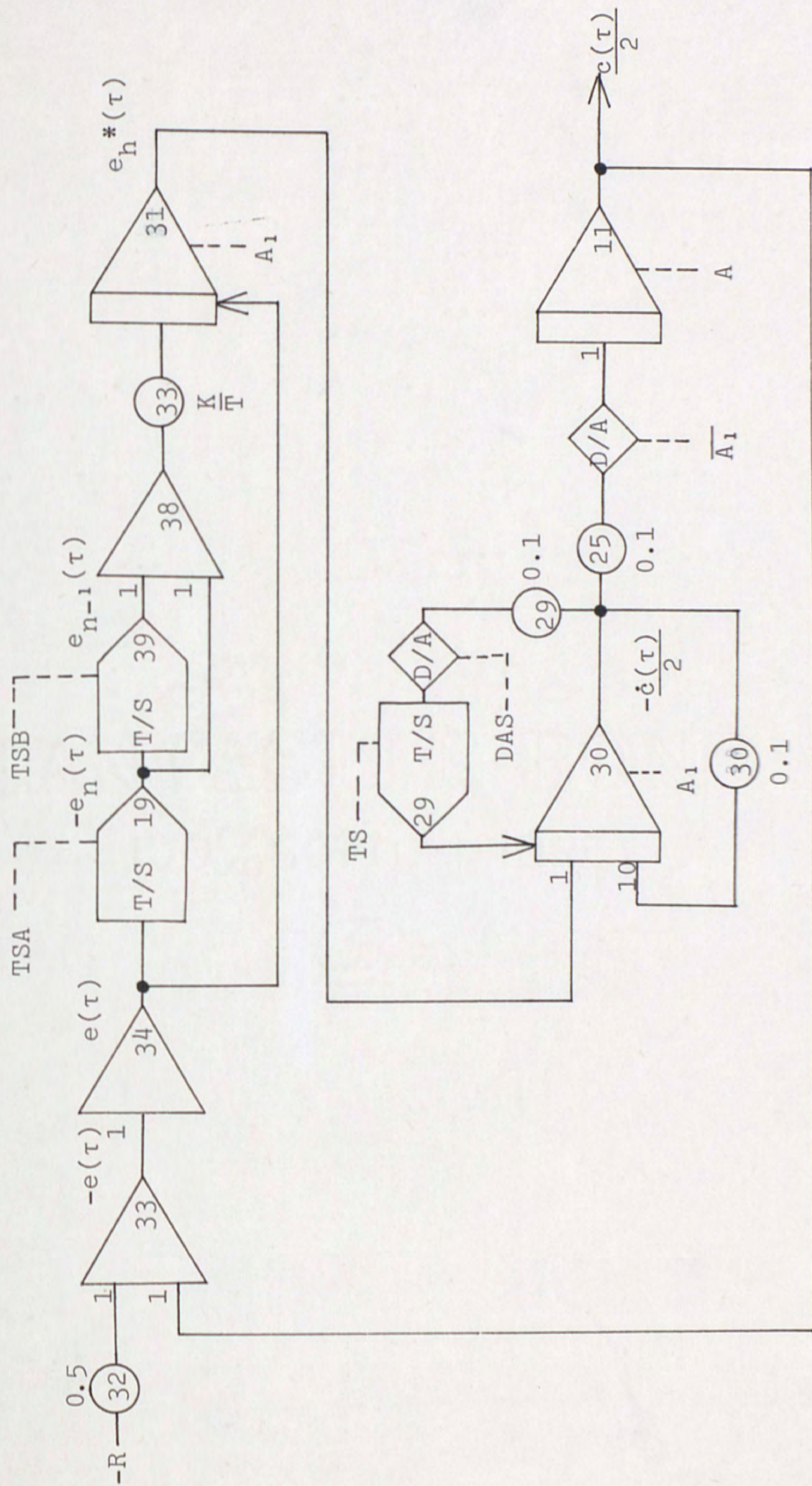


Figure 19. Second-Order System With First-Order Hold

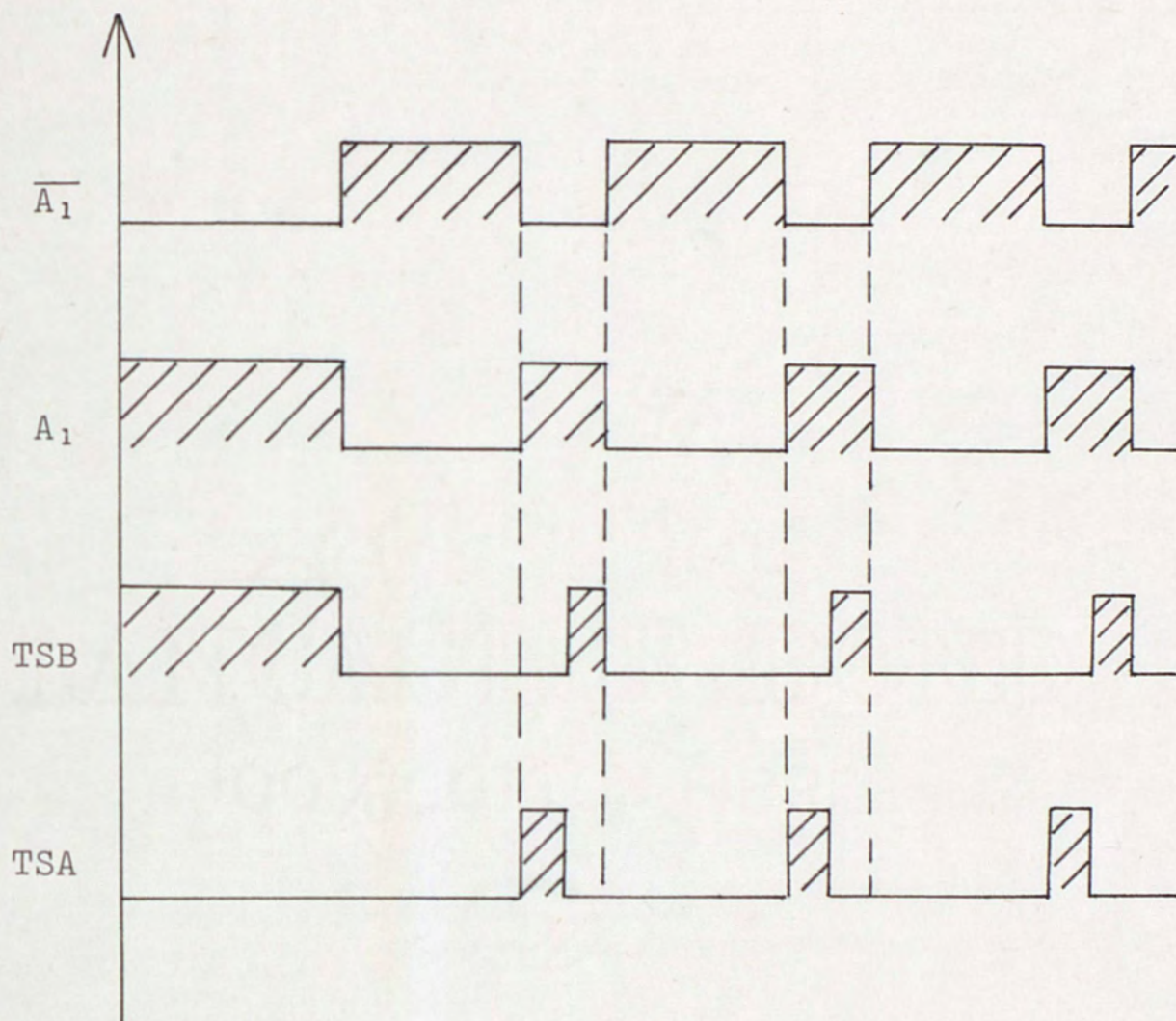


Figure 20. Additional Logic Control Variables Needed for First-Order Hold Microprogram.

the first order and it could make the simulation into various fractional order hold devices. Without an update period in this particular system with a first-order hold, it will be shown later that twice as much hardware would be required along with two additional D/A switches.

Some of the variables throughout the first-order hold simulation are represented in Figure 21, while Figure 22 shows a comparison of the outputs of a continuous, a sampled but no-hold, a zero-order hold, and finally a first-order hold system.

If another control logic signal TSC is generated along with TSA and TSB for the first-order hold simulation, the second-order hold can be programmed as shown in Figure 23. The logic control variable TSC must appear after TSB but before the next sample period, \bar{A} , in the timing diagram.

The program for the generation of the second-order hold can be obtained in the following manner. Assume a form

$$f(t) = a_0 + a_1(t - nT) + a_2(t - nT)^2 \quad (10)$$

this is a truncated form of the Taylor series

$$f(t) = f(t_0) + f'(t_0)(t - t_0) + \frac{f''(t_0)}{2!}(t - t_0)^2 + \dots \quad (11)$$

$$f(nt) \triangleq f_n = a_0 \quad (12)$$

$$f_{n-1} - f_n = -Ta_1 + T^2a_2 \quad (13)$$

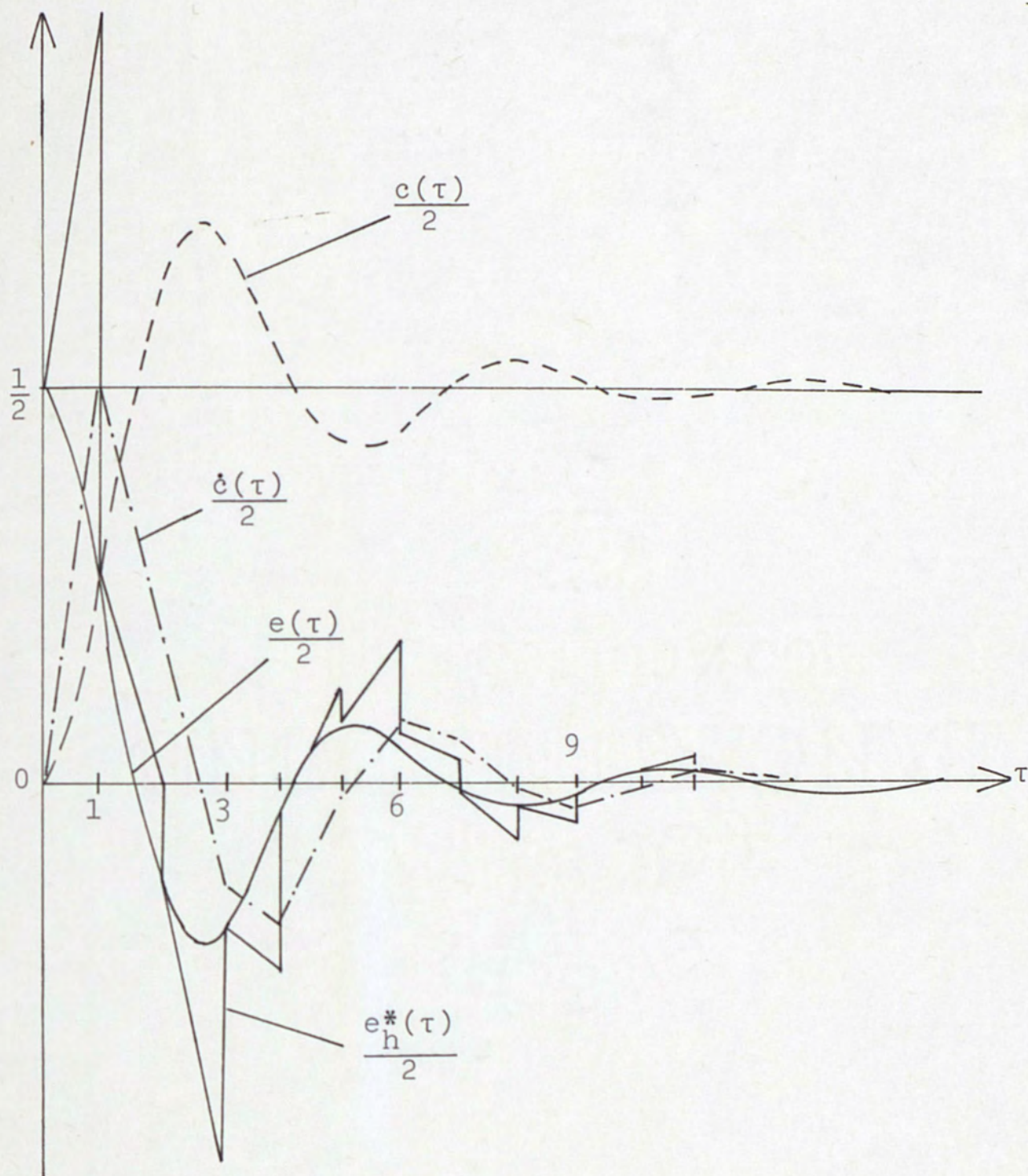


Figure 21. System Variables for the Simulation Shown in Figure 19.

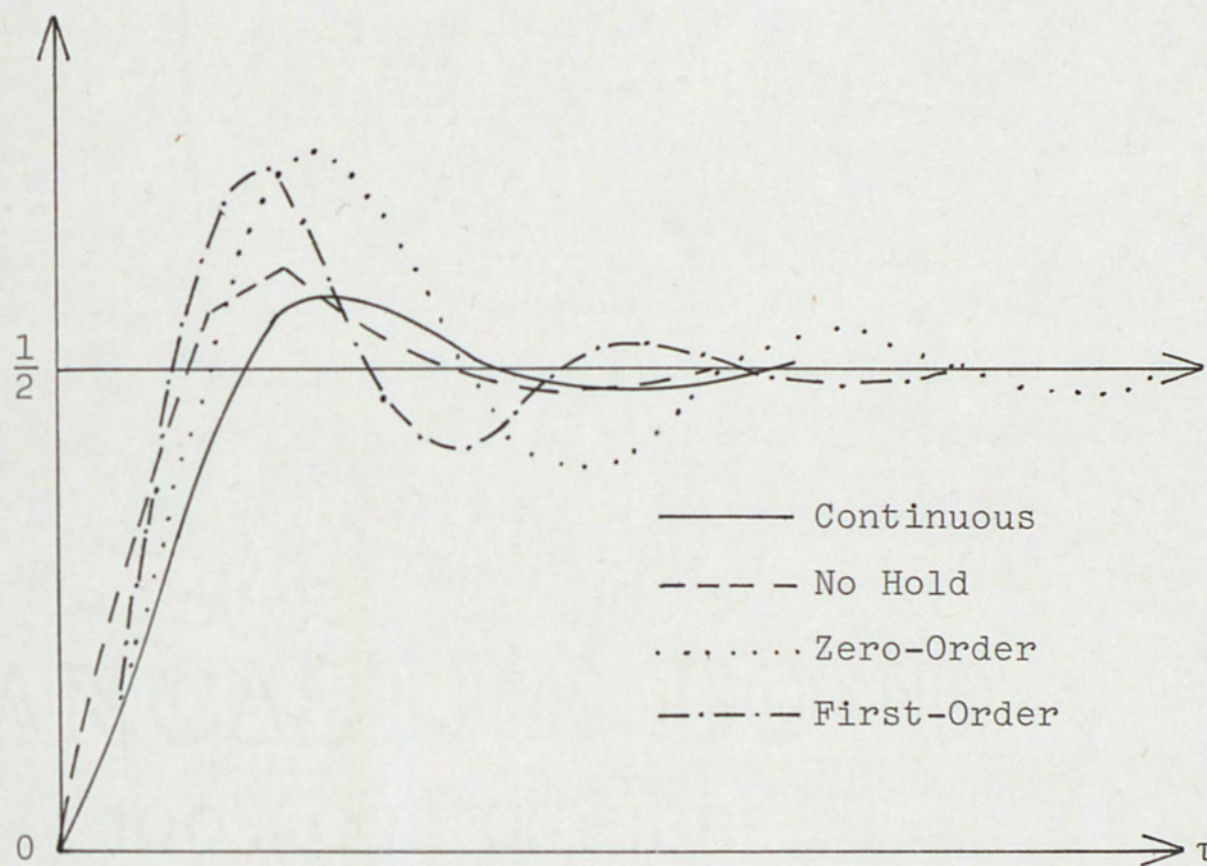


Figure 22. Output Comparison for Four Different Systems

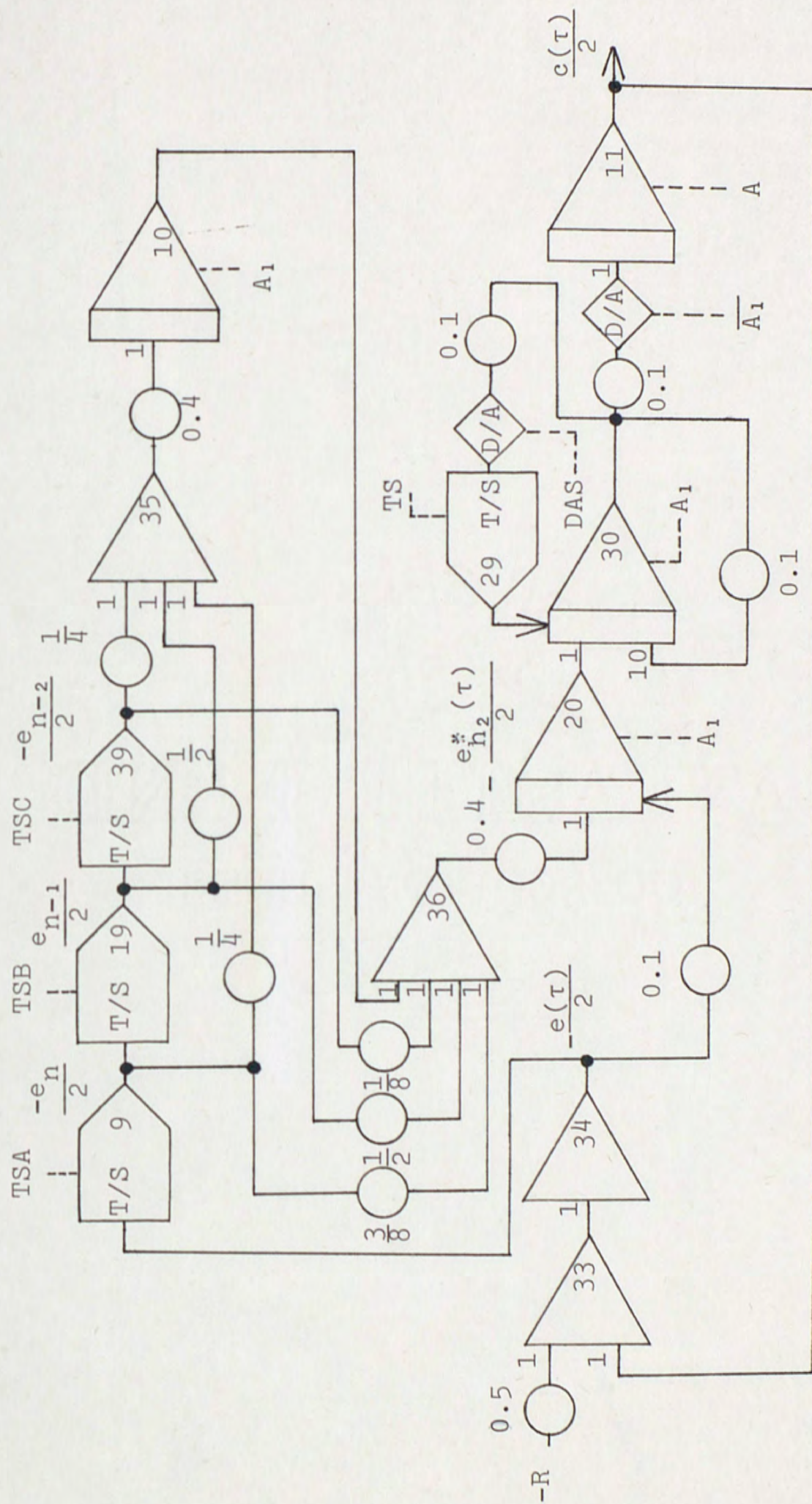


Figure 23. Second-Order System With Second-Order Hold

$$f_{n-2} - f_n = -2Ta_1 + 4T^2a_2 \quad (14)$$

Multiply Equation (13) by 2 and then subtract Equation (14) to obtain

$$2f_{n-1} - f_n - f_{n-2} = -2T^2a_2 \quad (15)$$

$$a_2 = \frac{1}{2T^2} [f_n + f_{n-2} - 2f_{n-1}] \quad (16)$$

Substitute Equation (16) into (13)

$$a_1 = \frac{1}{T} \left[\frac{3}{2}f_n - 2f_{n-1} + \frac{1}{2}f_{n-2} \right] \quad (17)$$

and finally

$$\begin{aligned} f(t) = & f_n + \left[\frac{3}{2}f_n - 2f_{n-1} + \frac{1}{2}f_{n-2} \right] \left[\frac{t-nT}{T} \right] \\ & + \left[\frac{1}{2}f_n - f_{n-1} + \frac{1}{2}f_{n-2} \right] \left[\frac{t-nT}{T} \right]^2 \end{aligned} \quad (18)$$

Equation (18) is modified to put in programmable form and the result is

$$\begin{aligned} f(t) = & f_n + \frac{4}{T} \left[\frac{3}{8}f_n - \frac{1}{2}f_{n-1} + \frac{1}{8}f_{n-2} \right] [t-nT] \\ & + \frac{4}{T} \left[\frac{1}{4}f_n - \frac{1}{2}f_{n-1} + \frac{1}{4}f_{n-2} \right] \left[\frac{(t-nT)^2}{2} \right] \end{aligned} \quad (19)$$

The above equation was programmed to obtain the second-order hold portion of Figure 23.

Figure 24 indicates some of the various variables throughout the simulation. The valuable e_{n-1} which is the output of T/S 19 has the same shape as the variable e_n except it is delayed one sample period (T) in time. The

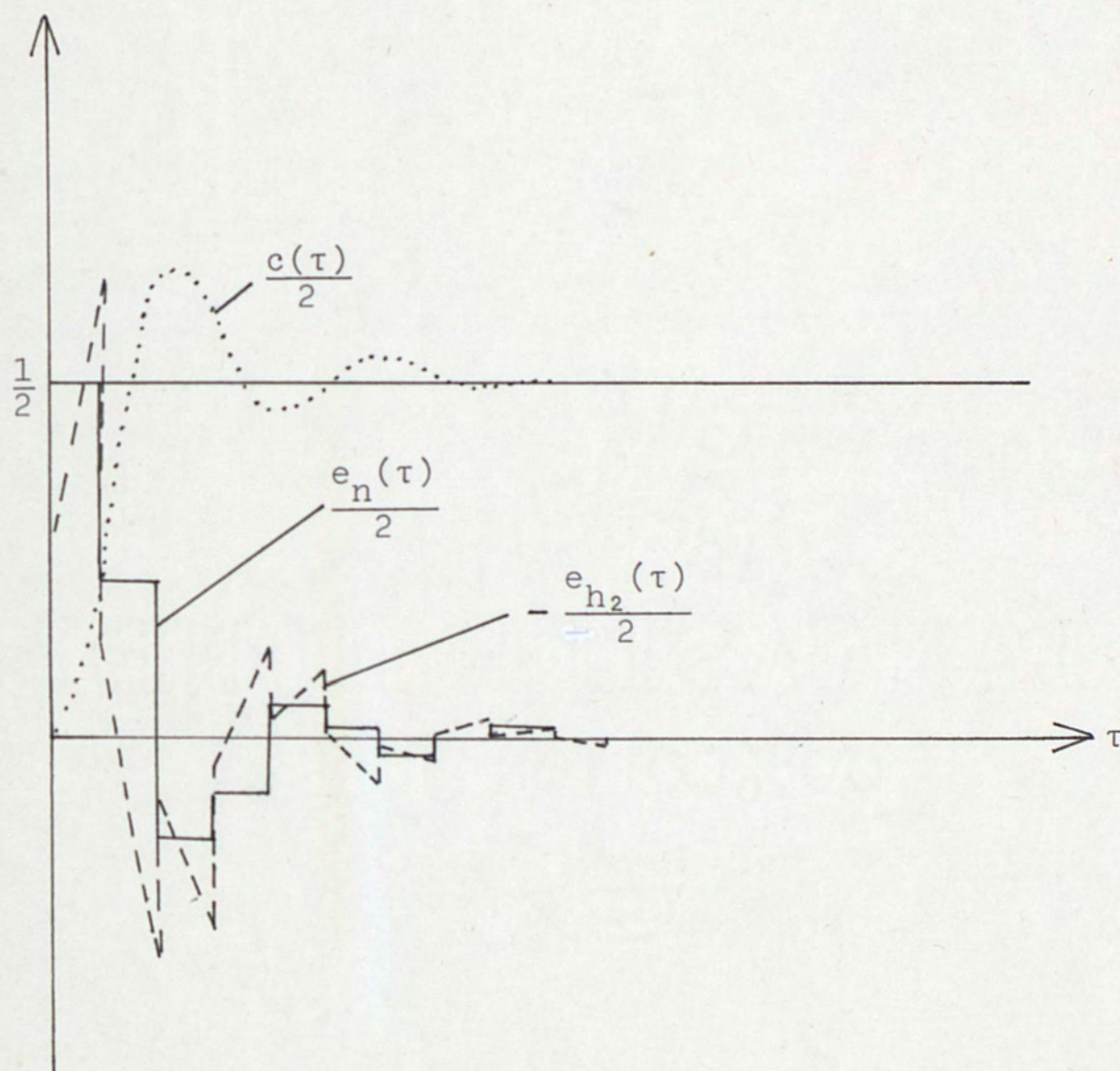


Figure 24. Three System Variables of Simulation Shown in Figure 23.

same relationship is true between e_{n-1} and e_{n-2} which is the output of T/S 39. A comparison of the output responses of some of the systems used in the study is shown in Figure 25. Each output response is the result of a step function on the input to the system.

EAI 680 Analog Programs

Once the EAI 580 analog portion of the study had been accomplished, the majority of the simulation techniques were proven. Since the two computers have different types of integrators, the integrator mode control was the next item to be investigated. As previously stated, the integrators in the EAI 680 are of the three Electronic Mode Control (EMC) variety. This means that through the use of two logic control variables (IC and OPERATE) three distinct states may be obtained for each integrator (IC, OPERATE, and HOLD). The control matrix for each operation is shown below.

Logic Control Variables	Mode Control		
	IC	OP	HOLD
IC	1	0	0
OPERATE	0	1	0

The logic control variables that were used to exercise control over these integrators are shown in Figure 8.

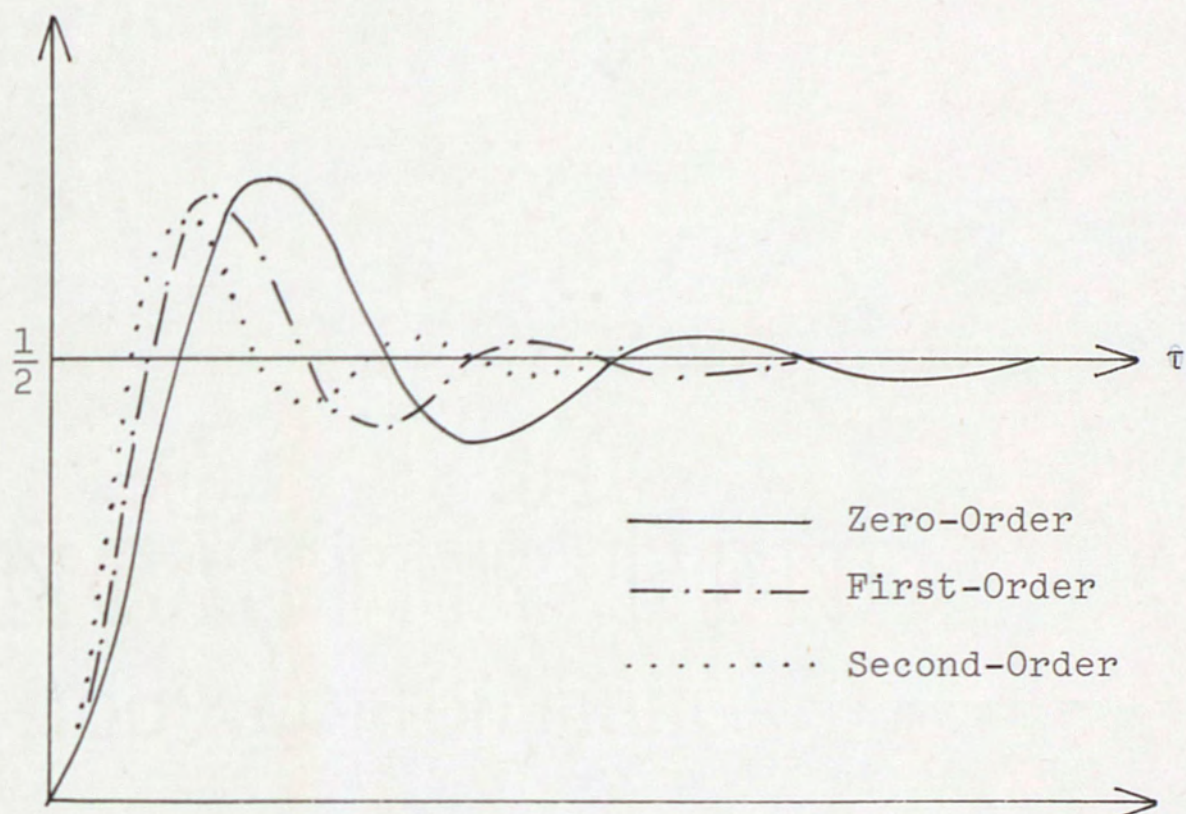


Figure 25. Comparison of the Output Responses of a Second-Order System With Various Order Hold Devices.

As in the EAI 580 program, a problem time base generator was required which would put the problem in sequences of OPERATE, HOLD, OPERATE, HOLD, etc. states as depicted in Figure 26. Integrator 5 in Figure 26 accomplished this purpose by utilizing the newly created IC and OPERATE busses on the mode control inputs. Also, the EAI 680 version of the second-order continuous system is shown in Figure 26. As can be seen, the impulse function microprogram is again included.

The sampled-data system, but with no hold simulation, is shown in Figure 27 and it is the same as the EAI 580 simulation except for the integrator mode control variables. The same statement can be made about the system with a zero-order hold in the loop. Since this microprogram is basically the same as that shown in Figure 16 no microprogram or system variables will be presented in this section. The output responses of these configurations are the same as those on the EAI 580 and are shown in Figure 22.

A/D-5 Analog Program

The analog programs on the A/D-5 computer are almost identical to the ones on the EAI 680. Here, again, an integrator mode control variation is encountered. On this computer the two mode controls available for logic control variables are the OPERATE and HOLD busses. The control

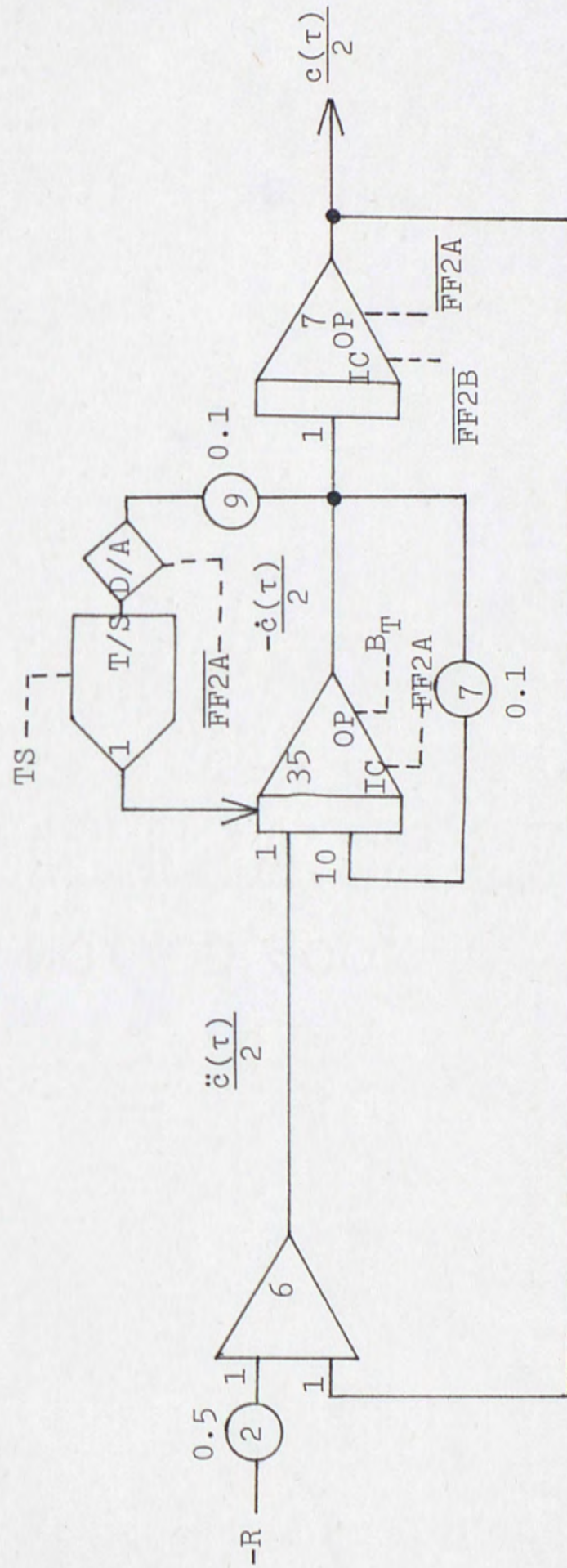
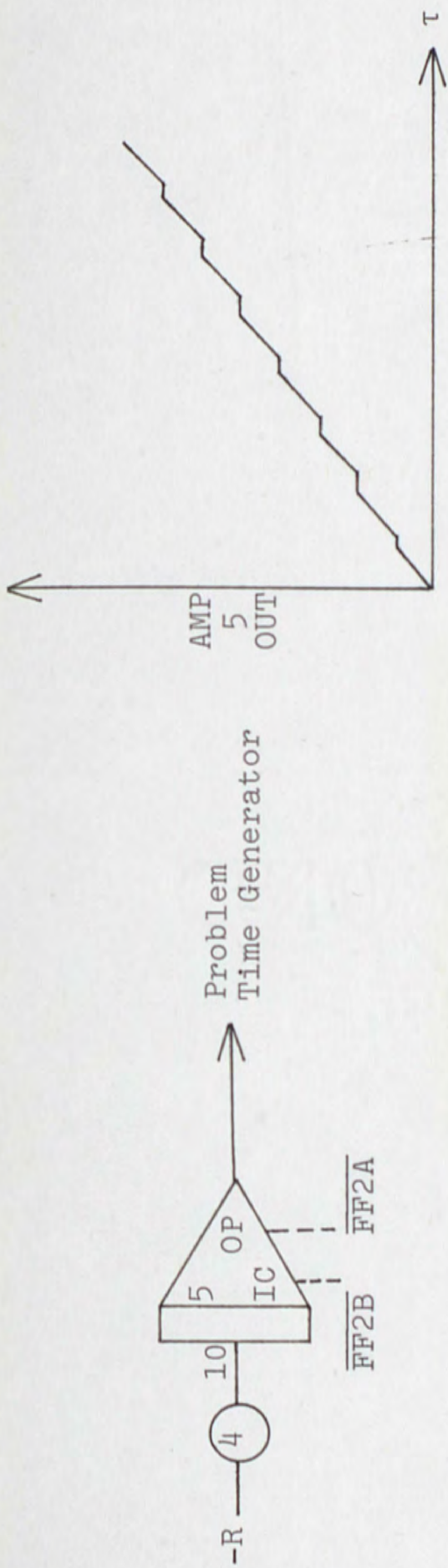


Figure 26. EAI 680 Time Base Generator and Second Order Continuous System

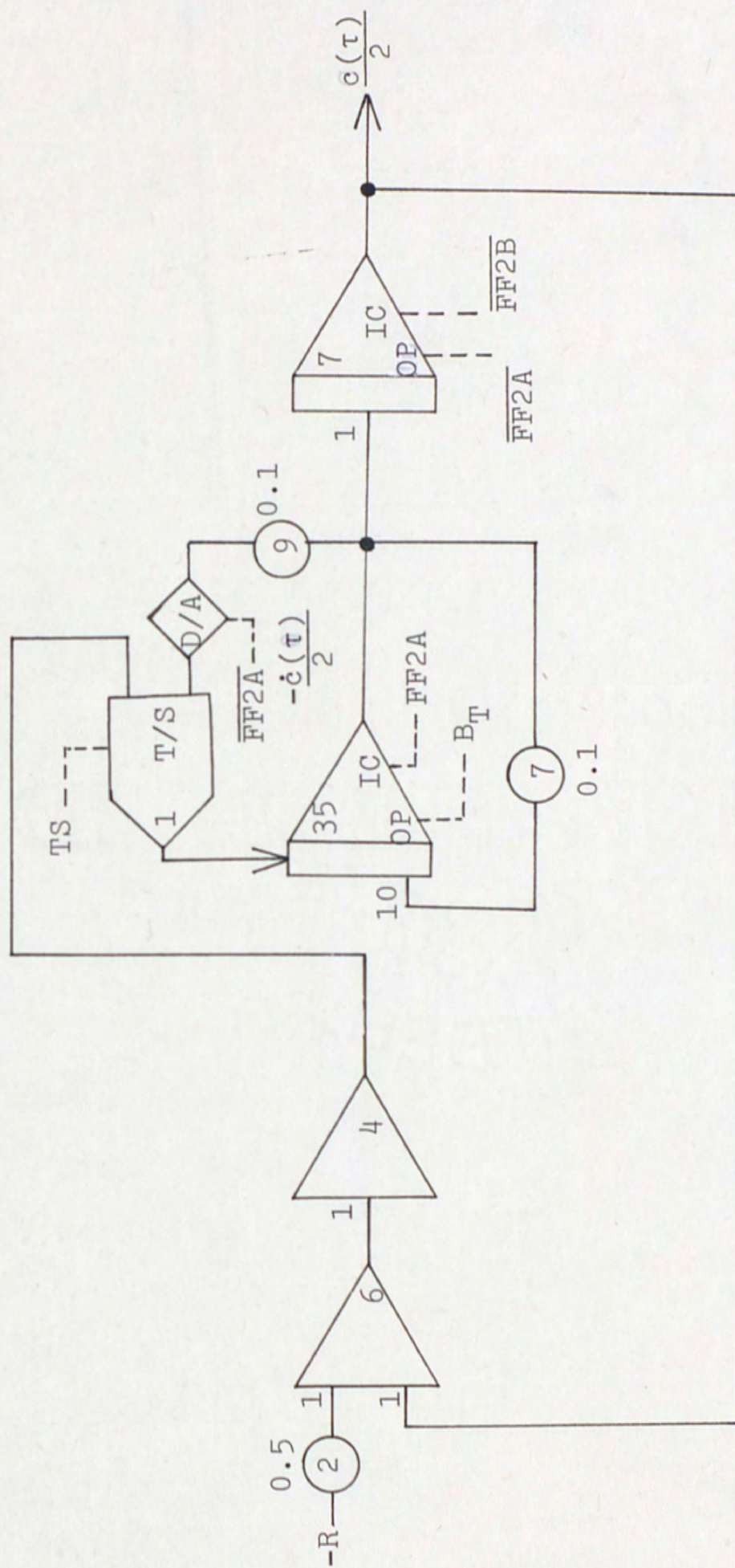


Figure 27. Second-Order System With Sampler But No Hold Device (EAI 680)

matrix that explains the mode control operation is as follows:

Logic Control Variable	Mode Control		
	IC	OP	HOLD
OPERATE	0	1	1 or 0
HOLD	0	0	1

Figure 11 which is in the section on the A/D-5 logic program shows the control variables that were used to control the modes of the integrator in the different second-order configurations. The OPERATE mode control buss was controlled by the variable B_T and the HOLD mode control buss was controlled by the complement of the variable out of gate 21, ($\overline{21}$).

The second-order continuous system and the second-order system with a sampler but no hold device programs were the same as those of the EAI 680 with the exception noted above. There was a difference in the second-order system with a zero-order hold device problem. Here an integrator (INT 10) had to be used in place of a physical track/store unit. It was programmed as in Figure 28 to go through an IC, HOLD, IC, HOLD mode of operation. The results were exactly like those obtained in the EAI 680 program.

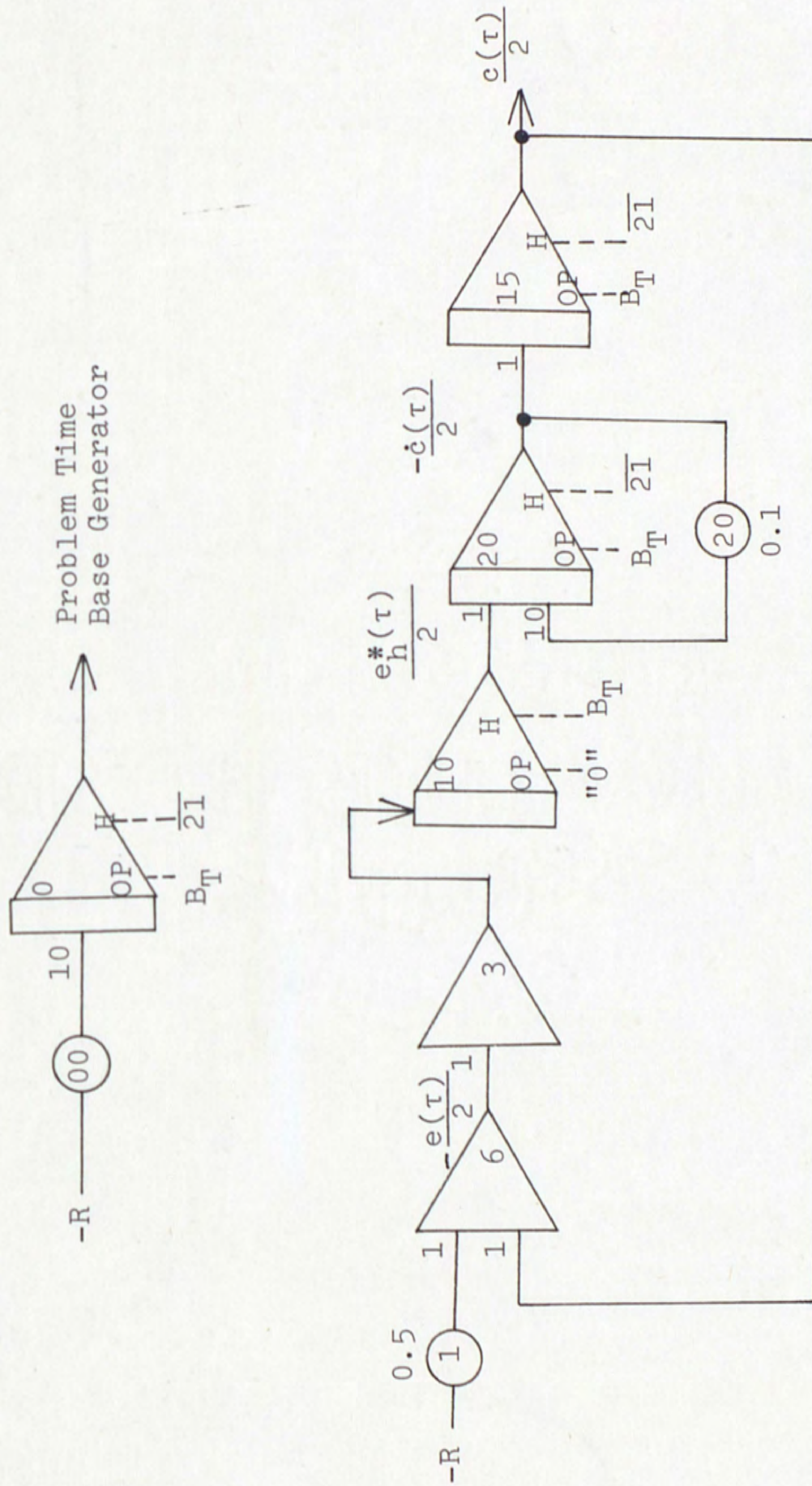


Figure 28. Second-Order System With Zero-Order Hold (A/D-5)

CHAPTER III

REAL TIME SIMULATION OF SAMPLED-DATA SYSTEMS

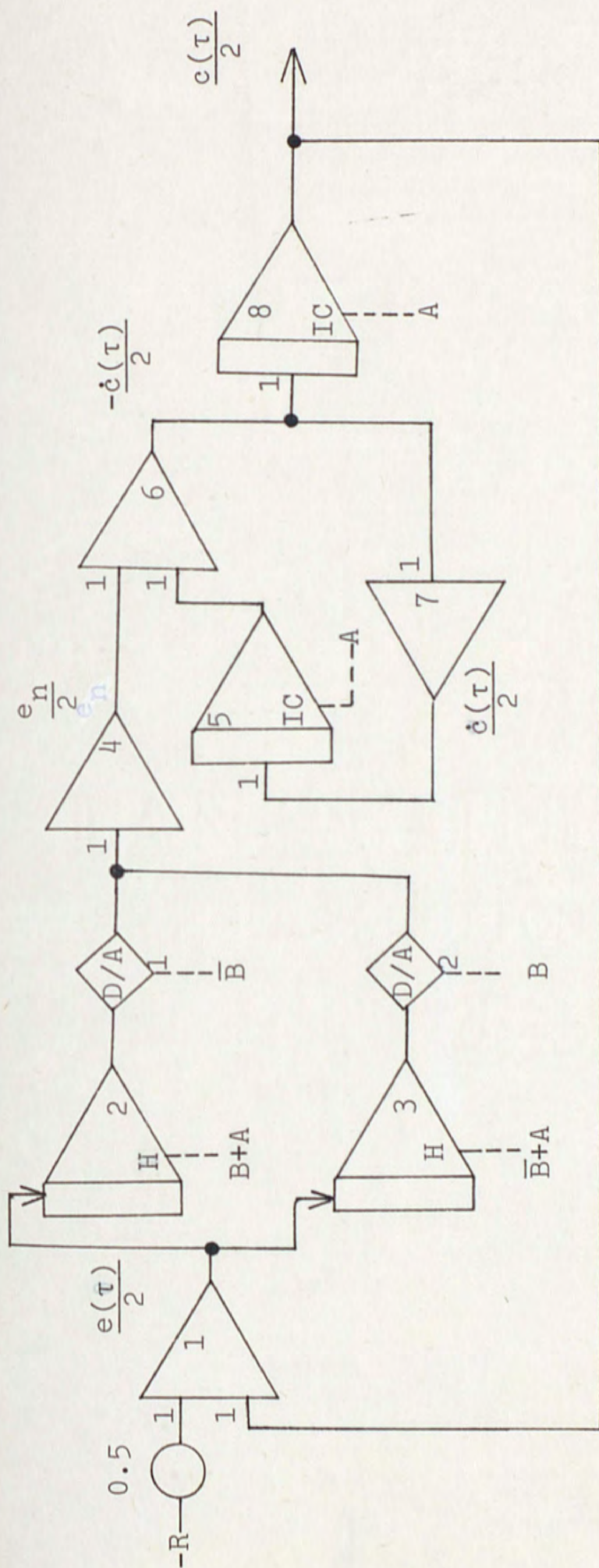
All the simulation techniques developed thus far have been accomplished using the update period as previously described. The logic control variables to generate this period of time where information is updated are not at all mandatory. As a matter of fact, there are some good advantages of the update technique such as hardware requirements are kept to a minimum, but there is also a distinct disadvantage. This big disadvantage is that the microprogram for updating the information in a hold device simulation cannot be utilized with any physical hardware. Suppose the plant were a simple Type I servo system, as described, but were not simulated on the computer while the sampling and hold portion of the system is analog computer simulated. This analog computer and servo hardware interface could not be accomplished due to the fact the computer microprogram is using a "simulated time" base while the servo hardware is using a "real time" base. Here, "real time" is defined as not having an update time frame while "simulated time" means that one is required.

The way to overcome this particular interface problem would be to make the computer program utilize a "real time" base. Some "real time" simulations will now be discussed so that such an interface could be accomplished.

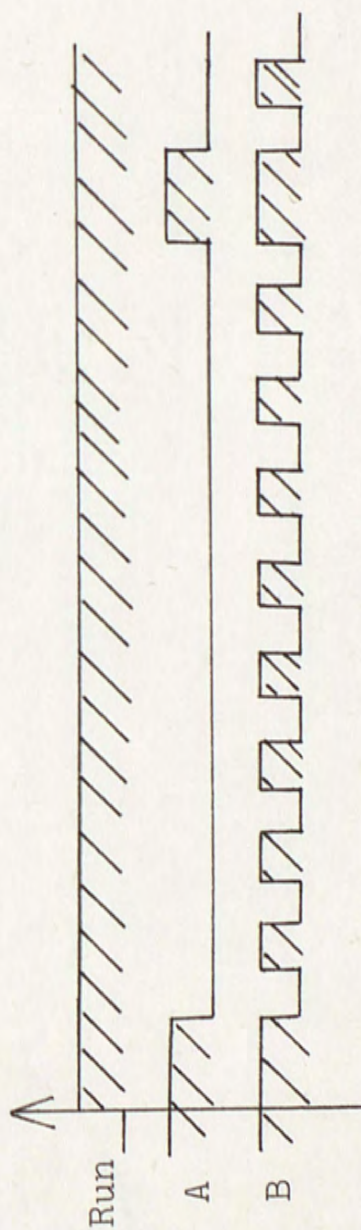
Figure 29 illustrates a program for a "real time" second-order system with a sampler but no hold device. Integrator 2 has its output (IC) fed through the D/A-1 switch during the \bar{B} period to summing AMP 4. At the same time, integrator 3 is in a hold mode. When the \bar{B} period is over, integrator 2 goes into a hold mode while integrator 3 goes into an IC state. In this simulation the closed loop that consists of amplifiers 5, 6, and 7 is used to provide the jumps required to make the system respond as if it received a train of impulse functions. This no hold case is one that is purely academic except under certain conditions.

The "real time" simulation of a second-order system with a zero-order hold device is shown in Figure 30. The logic control variables are the same as those shown in Figure 29(b) for the no hold simulation. The two track/store units (T/S-1 and T/S-2) and the D/A switches are used to perform the function of the sampler and the hold circuit. The track/store elements alternately store $e(nT)$ while the D/A switches gate the most recently stored value of $e(nT)$ to produce the zero-order hold output, $e_h(\tau)$.

Finally, Figure 31 shows the simulation of a "real time" second-order system with a first-order hold device.



(a)



(b)

Figure 29. "Real Time" Simulation of a Second-Order System With No Hold Device

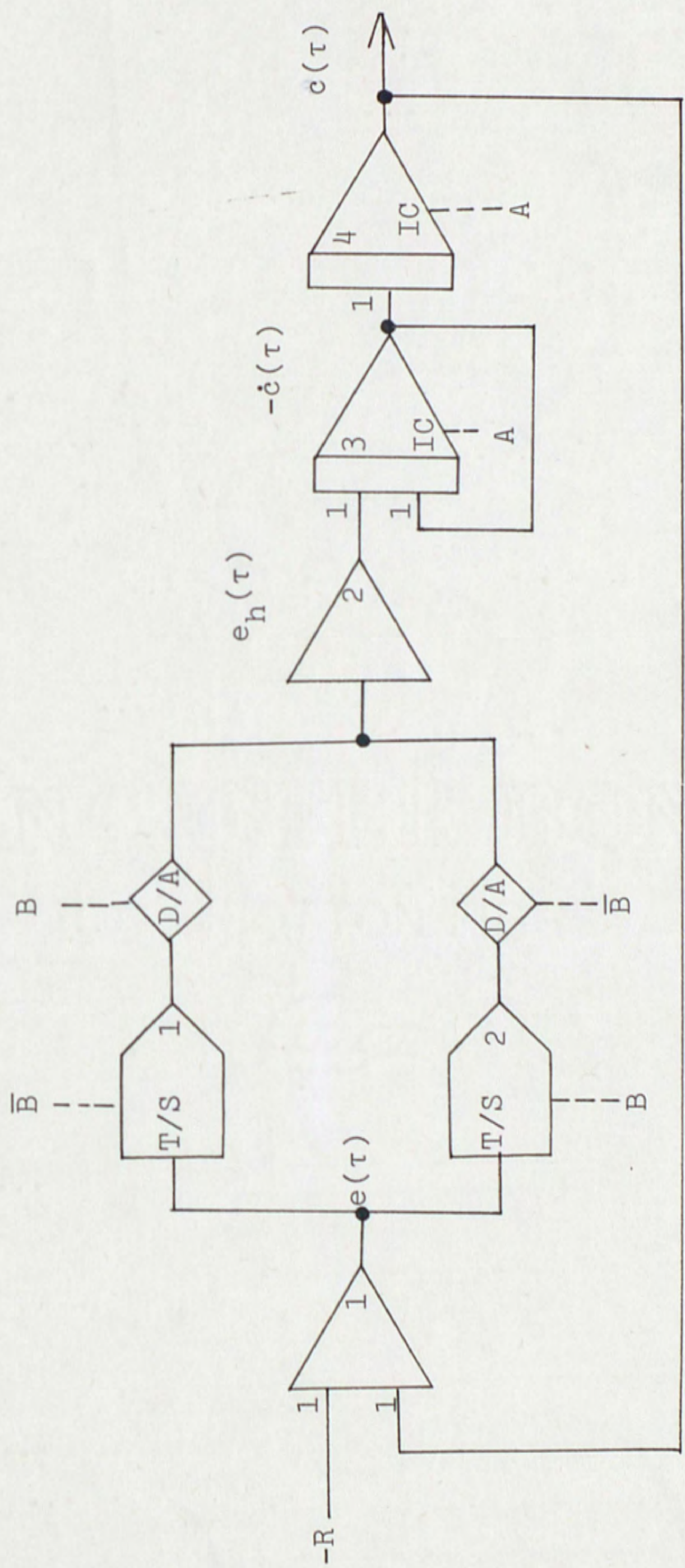


Figure 30. "Real Time" Simulation of a Second-Order System With Zero-Order Hold

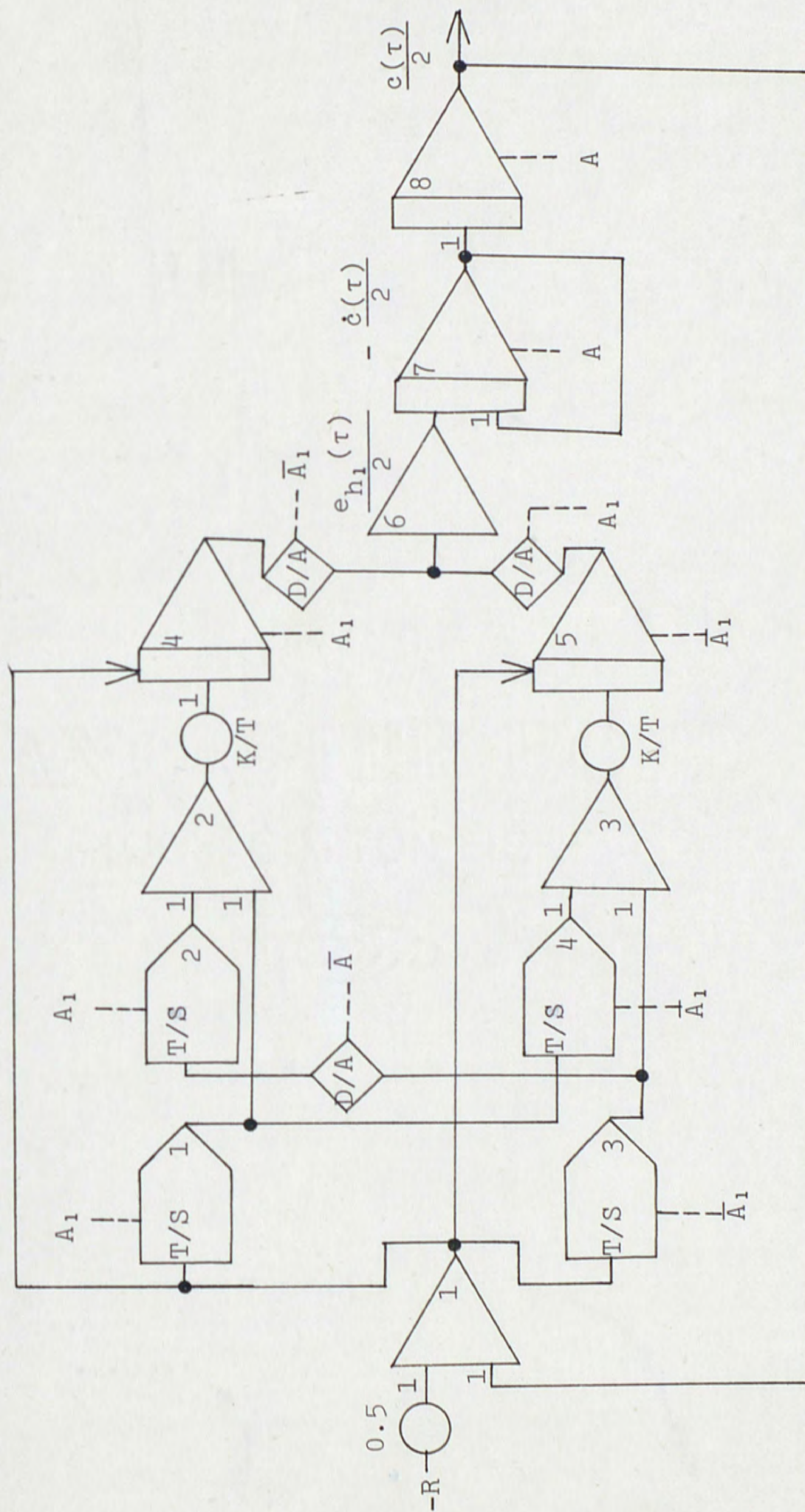


Figure 31. "Real Time" Simulation of a Second-Order System With a First-Order Hold

The four track/store, two summing amplifiers, two potentiometers, two integrators, and three D/A switch elements are required to simulate the sampler and the hold circuit. The first pair of track/store elements alternately store $e(nT)$, while the second pair controls $e(n-1)T$. The input to the integrators is the sum of output of each pair of track/store elements, $\frac{K}{T}[e(nT)+e(n-1)T]$. The D/A switches gate the output of the integrators to generate the output of the hold device, $e_{h1}(\tau)$. The parameter K is termed the fraction-order hold parameter, and its value is one for the first-order hold and of course other values may be chosen to produce fractional order hold devices.

As can be seen by comparing the "real time" simulation to the "simulated time" programs, especially Figures 31 and 19, more hardware is required in the "real time" simulation. When the update period is deleted exactly twice the hardware is required along with two additional D/A switches to alternately gate the signal. A "real time" simulation of a second-order hold would then require six track/stores, two integrators, and minor additional hardware, along with two gating D/A switches. A substantial savings in hardware is a result from using the "simulated time" base (update period).

Integrators, as shown previously in Figure 28, can be used as track/store units to perform zero-order hold characteristics. These can lead to distinct errors resulting from the tracking operation. The output signals'

rate of change can introduce these errors for signals of relatively high frequency and large amplitude.

The inherent error associated with any basic operational amplifier is generally due to its bandwidth. However, there is another limitation on an integrator and that is placed on the rate of change of the integrator's feedback capacitor. This is usually much more restrictive than the bandwidth limitation and is imposed by the current limits of the amplifier output. According to Graeme, Tomey, and Huelsman [9], the expression for this rate is the following:

$$\left(\frac{dv_o}{dt} \right)_{\max} = \left(\frac{dv_c}{dt} \right)_{\max} = \frac{I_{\text{Lim}}}{C} \quad (20)$$

where I_{Lim} = output current limit

C = feedback capacitor

This expression is based upon the fact, as in an inverting amplifier, the summing point of the amplifier is held at a virtual ground by the high gain and the feedback network of the amplifier. Therefore no input current can flow into the input terminal of the amplifier and is forced to flow into the feedback capacitor. Since one end of the capacitor is connected to the virtual ground point, the output voltage of the amplifier equals the capacitor charging voltage. The integrator circuit also has the low output impedance normally associated with a feedback amplifier.

Figures 32(a) and (b) show the step responses of an integrator and a summing amplifier, respectively. The velocity limit of the integrator due to the tracking errors is denoted by m_1 , and the specified velocity limit of the amplifier is m_2 .

$$m_1 = \frac{I_{Lim}}{C} \quad (21)$$

where $C = .02$ uf for gain 500, and $I_{Lim} = 20$ milliamperes

$$m_1 = \frac{20 \times 10^{-3}}{2 \times 10^{-8}} = 1.0 \text{ volt/microsecond} \quad (22)$$

To determine a time per volt error, use the area of the triangle

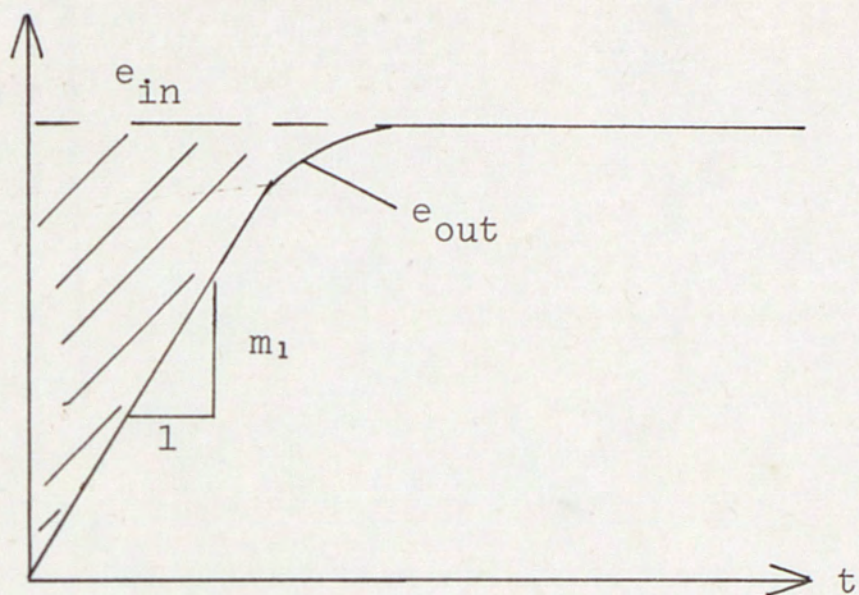
$$\Delta error_{(INT)} = error = \left(\frac{1}{2} \right) \left(\frac{10^{-6}}{1} \right) (1)(10^3)(.5) =$$

$$error = 2.50 \times 10^{-4} \quad (23)$$

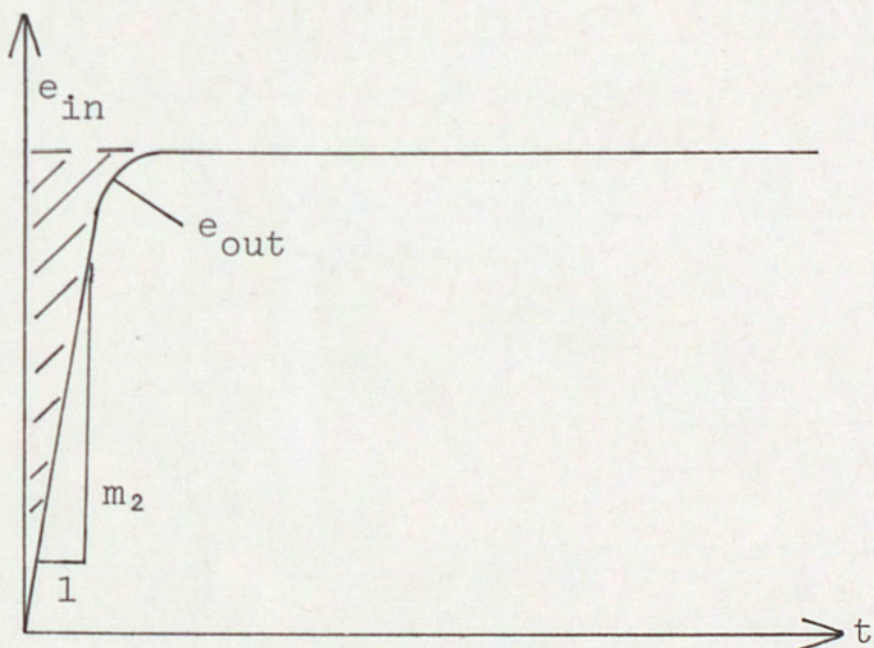
$$\% \text{ error} = 0.025\% \quad (24)$$

The specified value of m_2 is 25 volts per microsecond. Therefore, the tracking error of the summing amplifier to that of the integrator is 1/25 of what it was or about 0.001%. The calculations are based on data from the EAI-580 specifications.

Even though this error appears to be very small it must be remembered that this is one error that is accumulative. This much error is introduced for every unit used in this mode of operation for every sample period. This error, it



(a)



(b)

Figure 32. Output Response for Step Input

must also be pointed out, is aggravated by the high speed computing characteristics of the system. There is a way to reduce these errors. That method is to use a simulation similar to that shown in Figure 31. By using the dual D/A switches, track/store, integrator complementary circuits, the most error that will be encountered will be that of the switch times which is in the order of one microsecond.

CHAPTER IV

ADAPTIVE AND MULTIRATE SAMPLING

Adaptive sampling, as described earlier, is a technique where the sampling frequency is varied continuously, according to a control law, as the system operates. Here, "real time" simulations will be employed to develop some adaptive sampling techniques. The basic ingredient of any adaptive sampling system is the control law which varies the sampling frequency. The control law implemented was that of Bekey and Tomovic [4]. Their control law which evolved from the development of a "local sensitivity function" is a function of the derivative of the output signal of the second order sampled-data control system. The law is

$$T = \frac{1}{K_1 |\dot{c}_n| + B} \quad (25)$$

with $T_{\max} = 1/B$ when $\dot{c}_n = 0$.

Before the control law could be implemented a microprogram was improvised that would actually vary the frequency of the sampling device. The voltage controlled oscillator microprogram that was used to perform this function is shown in Figure 33. Logic timing diagram for the oscillator is also shown in the same figure. The signal V_f is the output of the control law.

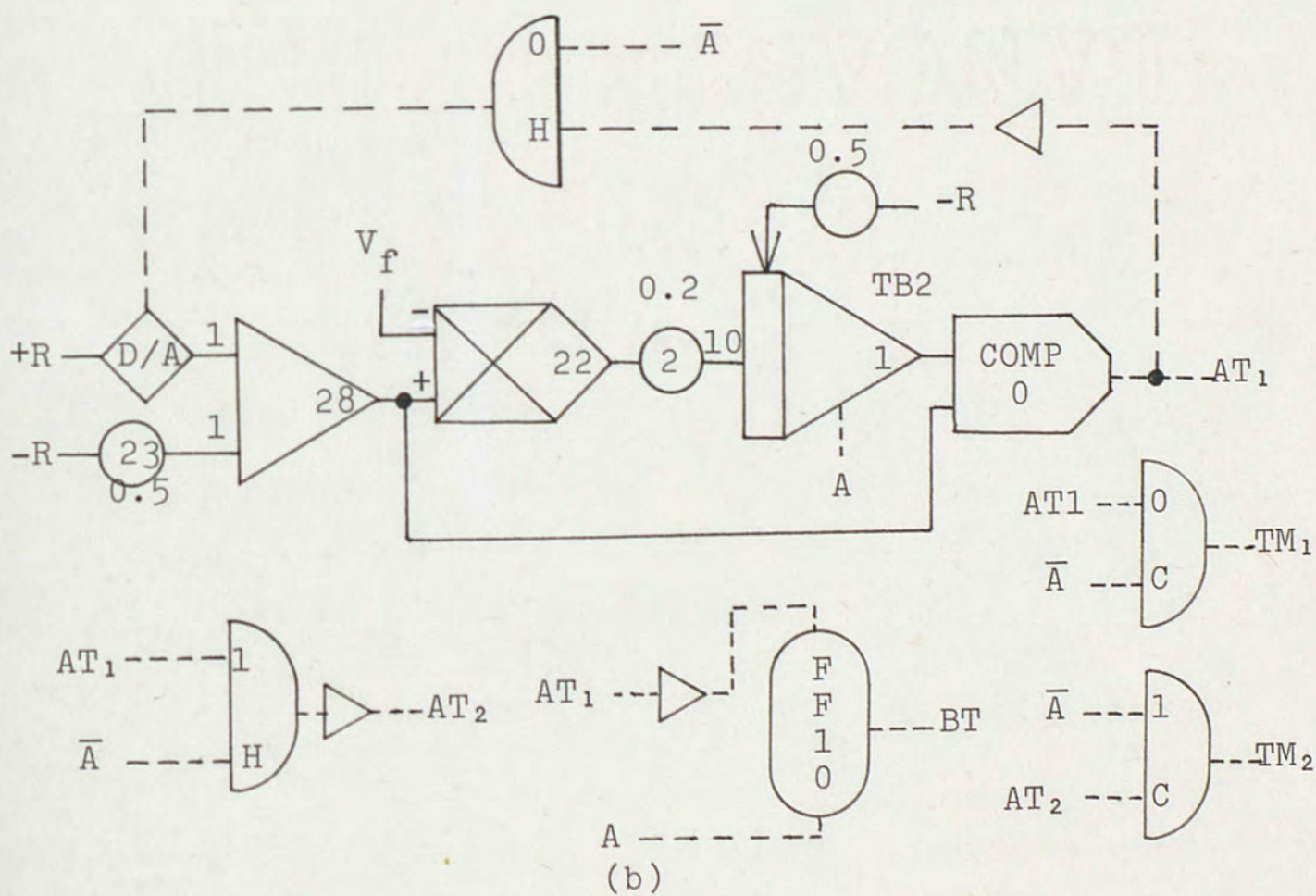
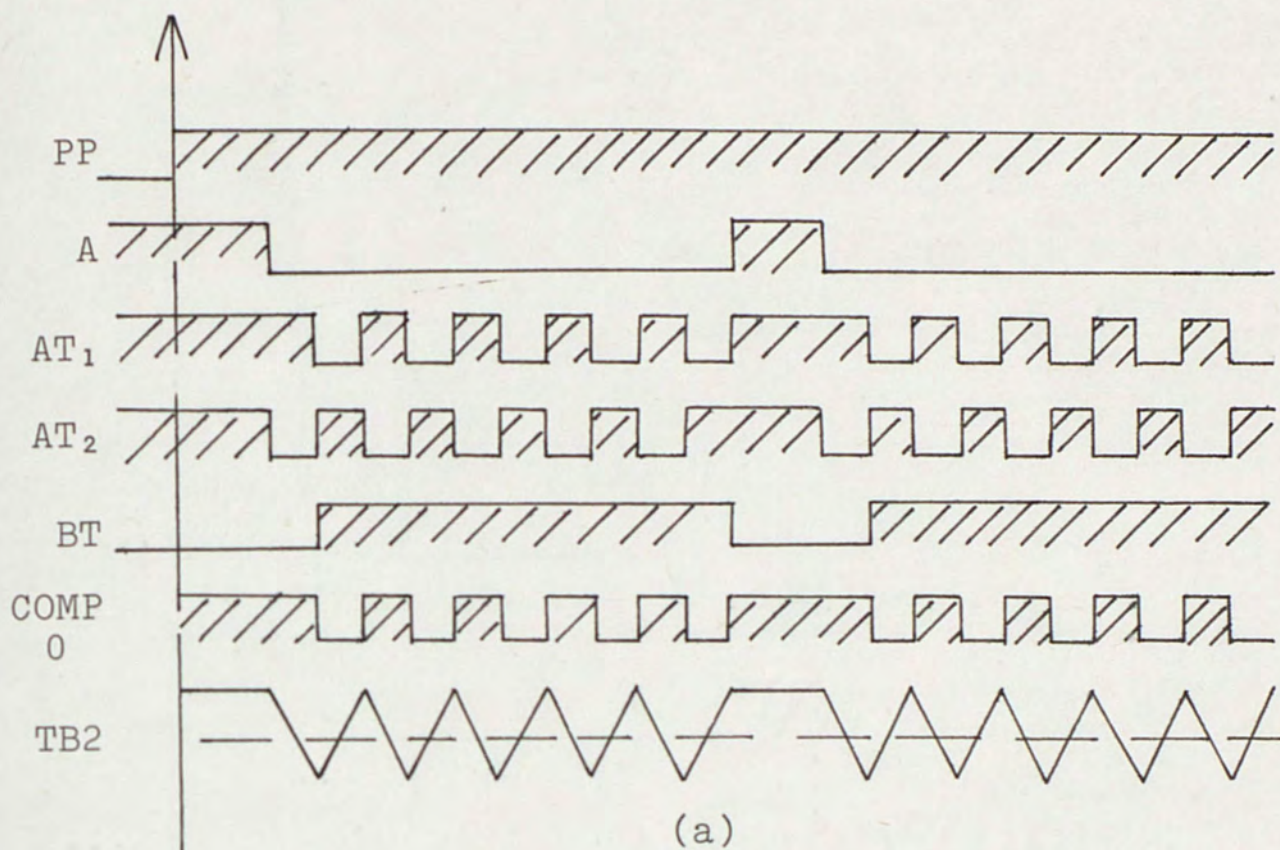


Figure 33. Voltage Control Oscillator

To explain the operation of the voltage control oscillator, assume $A(IC)$ is a logic one and the signal V_f is unity. This will make INT 1 be in an IC state, and its output will be positive due to its negative IC input voltage. Since COMP 0 has a positive voltage on its input, its output will be at a logic one. AND gate OH is low because its COMP 0 input is low. This will in turn keep the D/A switch off.

Now, as A goes low and INT 1 goes into the OPERATE mode, the integrator starts to integrate the positive input and produces a negative going output. When the output of the integrator goes far enough negative so that the sum of COMP 0's inputs are less than zero, the output will go to logic zero. This will turn the D/A switch on and make the input to the integrator negative. Now, the integrator output will start to ramp positive until the sum of the inputs on COMP 0 is greater than zero which will turn off the D/A switch, and the process will repeat.

Now that the sampling frequency controller has been developed, the second order system was programmed as shown in Figure 34. This system simulation is different from the one described in Chapter III, Figure 31. It has two multipliers, M32 and M35, in place of the two potentiometers that are used to control the order of the error hold circuit. Here, one potentiometer, POT 5, is used for this purpose. Then, a microprogram was added that would generate

the integral square error (ISE) figure of merit. Instead of using the error signal, in this particular application, the output, c , of the sampled-data control system in Figure 34 is the input to this ISE system. This figure of merit will be used to compare the desired output, for a step function on the input, to the actual simulation output.

Figure 35 shows the microprogram of Bekey and Tomovic's adaptive sampling control law. Here the multipliers M42 and M45 and the potentiometer POT 3 have the same purposes as M32, M35, and POT 5 in the error hold circuit. The parameters k_e and k_c will be used to denote the order of the hold circuit in the error portion of system and the order of the hold circuit in the control law, respectively.

Before any adaptive sampling techniques were employed, the system variables for a second-order system with a zero-order hold were recorded for $T = 1$ second and $T = T_{\max}$. The variables for the system when $T = 1$ second are the same as those shown previously in Figures 17 and 18. Figure 36 shows the system variables when T was set to T_{\max} . Similar data were taken when a first-order hold was installed in the control loop. The system data for the case $T = 1$ second is shown in Figure 21 while the system variable for $T = T_{\max}$ is shown in Figure 37. In both cases, T_{\max} was selected by adjusting POT 2 to a position just prior to causing the system to go unstable. An oscilloscope display of the

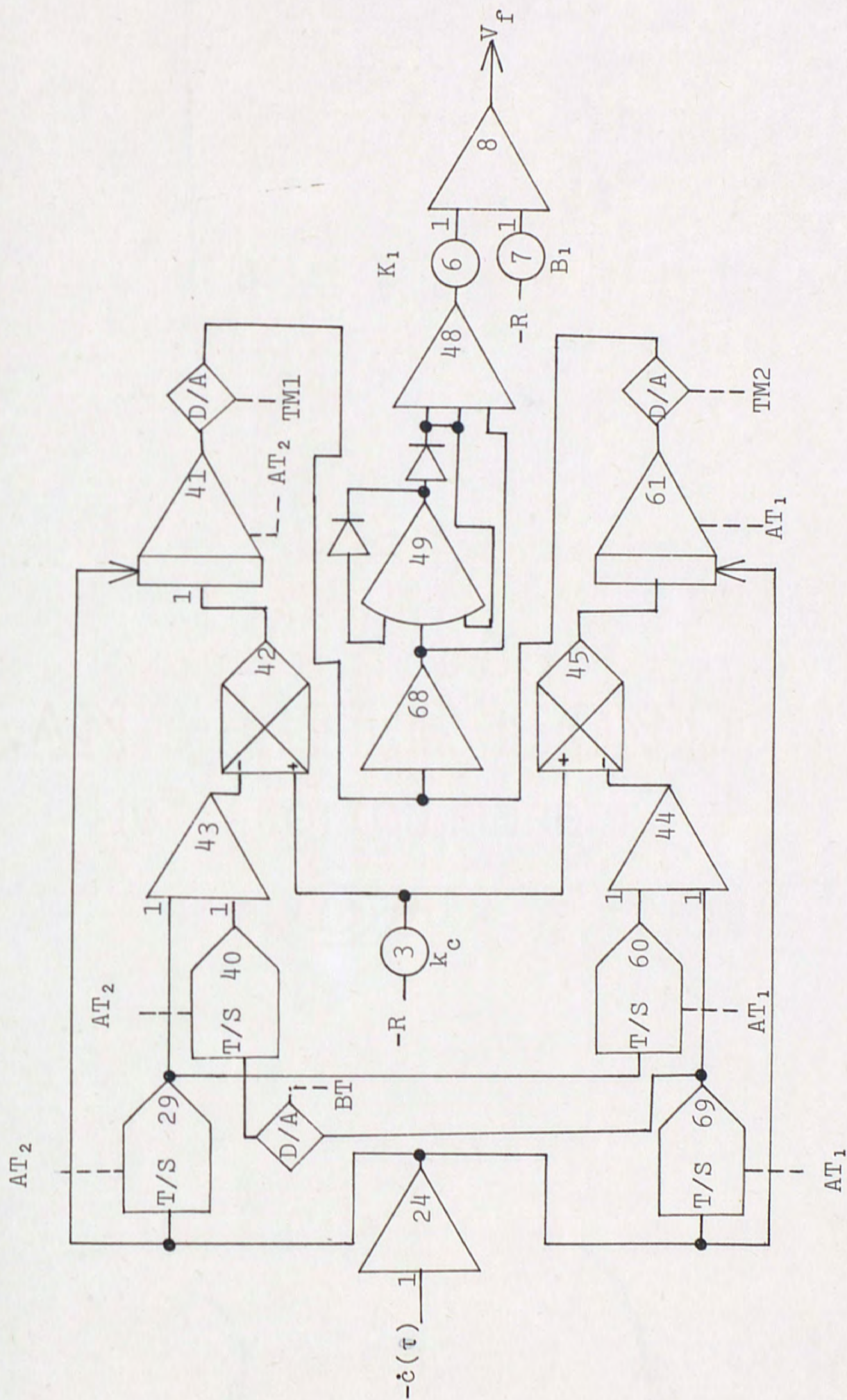


Figure 35. Bekey and Tomovic's Adaptive Sampling Control Law

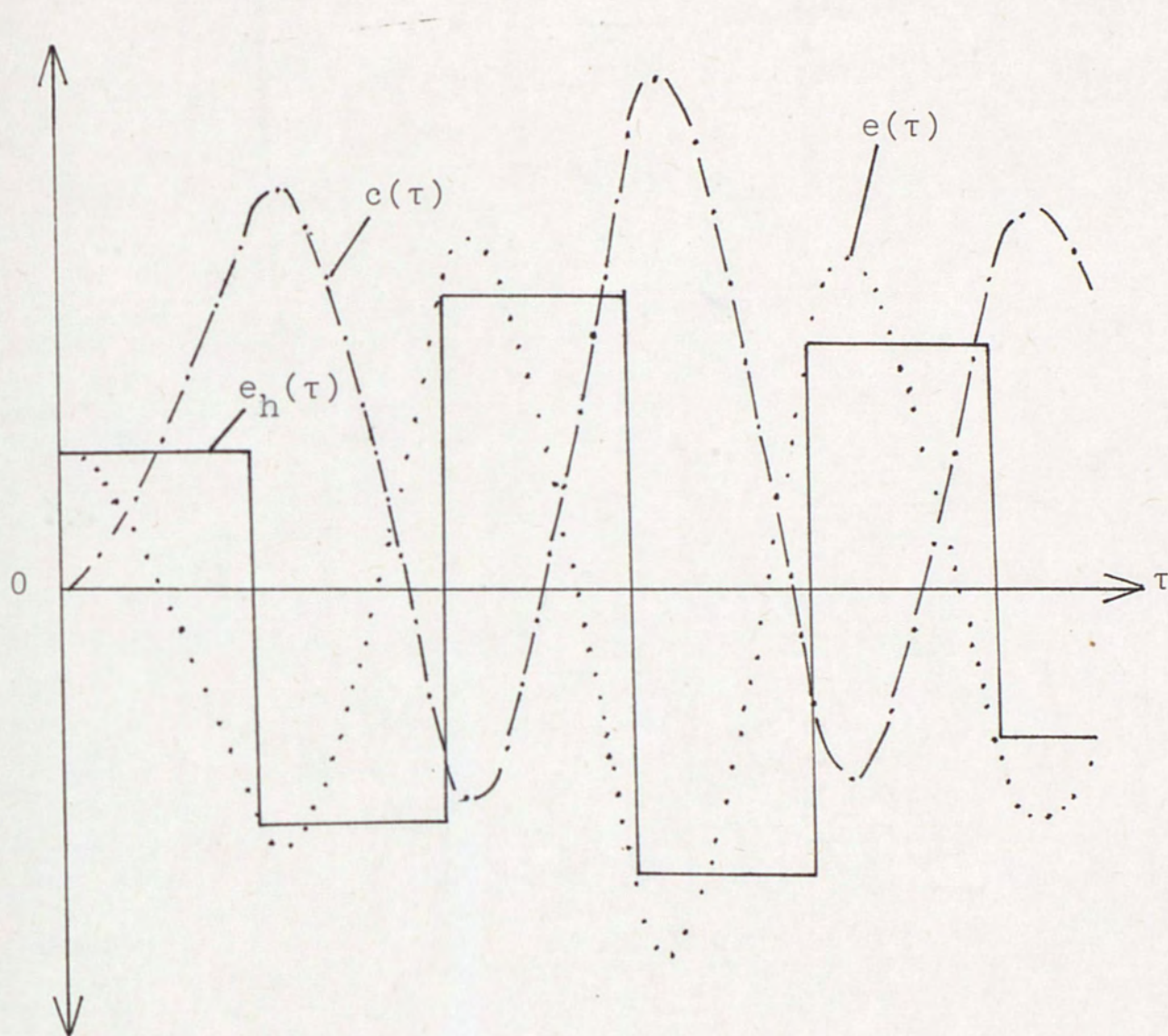


Figure 36. System Variables for Simulation in Figure 34 When Hold is Zero-Order and $T = T_{\max}$.

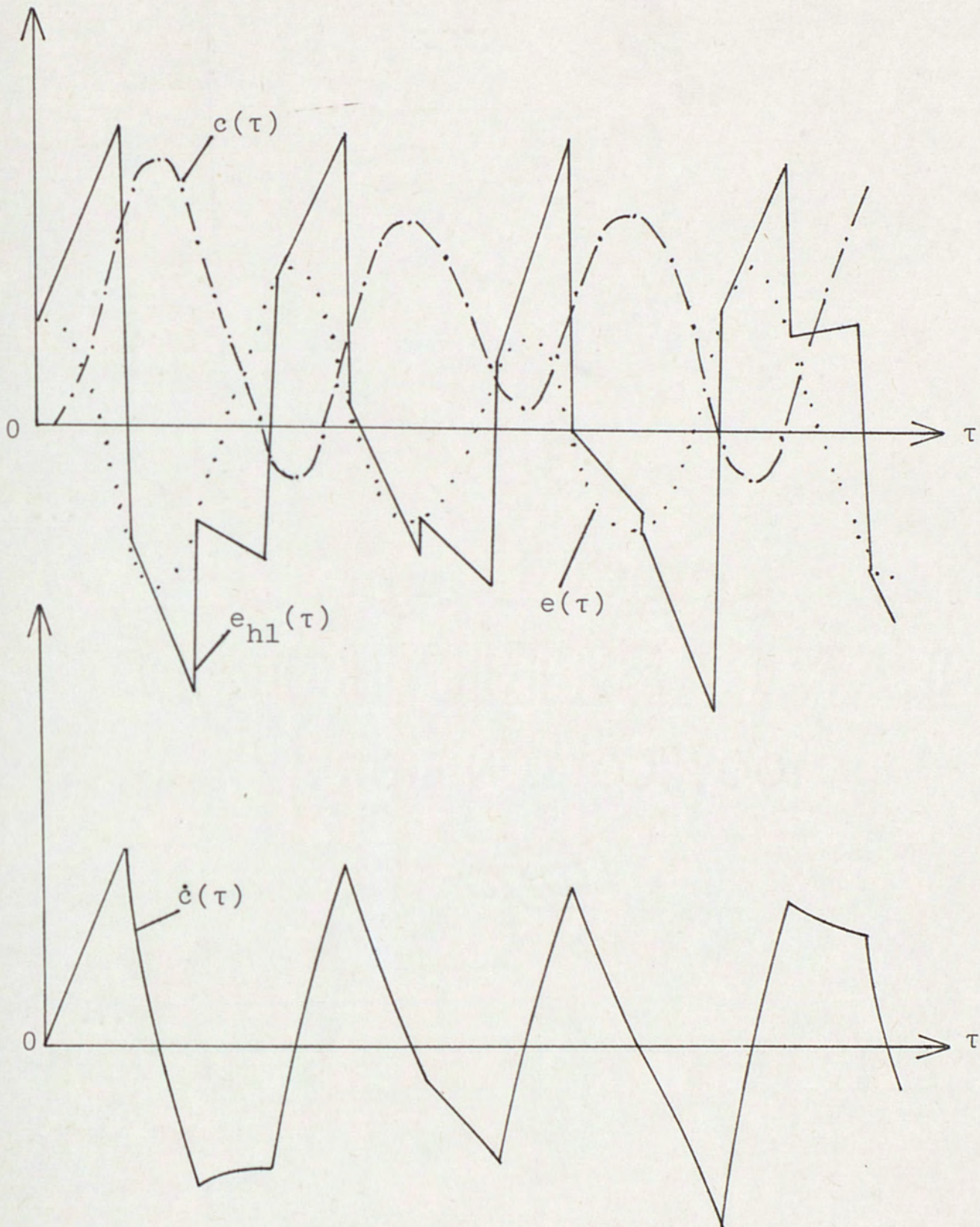


Figure 37. System Variables for Simulation in Figure 34 When Hold is First-Order and $T = T_{\max}$.

system output was used to determine when the system started to go unstable.

Data was also taken to determine the optimum value of the parameter k_e for $T = 1$ second. Here the ISE figure of merit was used as a guide in selecting this optimum value which is defined as that value which produces a minimum value of ISE. The ISE versus k_e data is shown in Figure 38, where the ISE curve is normalized about the ISE for a zero-order hold, ISE (0). The optimum value of k_e was 1.263. Potentiometer 5 was then set for $k_e = 1.263$ and system parameters were taken for a minimum ISE system. Figure 39 shows these system variables for the optimum k_e condition.

The adaptive sampling simulation schemes were implemented for four different cases and system performances were gauged by comparing the respective sampling efficiencies. As explained previously, these four implementations employed the Bekey and Tomovic control law shown in Figure 35. The four conditions used were when the order of the error hold device was zero while the control law hold was varied from a zero to a first-order hold. Next, the order of the error hold was a first-order while the control law hold was again varied from a zero to a first-order hold.

Some precautions had to be taken into consideration before the data could be obtained. POT 7 (B_1) had to be set low enough so that the following could be ensured:

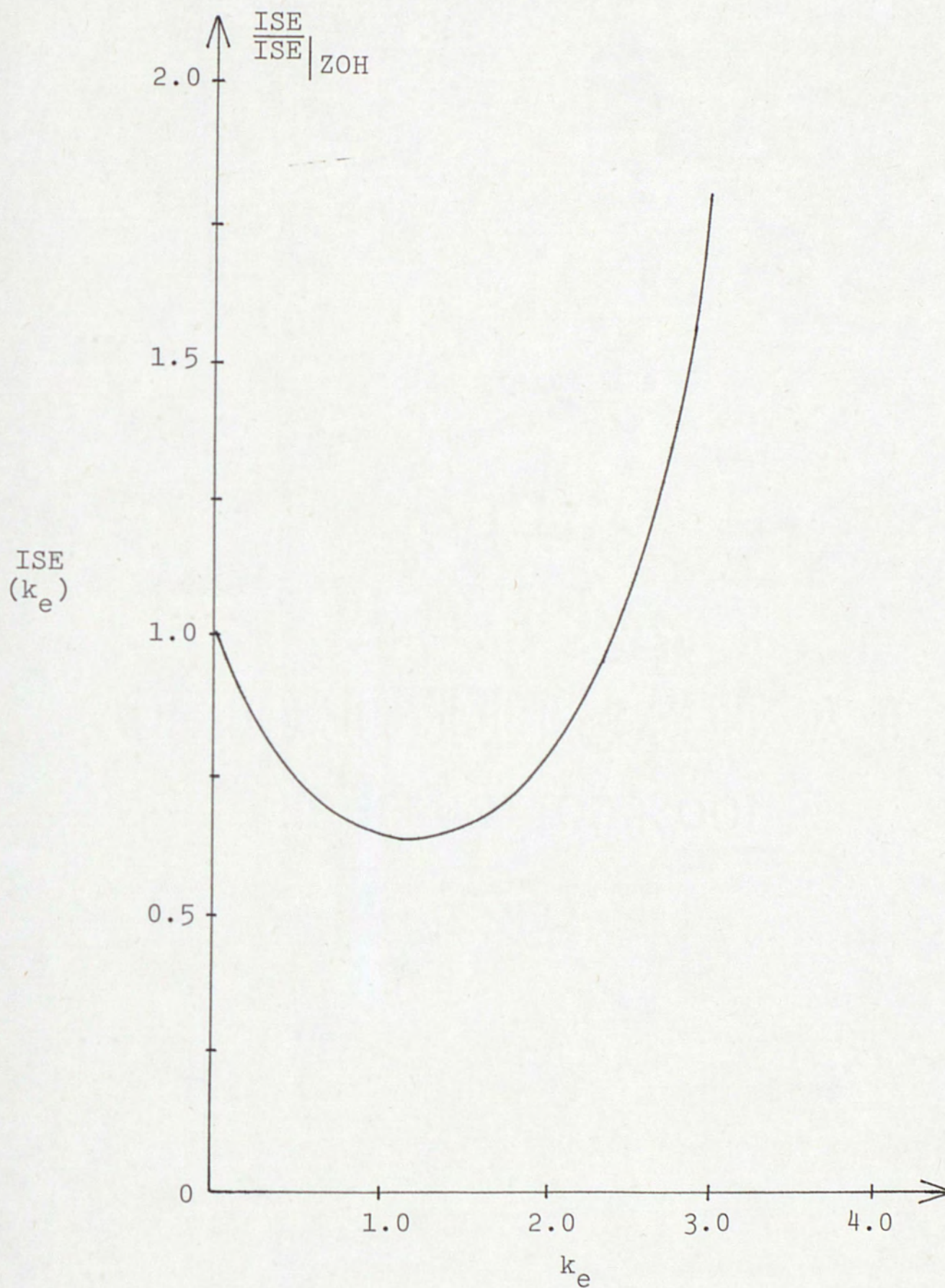


Figure 38. ISE as a Function of the Error Fractional Order Hold Parameter.

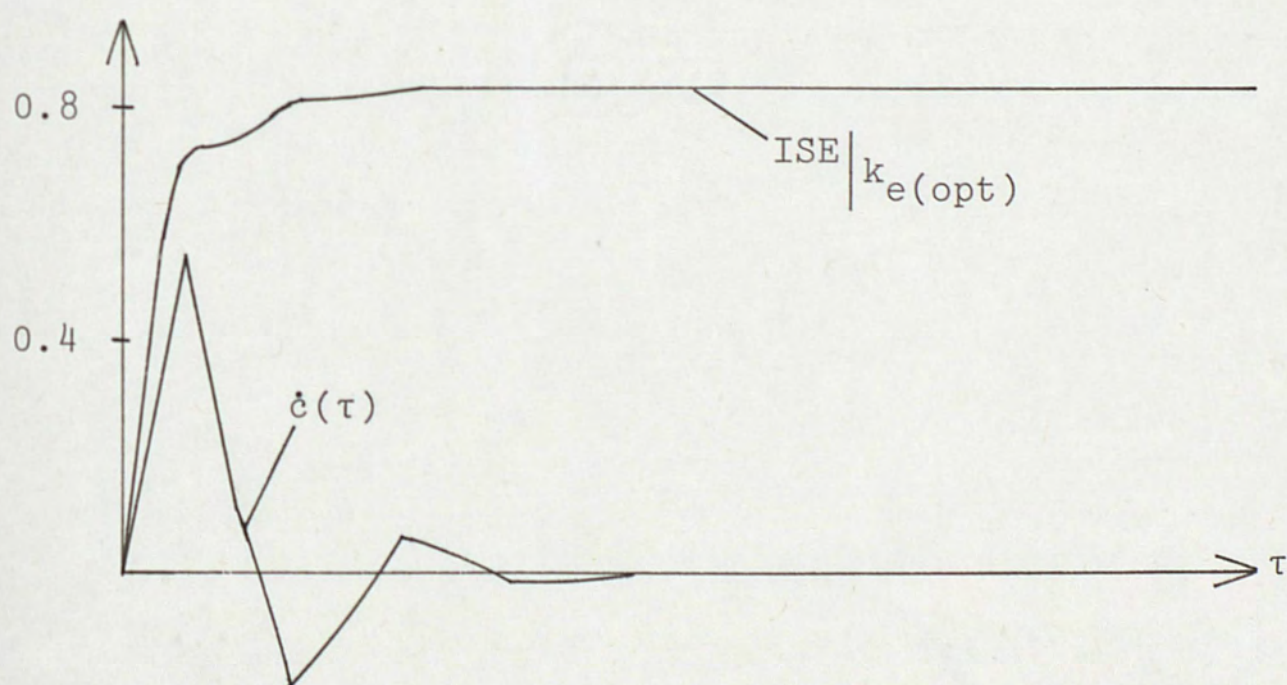
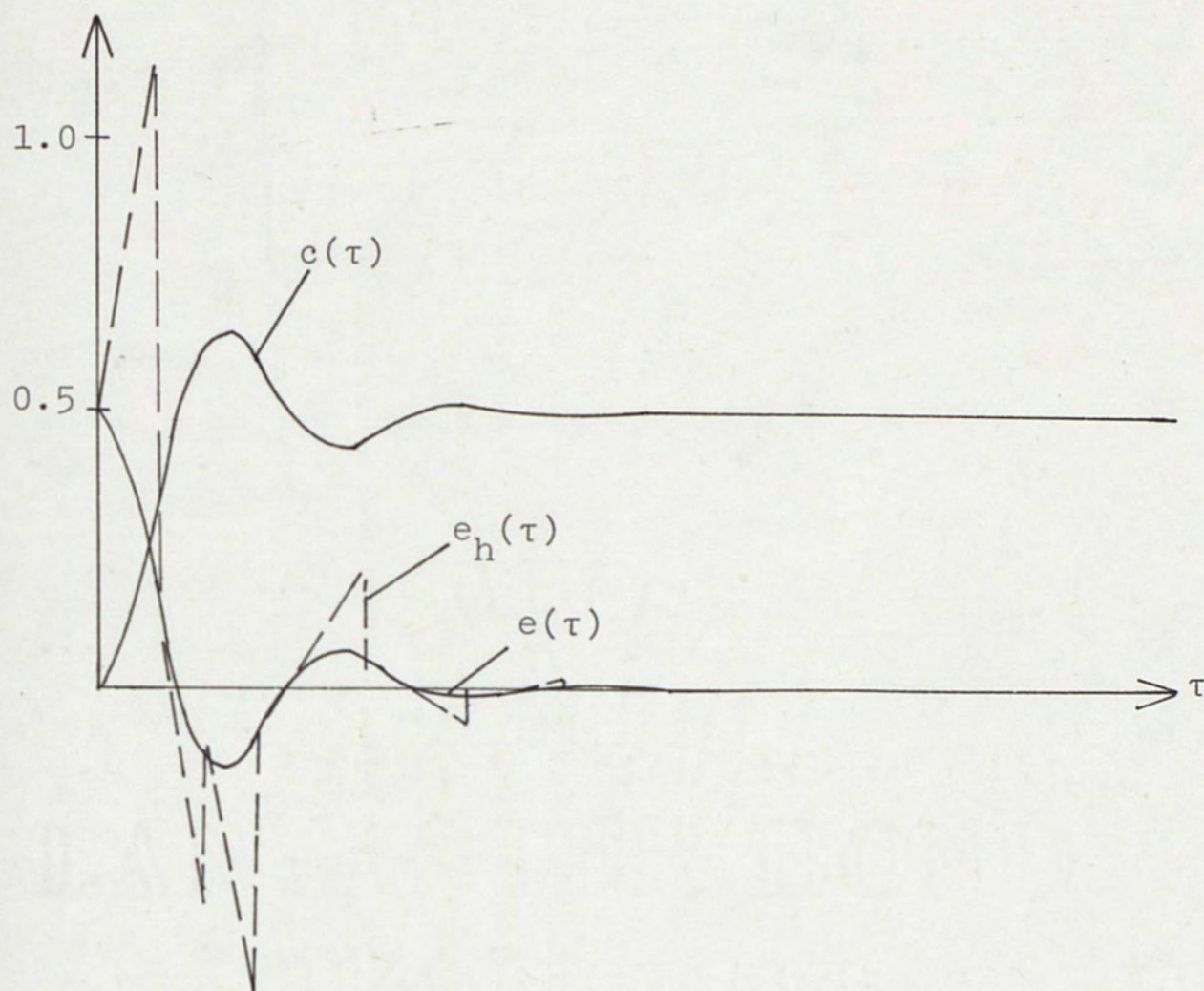


Figure 39. System Variables When $k_e = k_{e(opt)}$

$$[K_1 |\dot{c}_n| + B_1] |\dot{c}_n|_{\max} < \text{Ref} \quad (26)$$

and also low enough to satisfy

$$[T] |\dot{c}_n| = 0 \leq T_{\max} \quad (27)$$

This precaution helped to insure AMP 8 would not become overloaded. The program was then scaled for a maximum sample period (T) of 5 seconds when $|\dot{c}_n| = 0$. B_1 was selected to be 0.04 and POT 2 (B_2) was set at 1.0 and the dynamic operating range was judged to be acceptable.

Data was then taken for two distinct cases as both parameters k_e and k_c were varied from a zero-order hold to a first-order hold. Case one would be to adjust K_1 and B_1 to yield a minimum ISE figure of merit while the number of samples was fixed at 20. The second case was to adjust K_1 and B_1 to yield the minimum number of samples while keeping the ISE value equal to the ISE value for a zero-order hold with a sample period of 1 second. The performance data for these eight sets of conditions are shown in Table 1. In this table the value of all ISE numbers is normal with respect to the ISE value of a zero-order hold and a sample period (T) of one second.

Figures 40 and 41 show the two cases when both of the hold circuits in the error portion of the system (k_e) and also that of the control law (k_c) are zero-order units. There are a number of other system variables shown in the

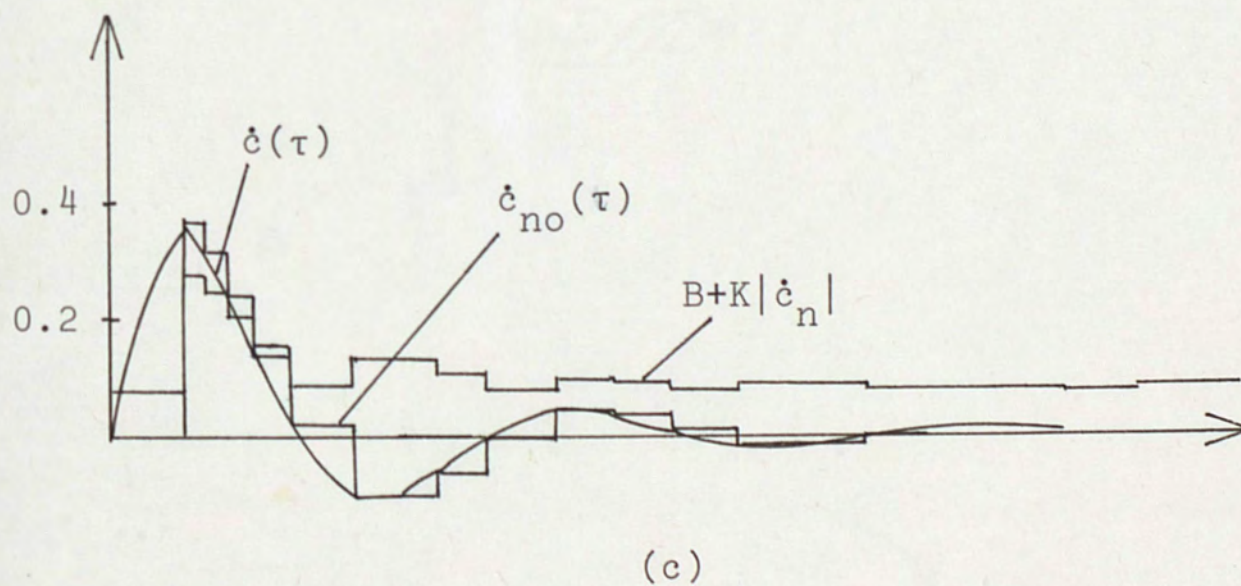
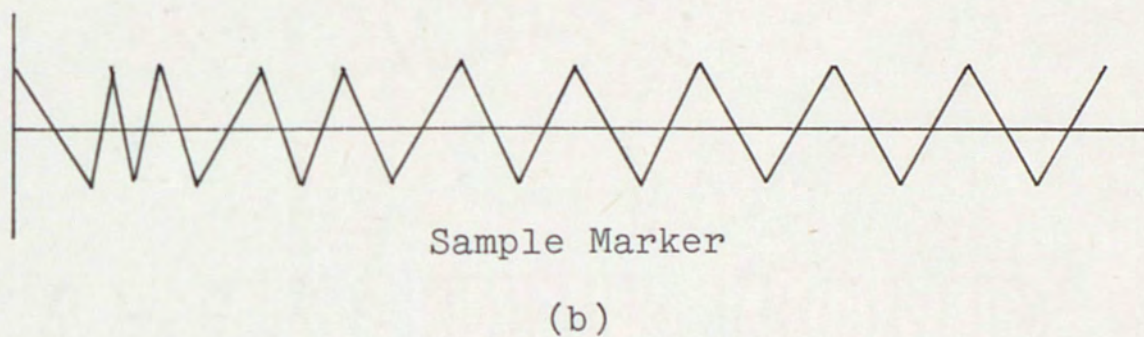
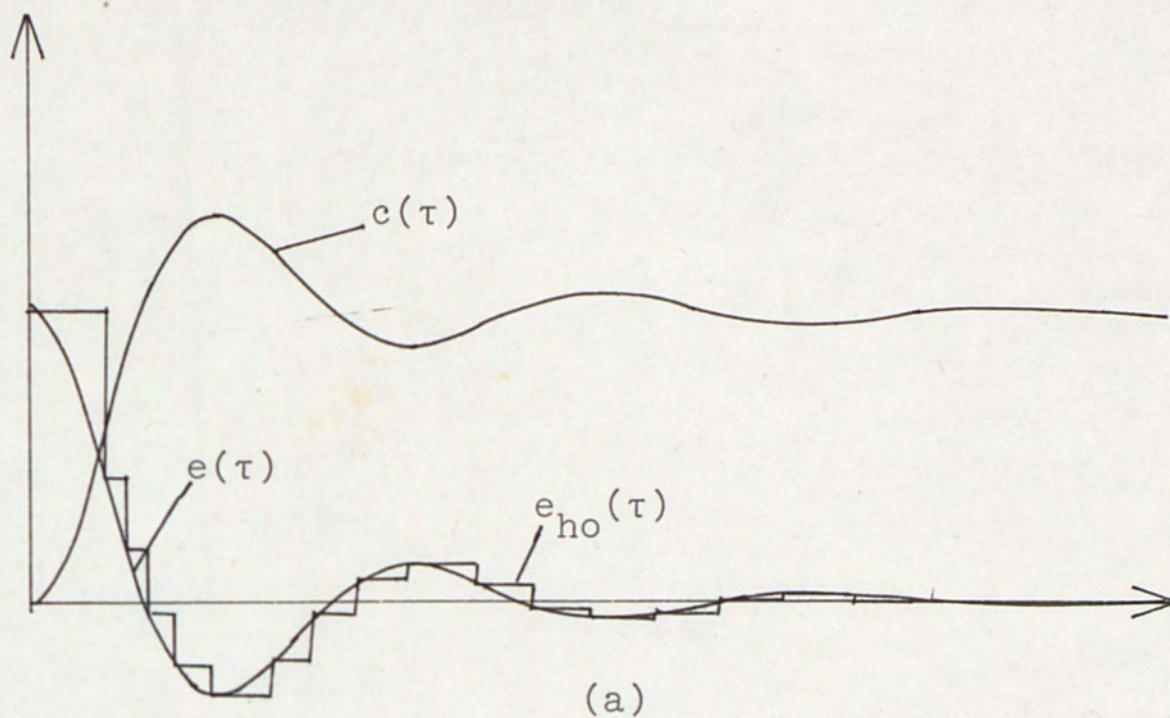


Figure 40. System Variables When $k_e = k_c = 0$, $N_s = 20$ and Minimum ISE.

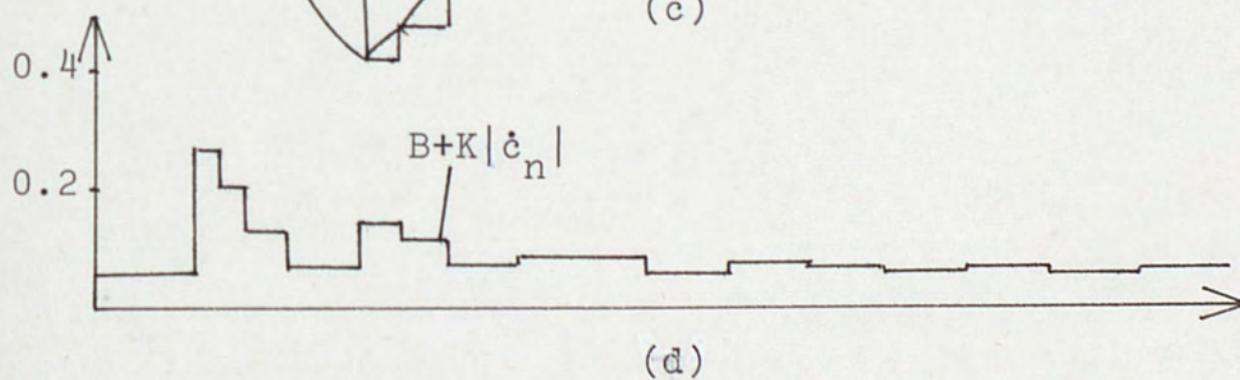
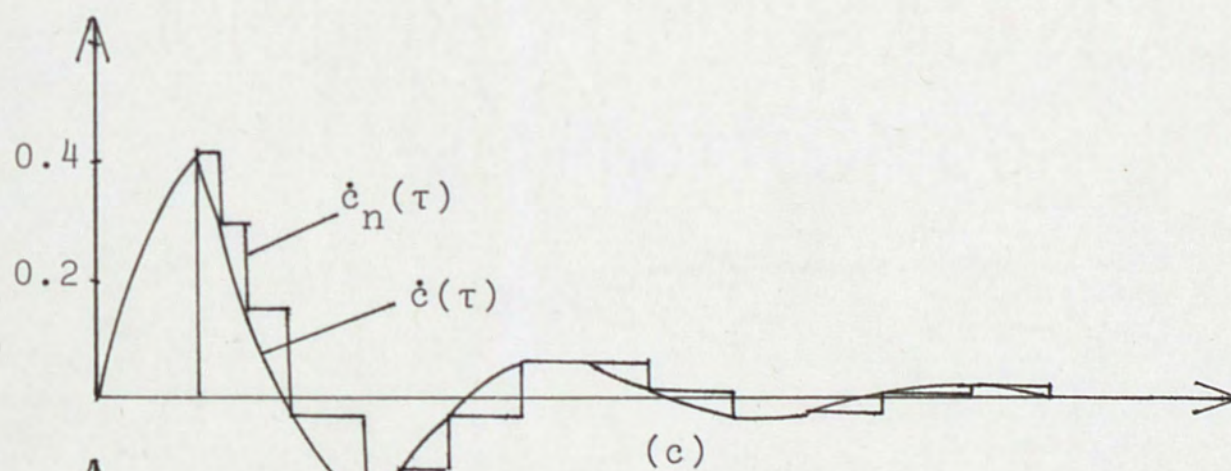
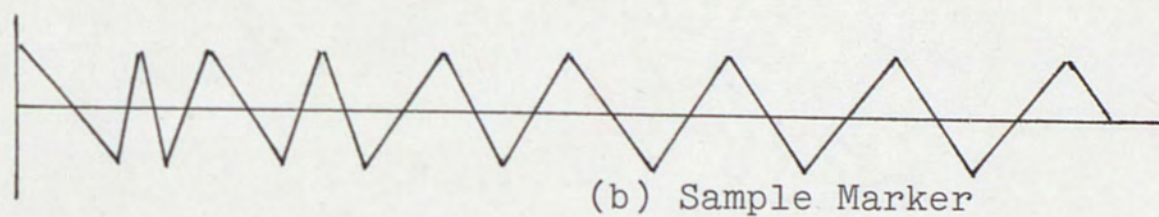
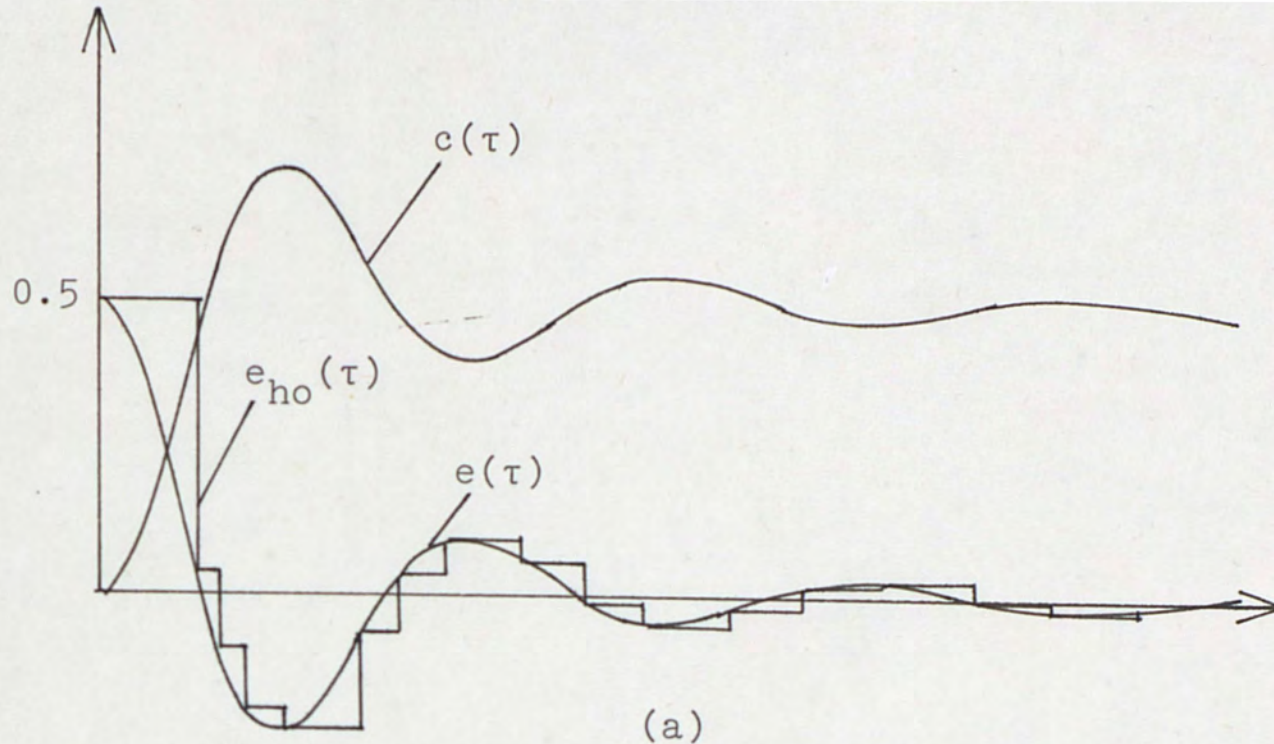


Figure 41. System Variables When $k_e = 0$, $k_c = 0$ and Minimum N_s .

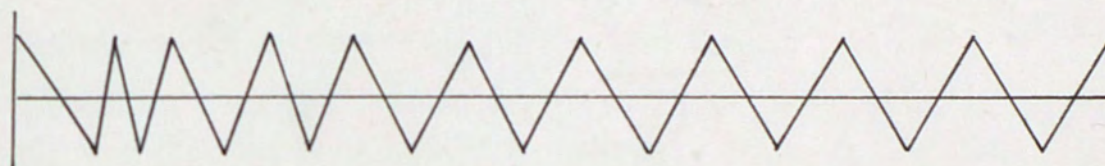
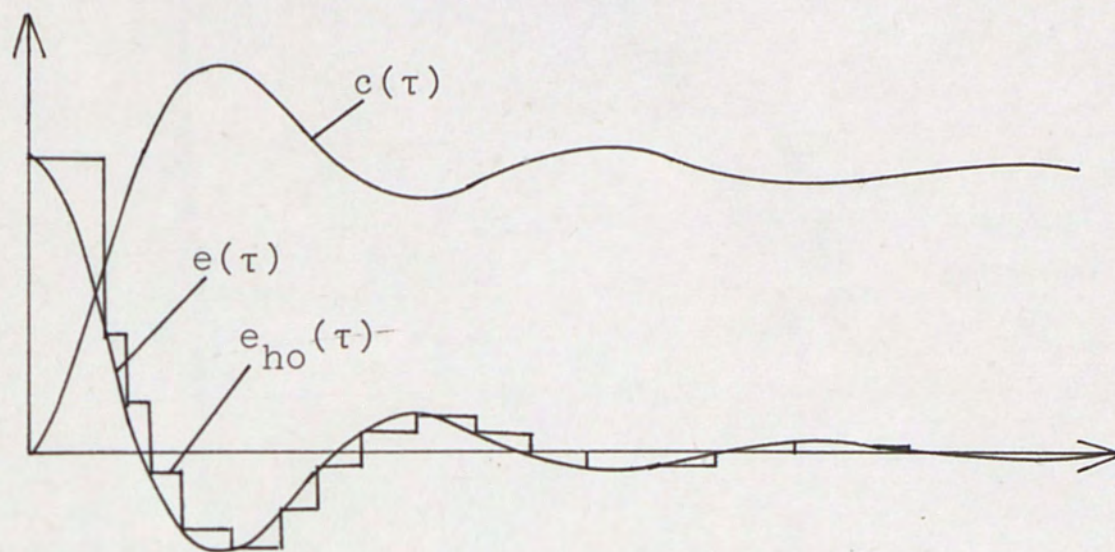
two figures. Figure 40(a) shows the system output, c , the continuous error, e , and the sampled error, e_n . The (b) part of Figure 40 is the adaptive time ramp, or the sample marker. Finally, in Figure 40(c), the derivative of the output, \dot{c} , the output of the hold circuit of the control law, \dot{c}_n , and the output of AMP 8, $B+K|\dot{c}_n|$ are depicted. Figure 41(a), (b), (c), and (d) shows the same system variables for the second case.

TABLE 1
PERFORMANCE DATA OF THE CONTROL LAW
WITH VARIOUS ORDER HOLDS

Order of Hold		Case 1			Case 2		
k_e	k_c	Normalized ISE	K_1	B_1	N_s	K_1	B_1
0	0	0.828	5.431	0.768	16.5	5.055	0.573
0	1	0.8325	4.845	0.755	15.6	2.403	0.607
1	0	0.935	0.654	0.743	15.7	0.871	0.717
1	1	0.925	1.045	0.688	14.3	1.451	0.624

System variables for the two cases when $k_e = 0$ and $k_c = 1$ are shown in the next two figures, Figures 42 and 43.

When the third condition shown in Table 1 was programmed, it was discovered that adaptive sampling did not help to reduce the ISE value when the number of samples, N_s , was set for 20. It was then determined that the system must



Adaptive Time Ramp

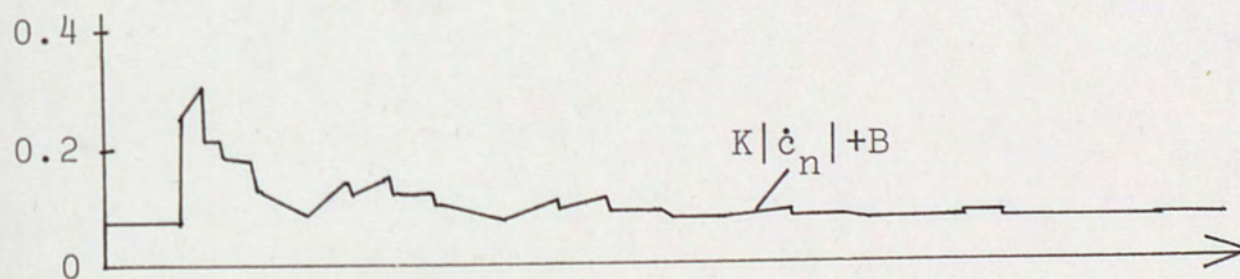
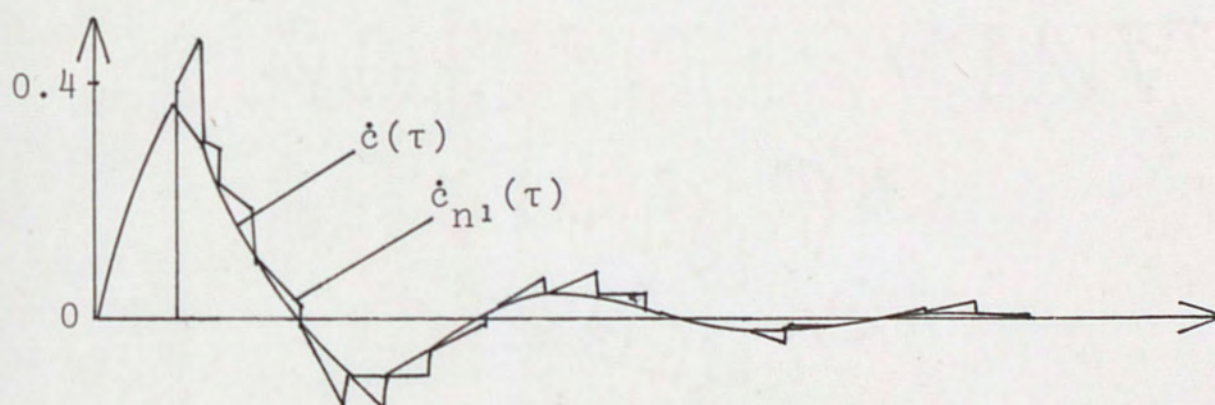


Figure 42. System Variables When $k_e = 0$, $k_c = 1$, $N_s = 20$ and Minimum ISE.

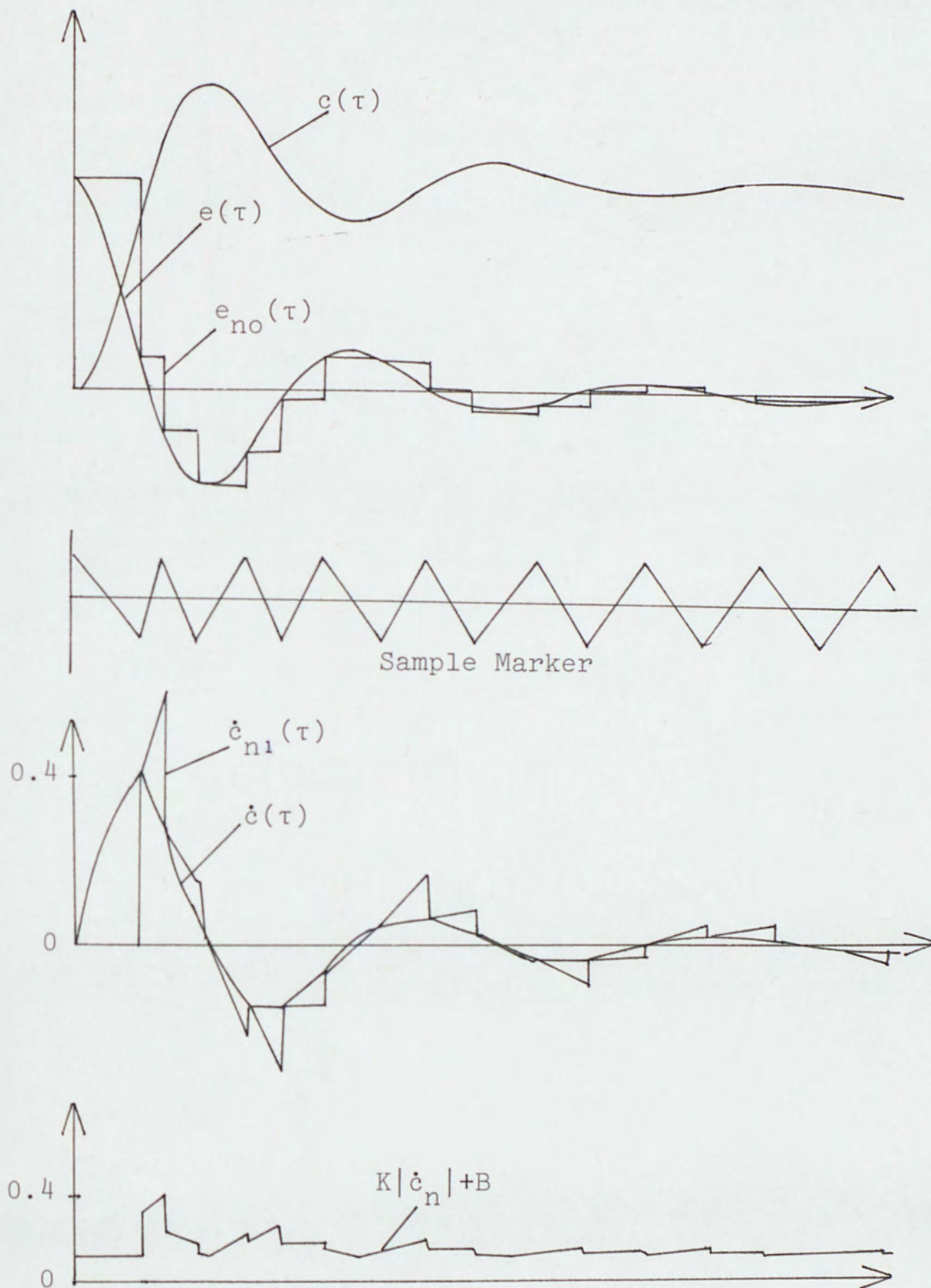


Figure 43. System Variables When $k_e = 0$, $k_c = 1$ and Minimum N_s .

have adequate sampling so that the ISE criteria could not be improved. The normalized ISE for this number of samples was 1.135. To prove that adaptive sampling would improve the value of ISE, the number of samples was reduced to 16. This produced an ISE of 0.935. The sample period (T) for the zero-order hold which is used for normalizing was 1.25 seconds. The other data was as shown in Table 1 with the system variables illustrated in Figures 44 and 45.

On programming the fourth set of conditions, problems again were encountered and the sample number was reduced to 15. This resulted in an ISE of 0.925 when again normalized about the ISE of a zero-order hold with a sample period of $4/3$ second. Figures 46 and 47 show the system variables for these conditions.

A microprogram was then generated to produce a true first-order hold. From Equation 9, Appendix B, it was determined that the second term of the defining equation for a first-order hold needed a $1/T$ multiplying factor in it. This factor would be multiplied by a $e^{kT}-e^{(k-1)T}$ factor which was already present in the existing program. This factor also would be fed into the program via potentiometers 3 and 5 so it would be multiplied by the outputs of AMPS 13, 33, 43, and 44. The microprogram to generate this $1/T$ factor is shown in Figure 48. The scaling for this program is shown in the figure and was selected so that there would be enough dynamic range in the program. POT 18 (T_0) was

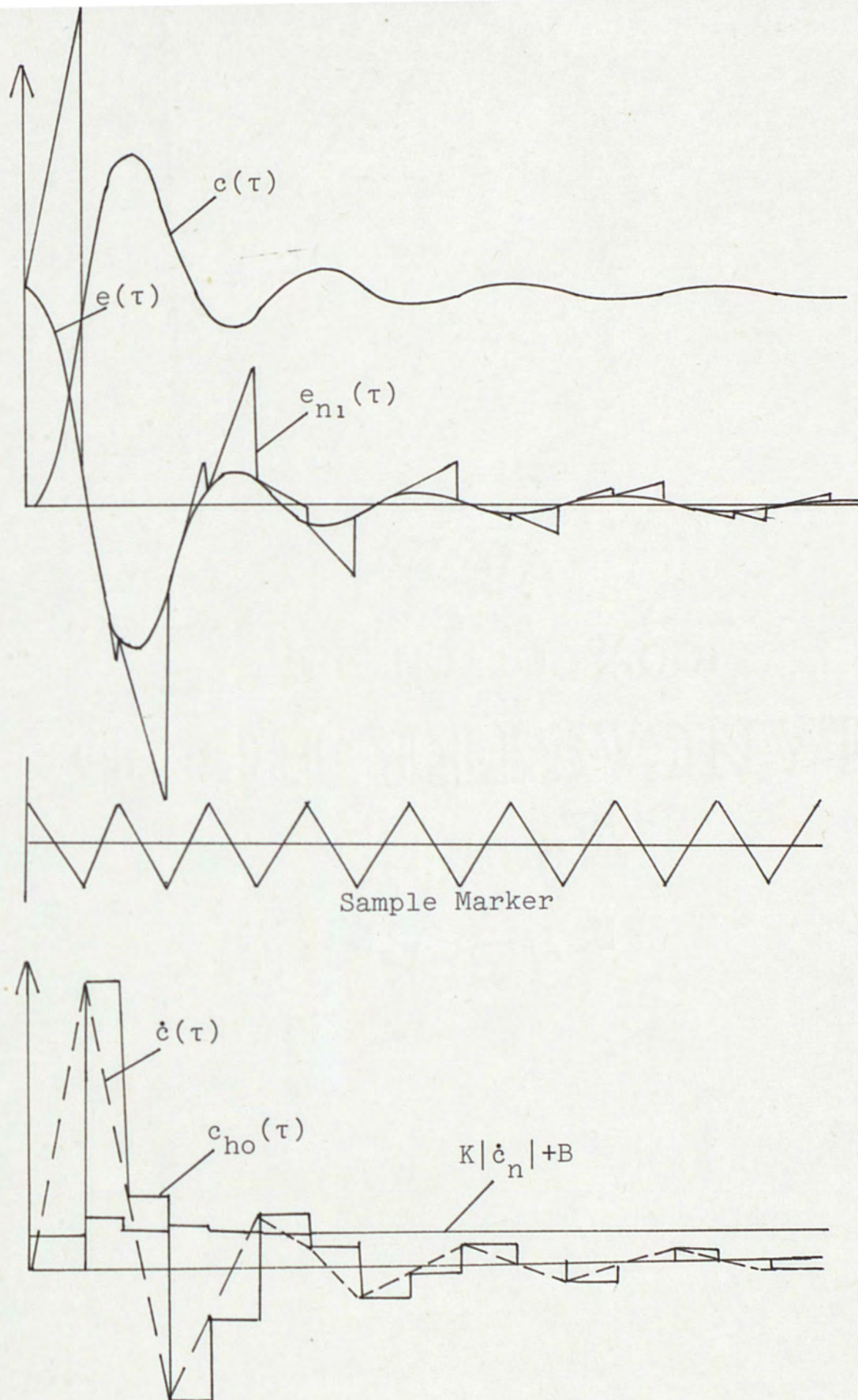


Figure 44. System Variables When $k_e = 1$, $k_c = 0$, $N_s = 16$ and Minimum ISE.

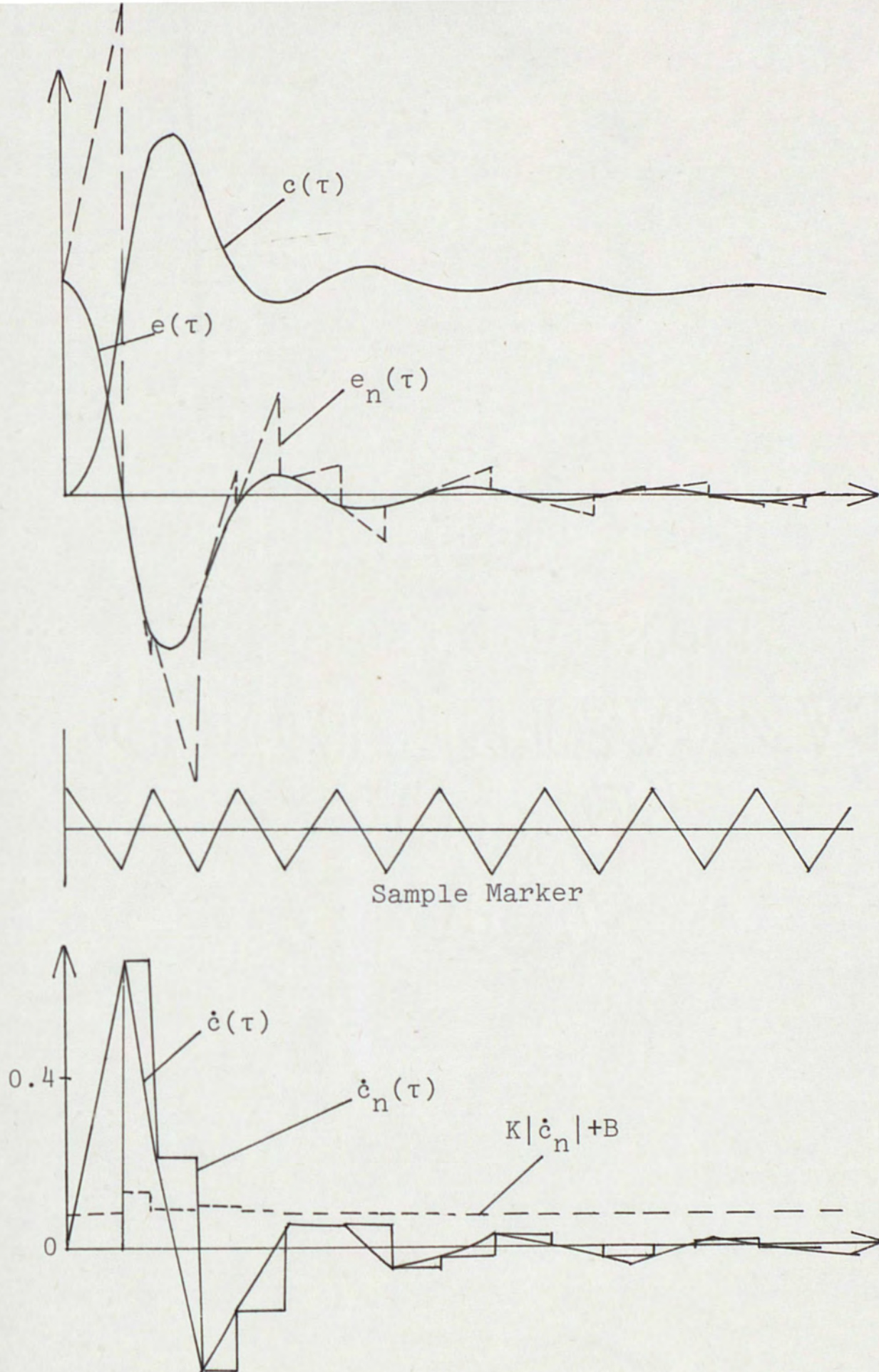


Figure 45. System Variables When $k_e = 1$, $k_c = 0$ and Minimum N_s .

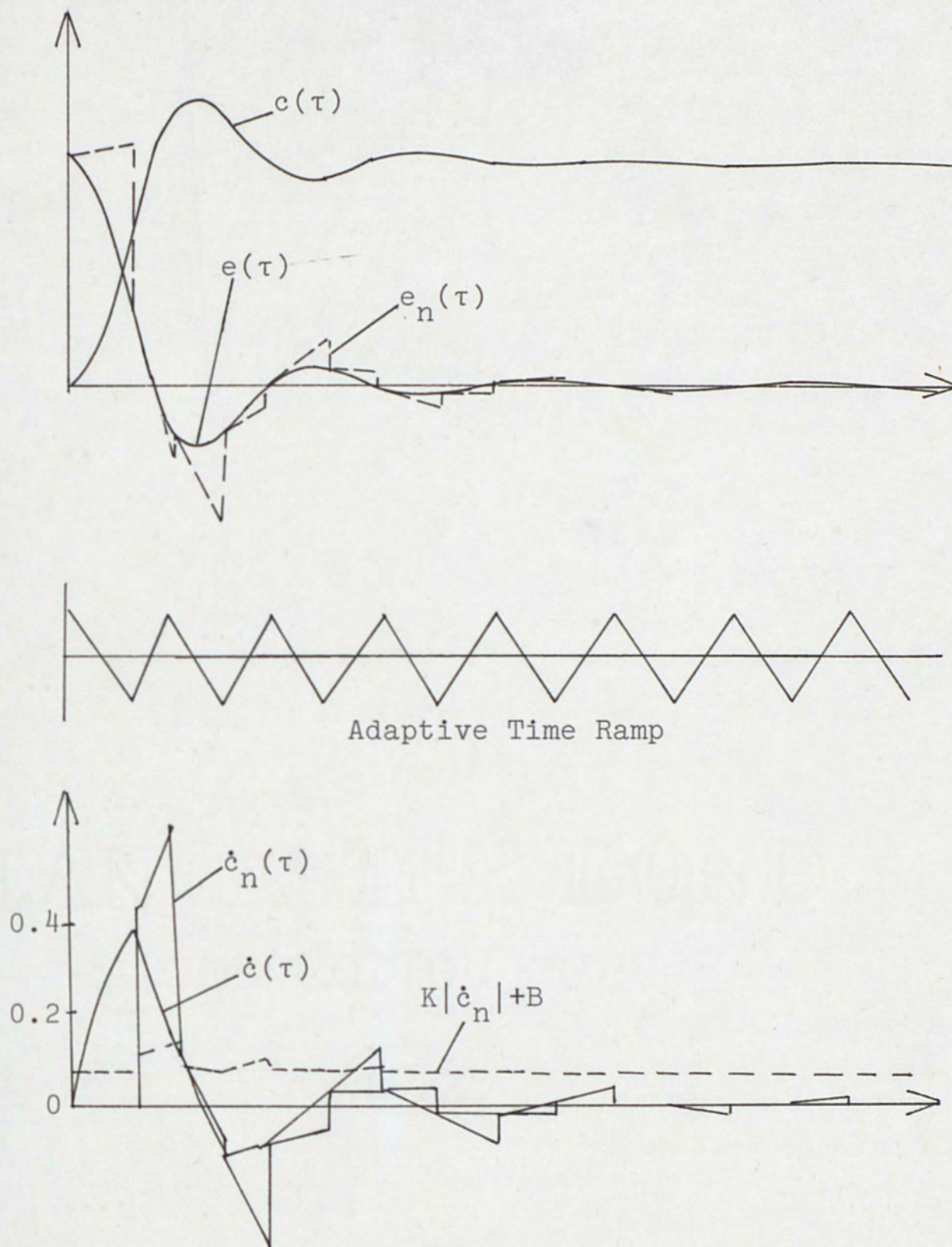


Figure 46. System Variables When $k_e = k_c = 1$, $N_s = 15$ and Minimum ISE.

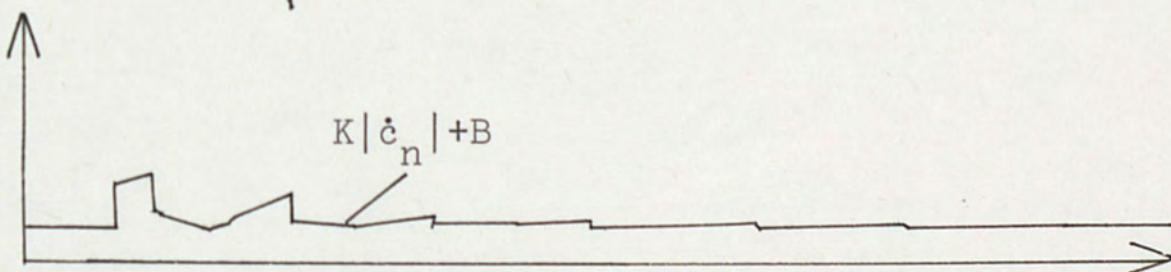
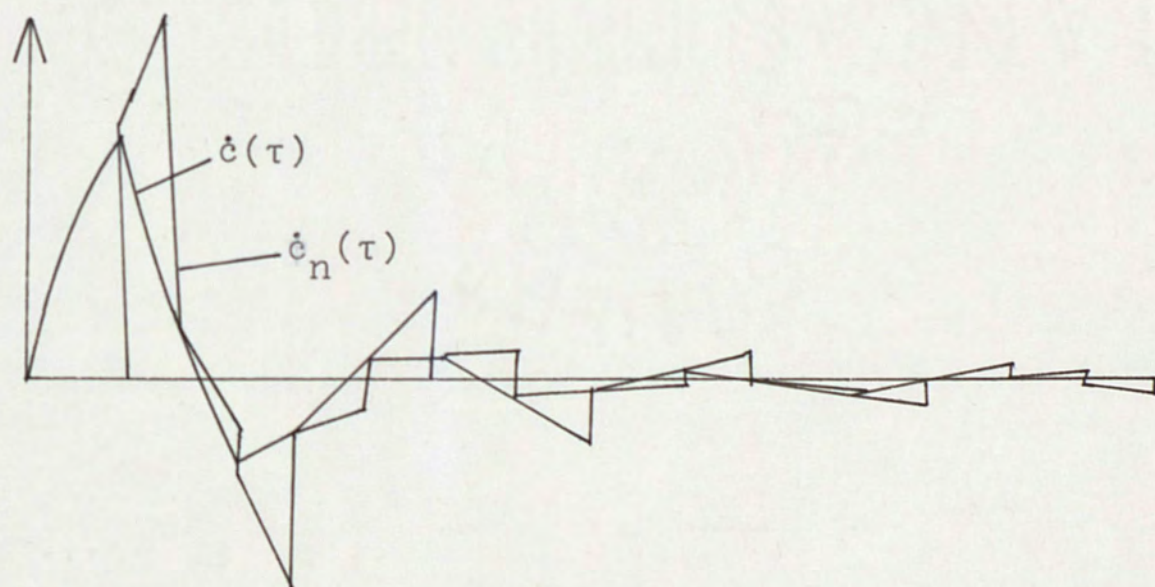
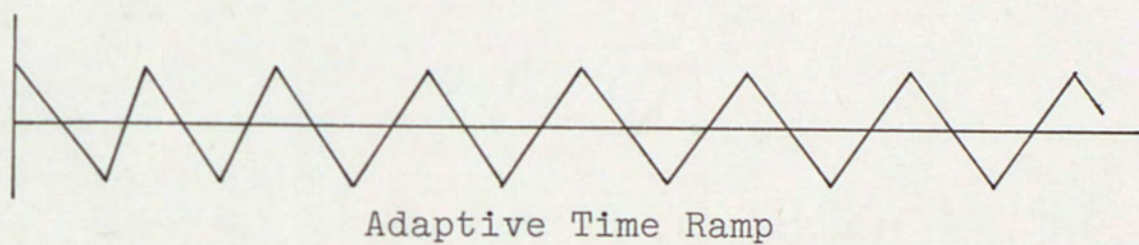
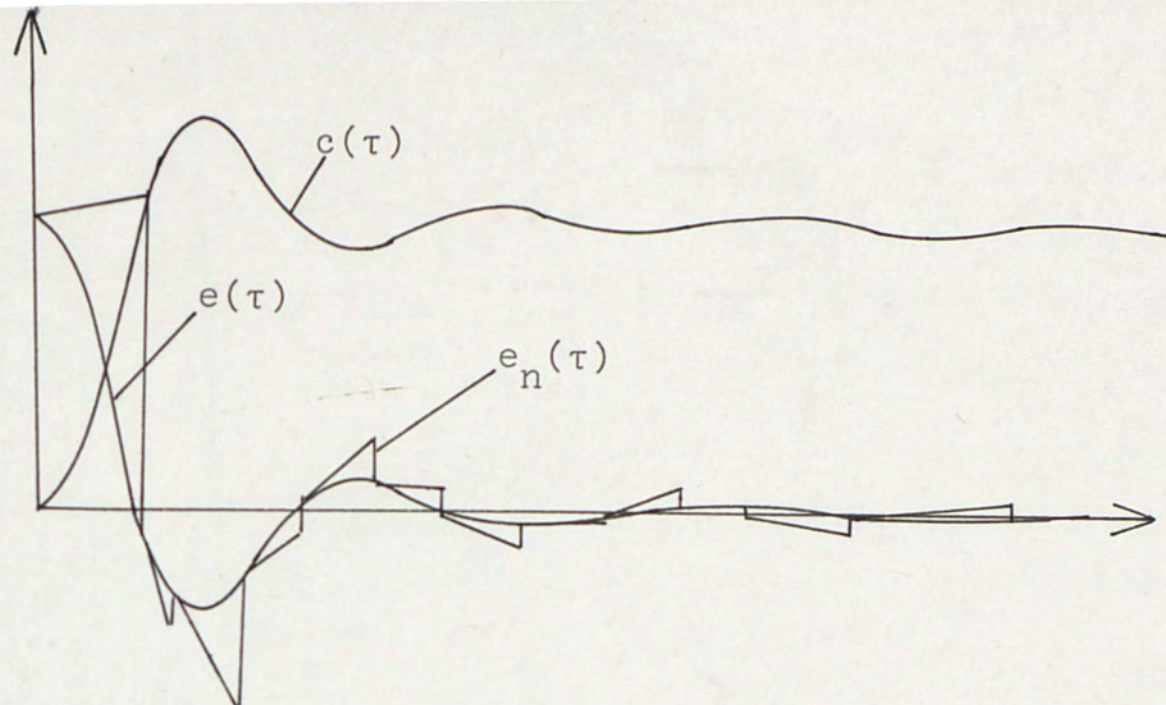
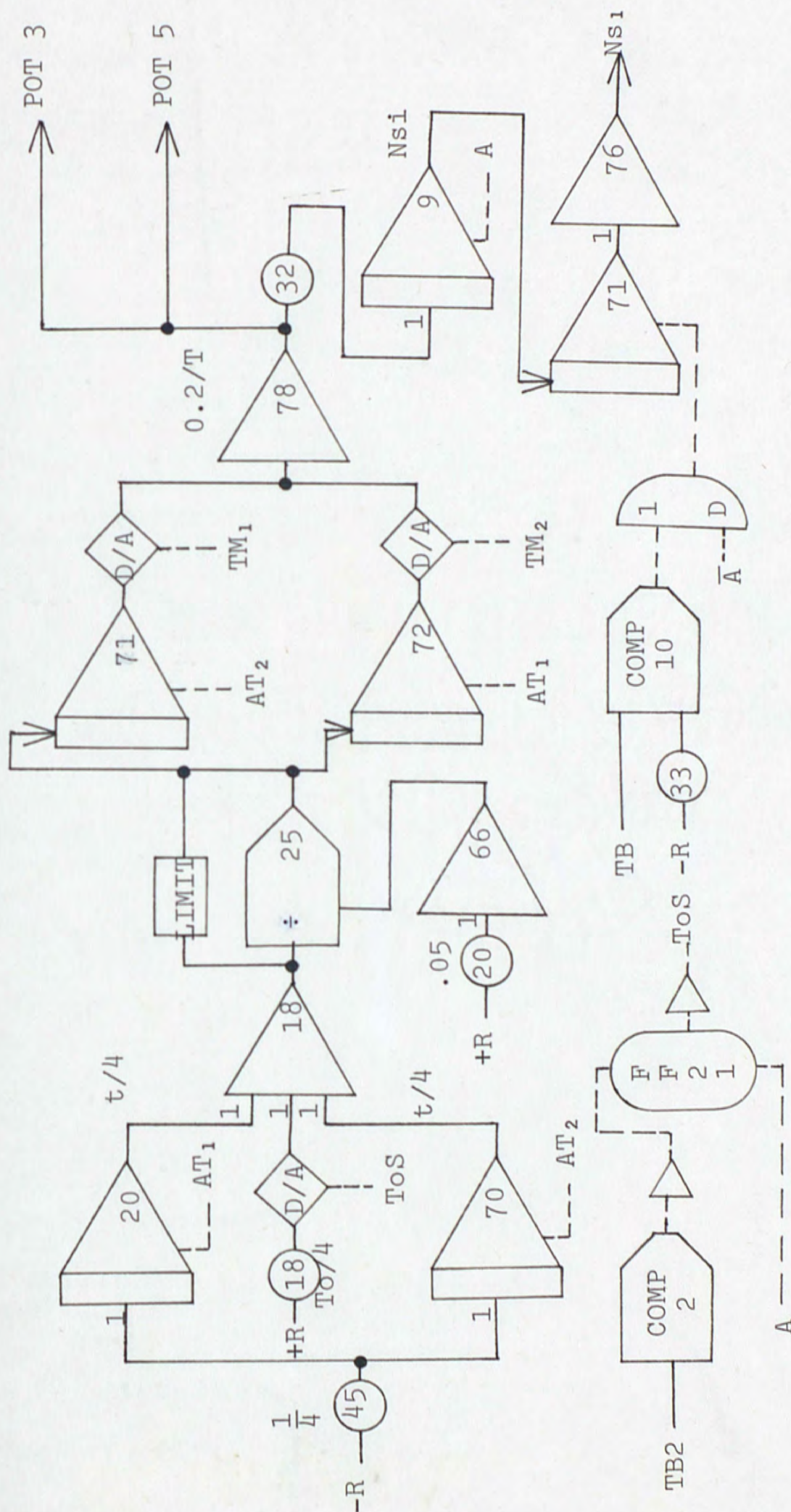


Figure 47. System Variables When $k_e = 1 = k_c$ and Minimum N_s

Figure 48. Microprogram to Produce $1/T_n$ Function

adjusted so that when the D/A switch was closed (IC), the sampling rate was at its maximum. Finally, T_{\min} was equal to 0.2 second and the k_e and k_c potentiometers were set for 0.5, and integrators 11, 31, 41, and 61 were set for a gain of 10.

A check was made with this true first-order hold program of some of the conditions in Table 1. In the $k_e = 1$ and $k_c = 0$ case there was no improvement as well as in the $k_e = 1$ and $k_c = 1$ situation. For the $k_e = 0$ and $k_c = 1$ situation the data is shown in Table 2.

TABLE 2

PERFORMANCE DATA OF THE CONTROL LAW WITH
A TRUE FIRST-ORDER HOLD

Case I			Case II		
Normalized ISE	K_1	B_2	N_s	K_1	B_1
0.8375	6.169	0.719	15.5	1.878	0.641

By comparing this data to the second line in Table 1 it can be seen that there was little difference, especially in ISE and N_s . Since there was a 22% change in K_1 and a 4 to 5% change in B_1 , it was decided to try to determine their sensitivity. This was accomplished by measuring ISE versus B_1 for various values of K_1 . The hold of the control law was held at a first-order while that of the error portion of

the system was varied from a zero to first-order. To accomplish this an amplifier had to be placed between the B_1 potentiometer and AMP 8, in Figure 35, and the negative reference input to the potentiometer had to be changed to positive reference voltage. These plots of this sensitivity study are shown in Figure 49(a) and (b).

Since a $1/T$ factor had been generated in Figure 48, a microprogram was added to produce an output which produced the total number of samples, N_s . This microprogram consists of AMPS 78, 9, 71, 76, COMP 10, and AND gate 1D shown in Figure 48. Amplifier 9 integrated the $1/T$ factor with respect to time and stored it in the IC input of integrator 71 until the unit went into an IC mode and then it was inverted by AMP 76. The system was scaled so that $N_{s(max)}$ was set for 50. Figures 50 through 54 show performance data for normalized ISE versus N_s for various values of K_1 . The orders of the two hold circuits were varied as well as B_1 for this data. As the value of K_1 increased, the ISE value began to reduce when both of the hold circuits were zero-order holds. For the conditions of $k_e = 1$ and $k_c = 0$, no value of K_1 would produce a value of ISE greater than the ISE for a zero-order hold, as can be seen in Figure 54.

This investigation led to another control law which was similar to Bekey and Tomovic's law except it was a function of the system error. That law was given by the following expression:

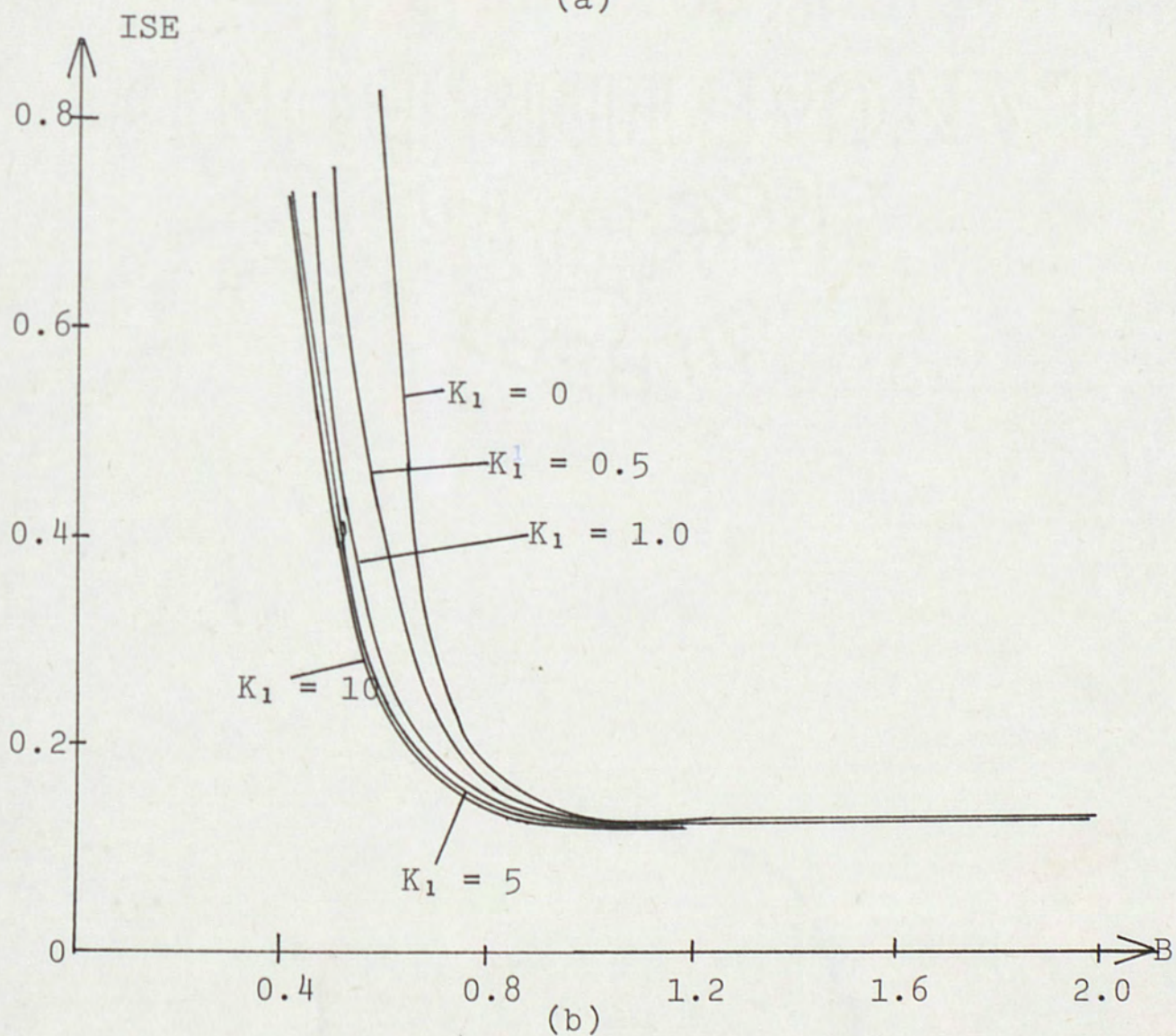
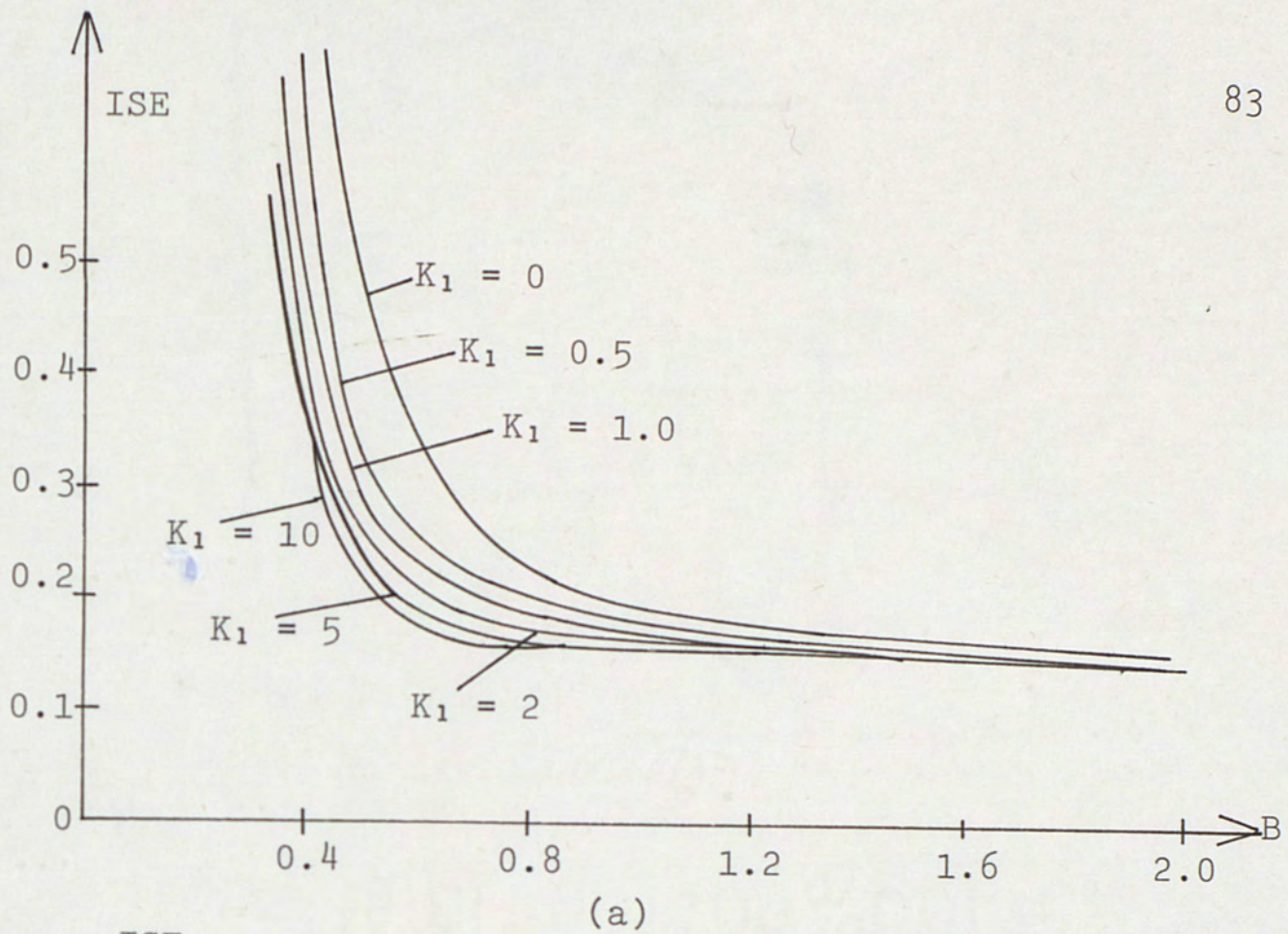


Figure 49. ISE Versus B

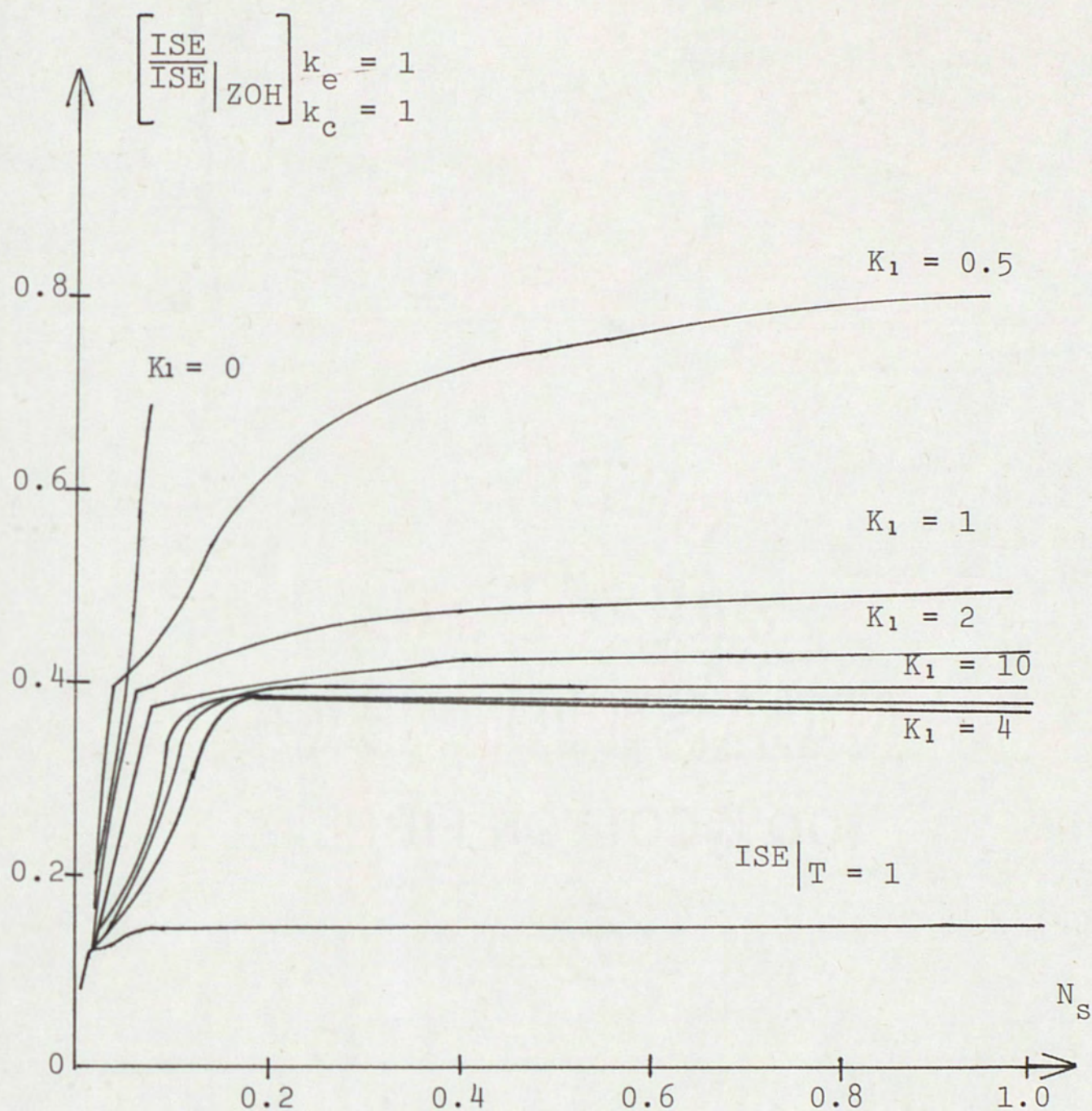


Figure 50. Normalized ISE Versus N_s for $k_e = 1$, $k_c = 1$ and T_{\max} .

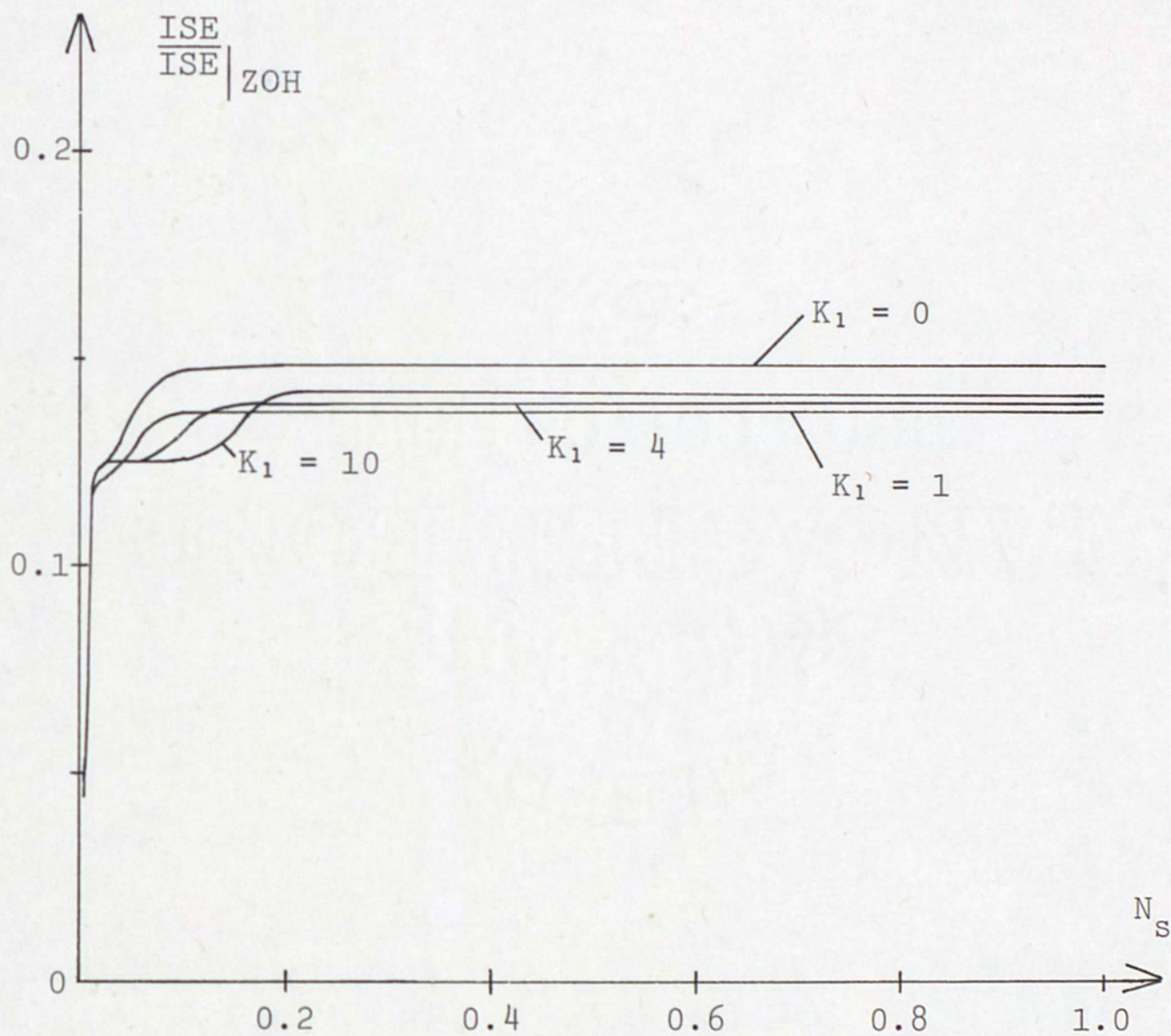


Figure 51. ISE Versus N_s for $k_e = 1 = k_c$; $B_1 = 0.1$

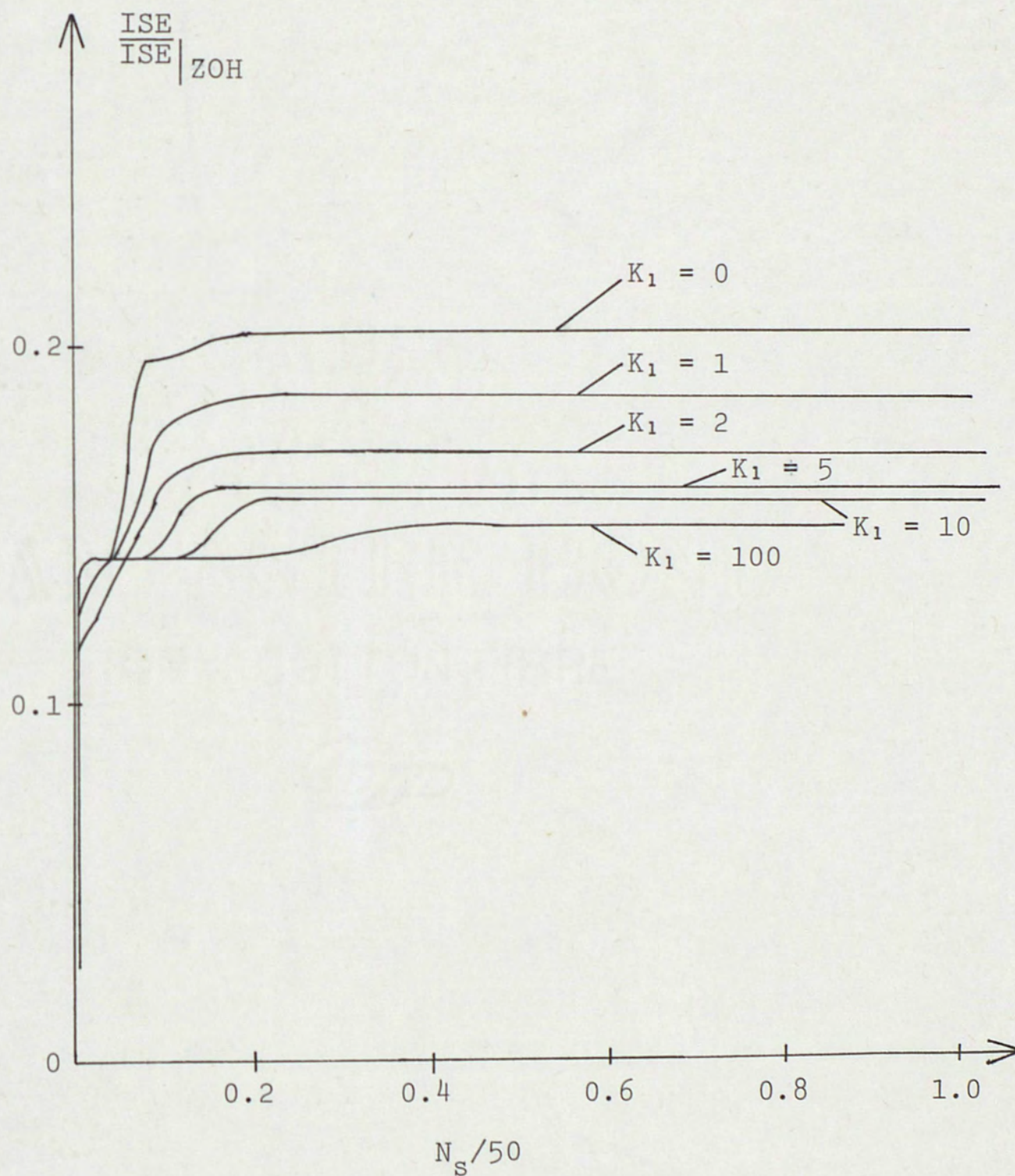


Figure 52. Normalized ISE Versus Number of Samples for $k_e = 0$, $k_c = 1$.

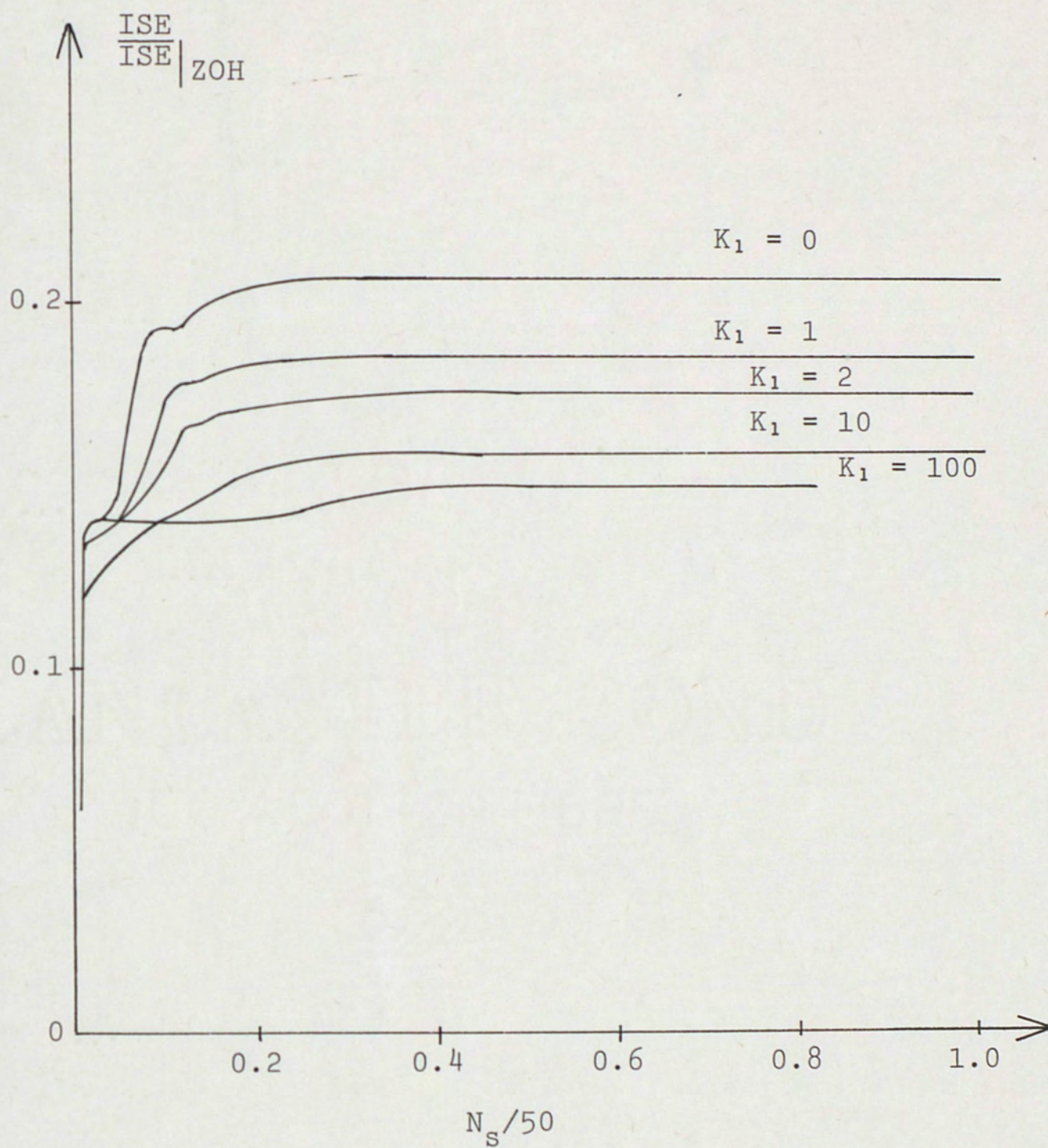


Figure 53. ISE Versus N_s for $k_e = k_c = 0$

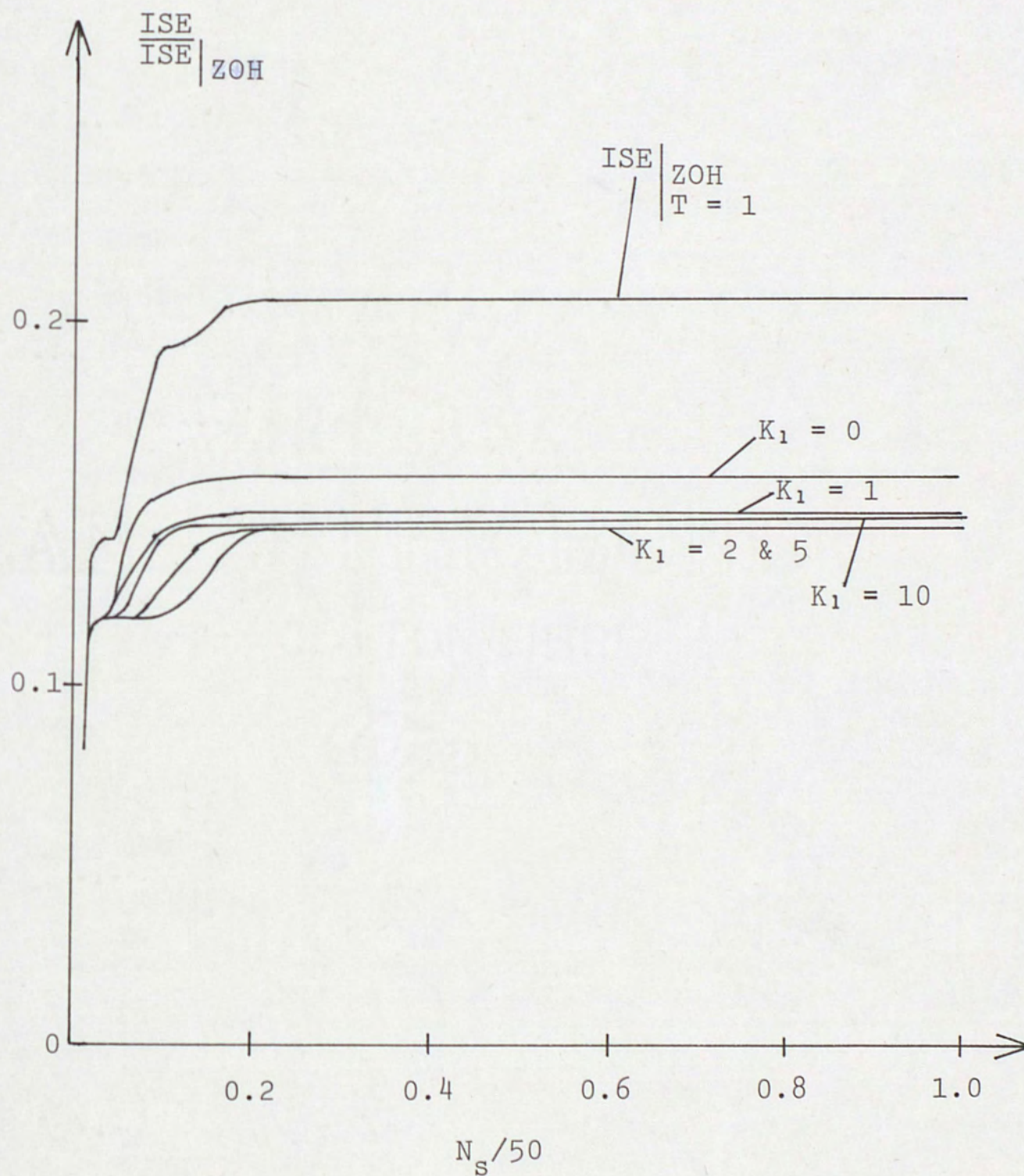


Figure 54. Normalized ISE Versus Number of Samples for $k_e = 1$, $k_c = 0$.

$$T = \frac{1}{K_1 |\dot{e}_n| + B} \quad (28)$$

where

$$\dot{e}_n = e_n - e_{n-1} \quad (29)$$

In the implementation of this law, the error function was the input to AMP 24 in Figure 35, rather than the derivative of the output. Figure 55 shows a normalized ISE versus number of samples, N_s , set of curves. Some of the system variables are shown in Figures 56 and 57 when K_1 was 0 and 10, respectively. In this control law as K_1 increased the ISE reduced for the same number of samples.

This control law then led to the development of another new control law that was defined by the following expression:

$$T = \frac{1}{K_1 |\dot{e}_n| + K_2 |e_n| + B} \quad (30)$$

where $|e_n|$ was created by taking the output of AMP 58, e_n , and going into a negative absolute value circuit and then into AMP 8 through a potentiometer to control the value of K_2 . ISE versus N_s data is shown in Figure 58 while Figures 59 and 60 show some of the system variables when K_1 and K_2 were 10 and B_1 was set for unity. It can be pointed out that this control law offered definite improvement when both K_1 and K_2 were equal to 10, and the number of samples

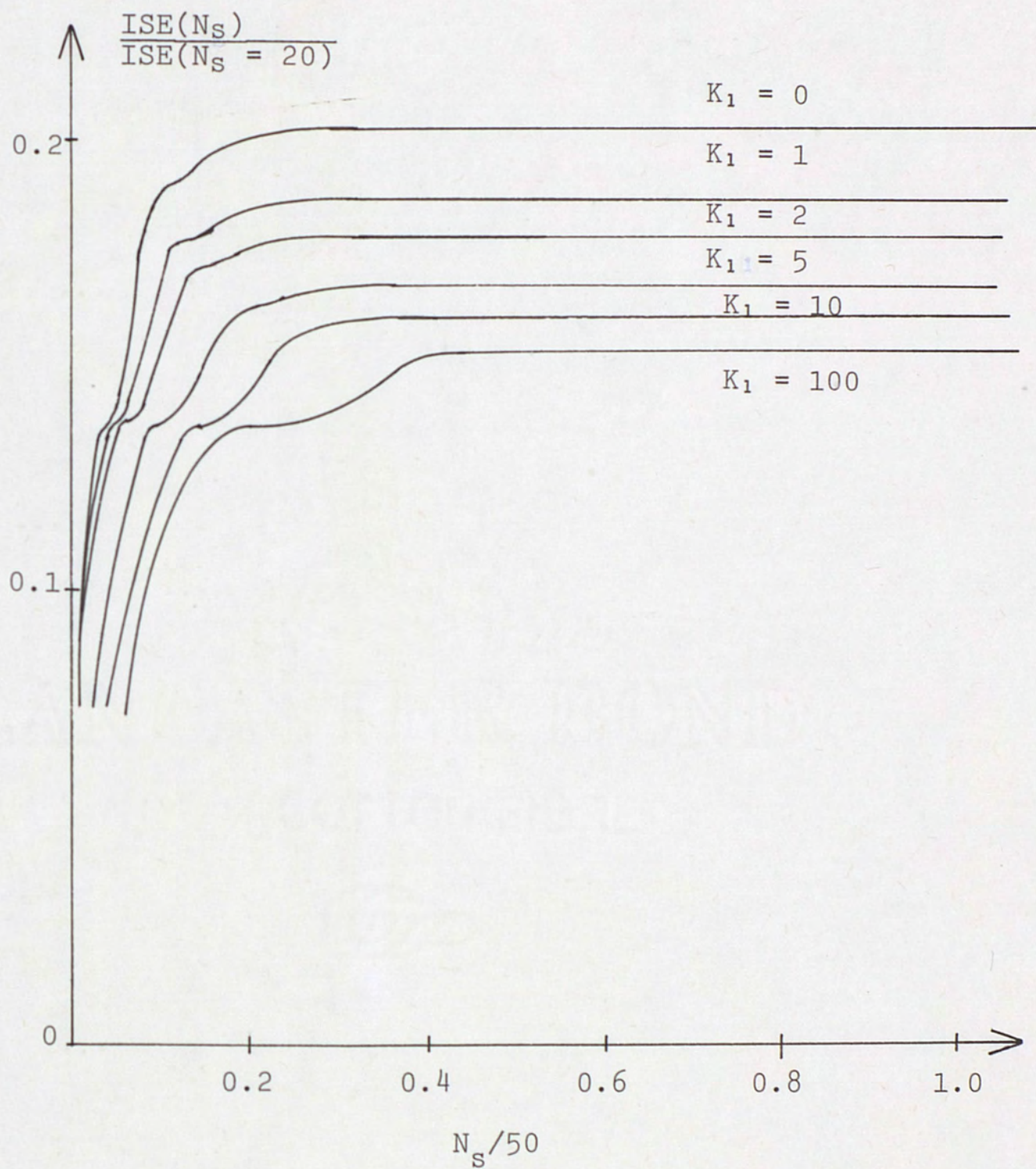


Figure 55. ISE Versus N_s for Control Law in Equation 28

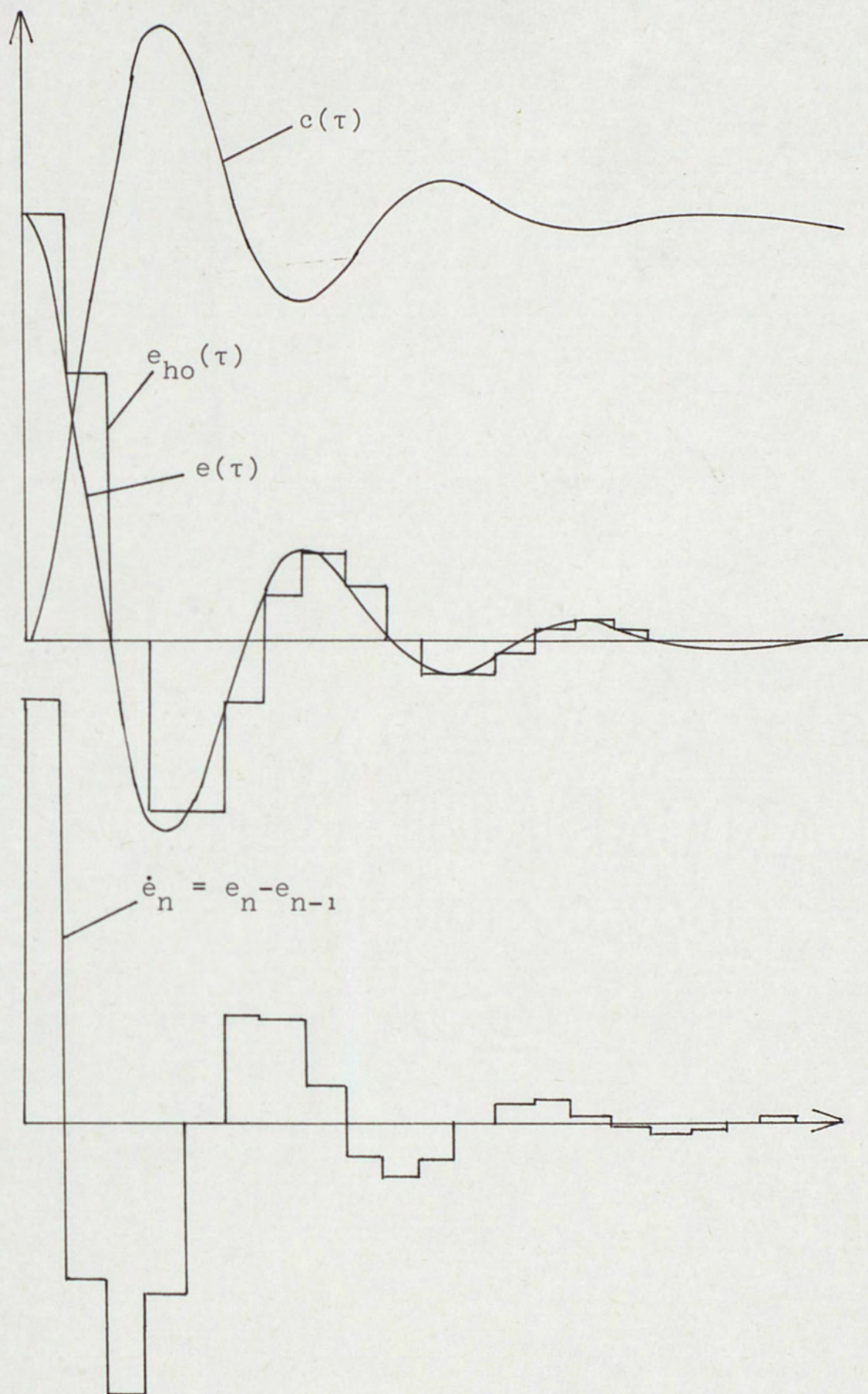


Figure 56. Equation 28 Control Law System Variable
When $K_1 = 0$.

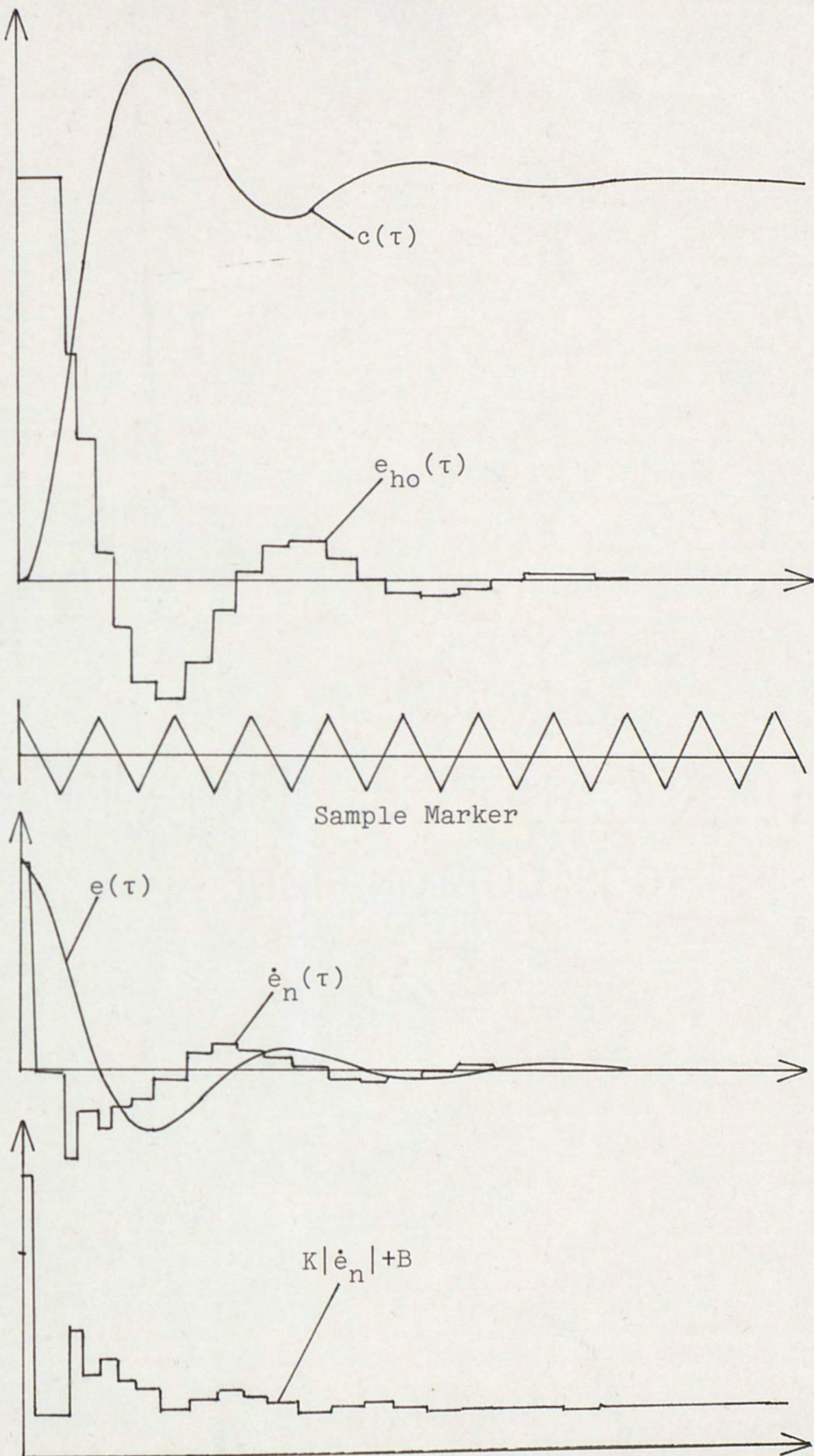


Figure 57. Equation 28 Control Law System Variables
When $K_1 = 10$.

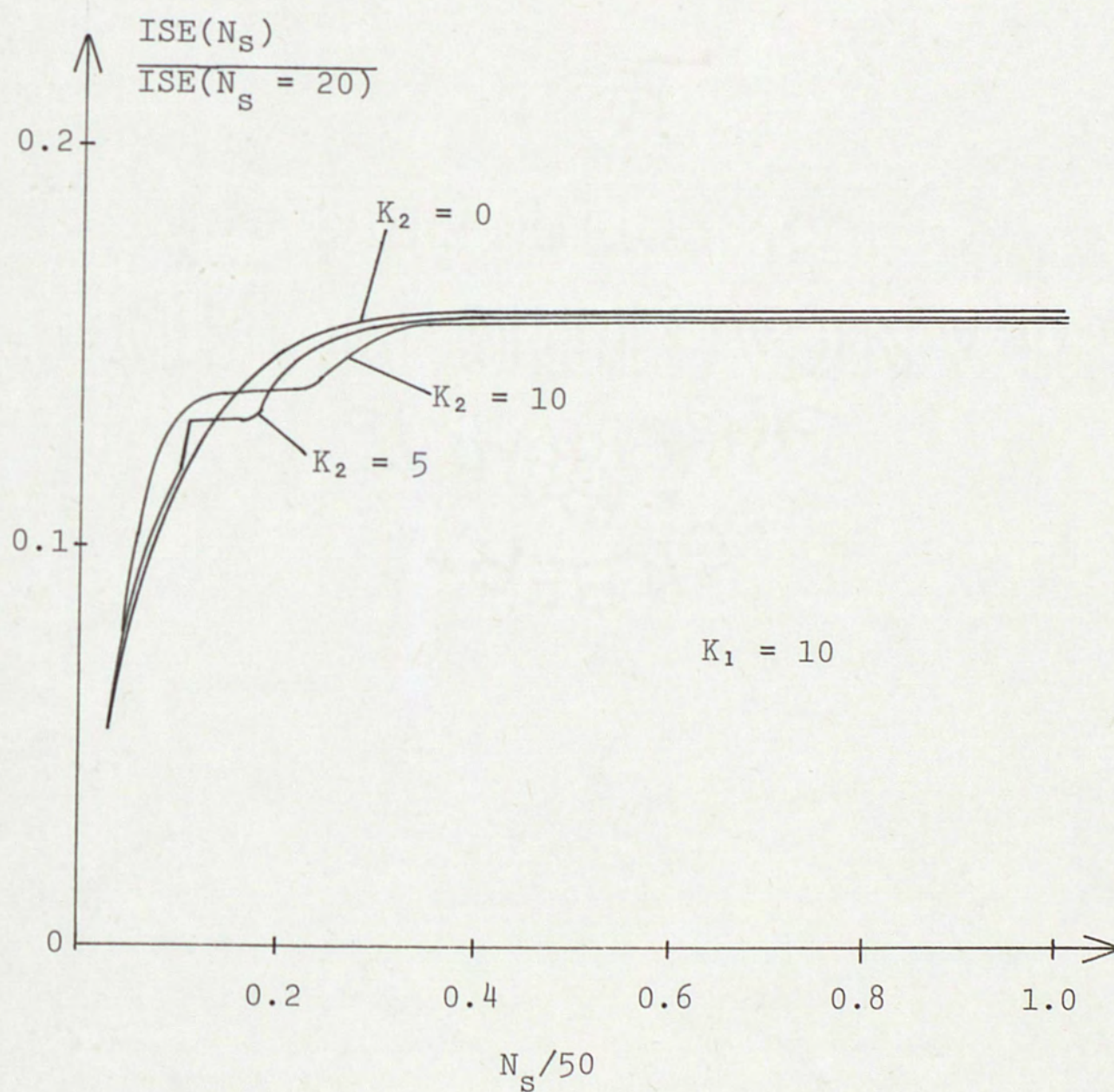


Figure 58. ISE Versus N_s for Control Law in Equation 30

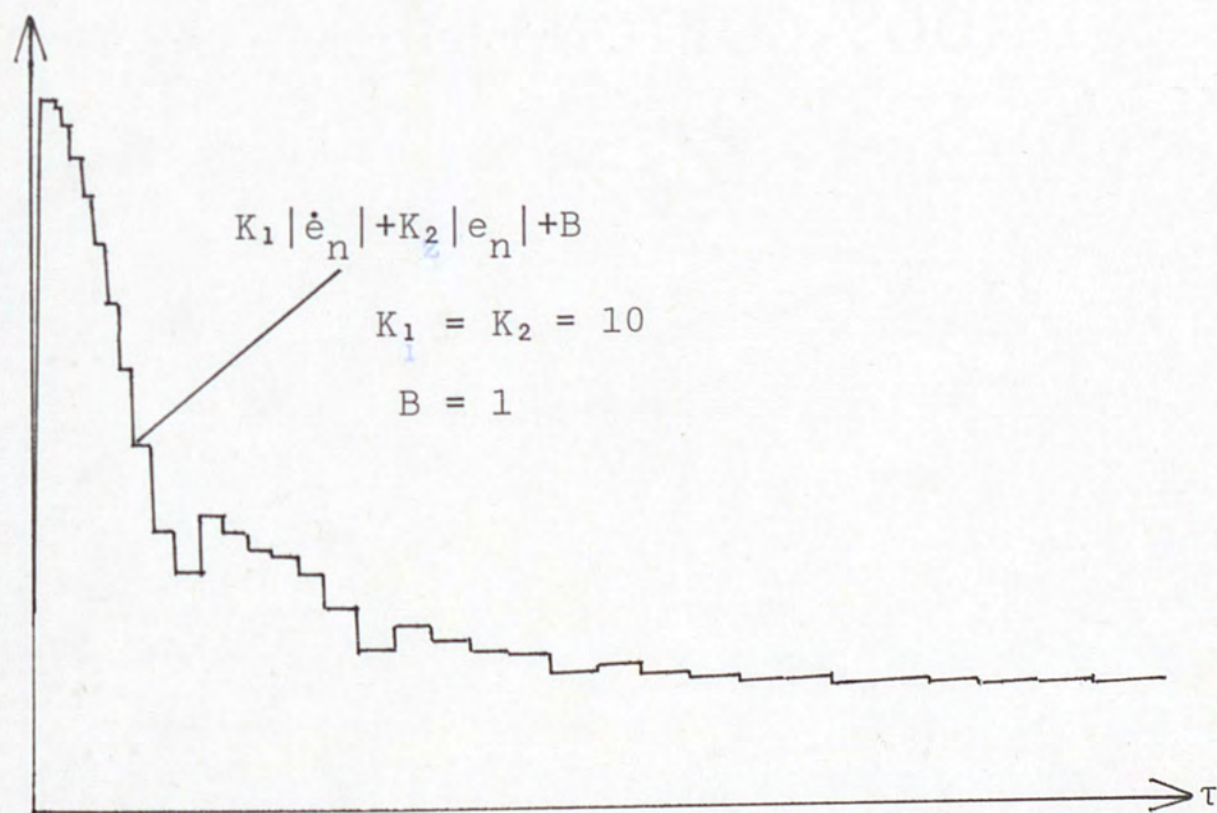
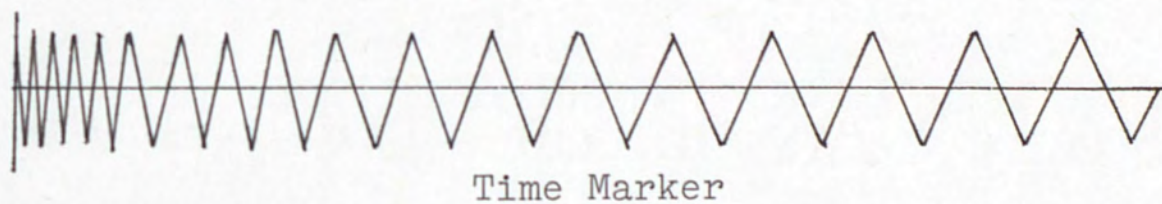
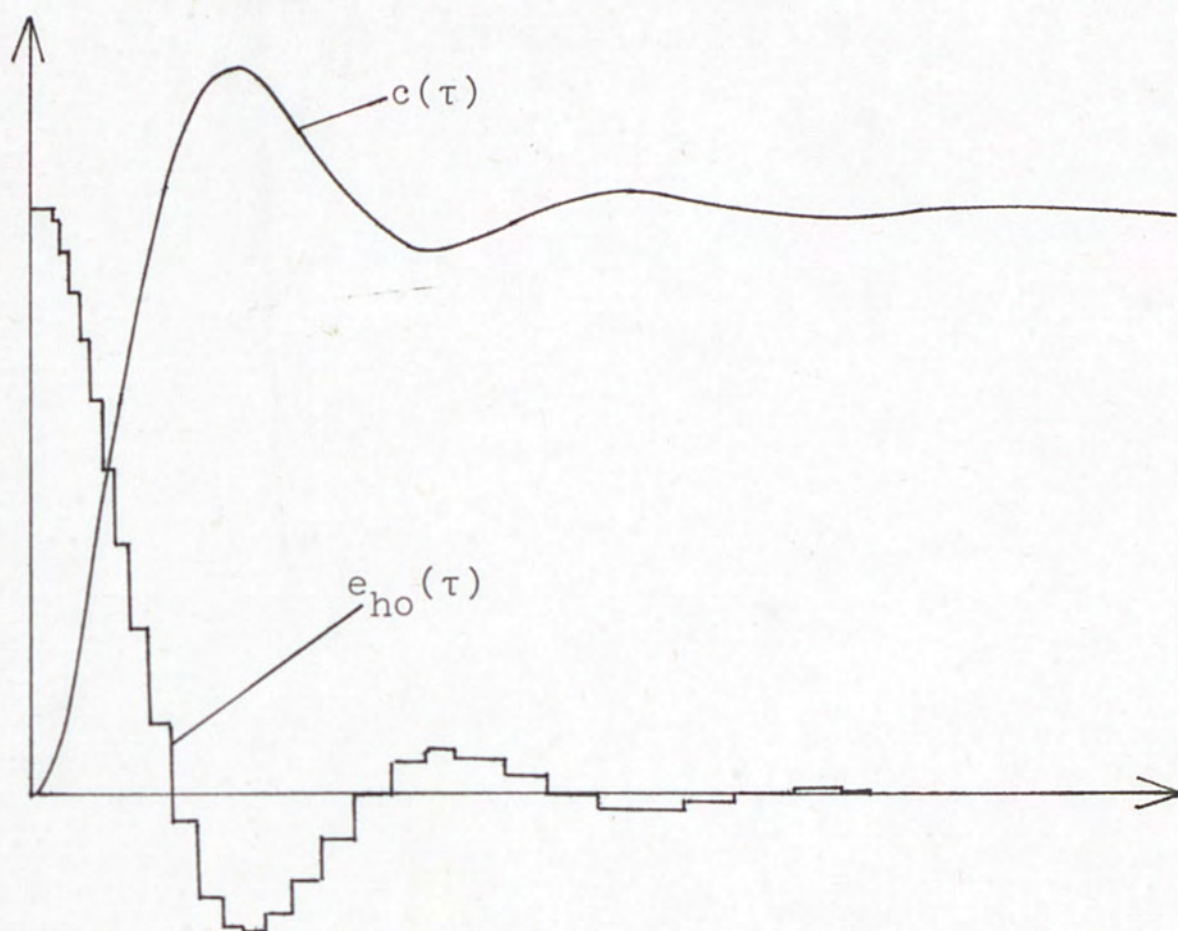


Figure 59. Equation 30 Control Law System Variables
When $K_1 = K_2 = 10$.

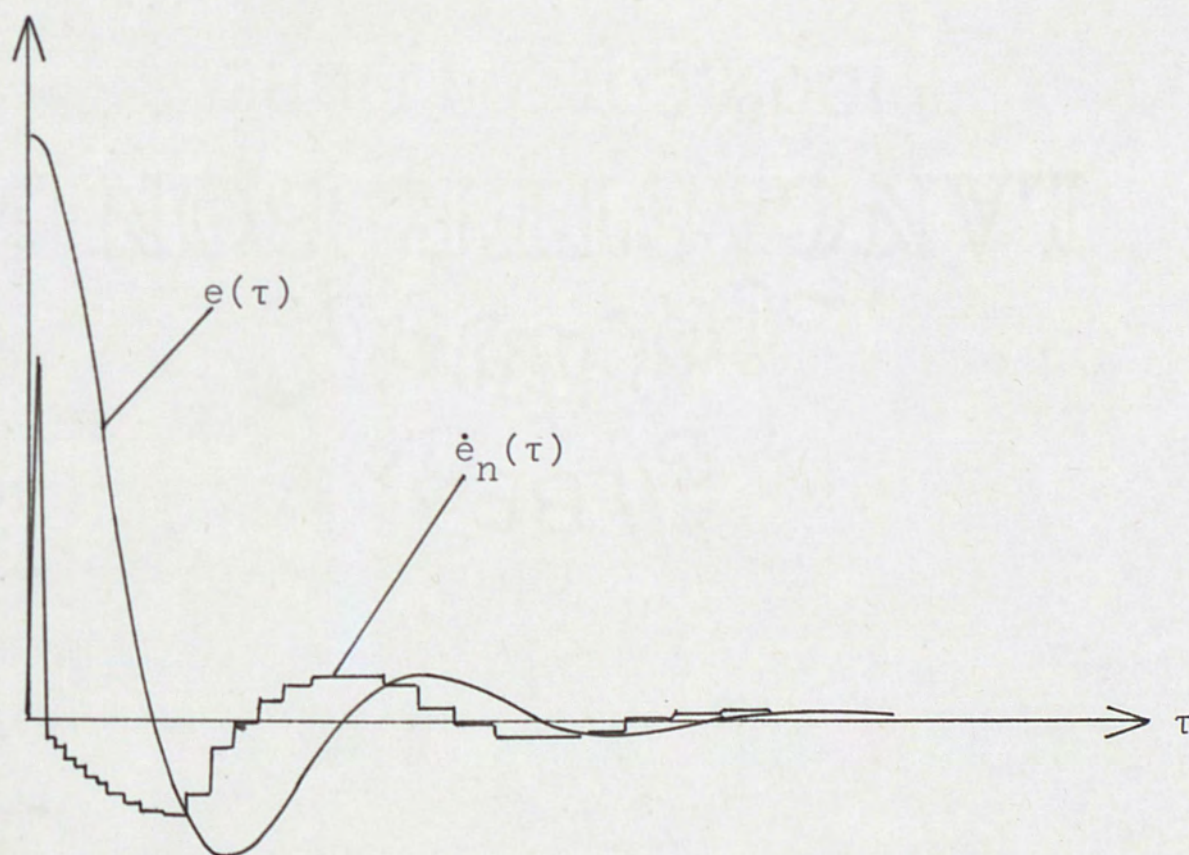


Figure 60. More Equation 30 Control Law System Variables When $K_1 = K_2 = 10$.

was around 10 to 12 which meant the sample period was approximately 1.6 to 2.0 seconds.

In the preceding sections the sampled-data systems were of the fixed frequency variety or in the case of adaptive sampling both samplers operated at the same rate. However, when a system contains two or more samplers, the sampling scheme is something other than all the samplers operating at the same rate. When a system has two or more samplers that operate at different rates, the system is called a multirate sampled-data system. All of the samplers operate at some multiple of the basic sampling rate. For instance, if the basic rate is taken as unity, some of the samplers could be operating at a rate which is two or even three times the basic sampling rate.

Figure 61 shows a typical open-loop block diagram of a system with an input sampler as well as an output sampler. This is a multirate sampling scheme due to the fact one of the samplers is operating at a rate which is N times faster than that of the other sampler. The samplers in this kind of operation are normally synchronized. In order to simplify the mathematics of analysis, it is also necessary to assume that N is an integer and the sample period of each sampler is measured by a common base line.

If the voltage control oscillator microprogram shown in Figure 33 is used as the generator of the basic sampling rate, an addition can be made which will result in a

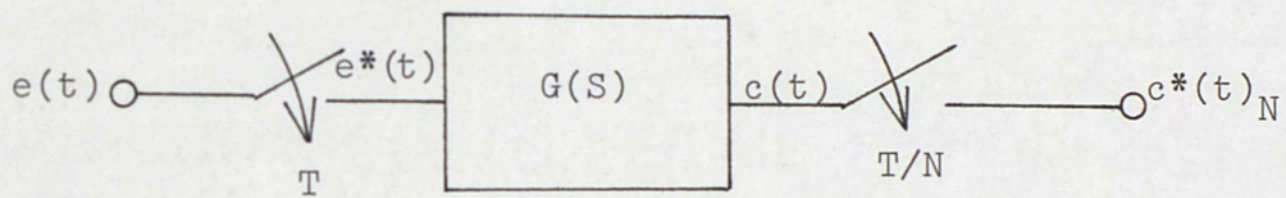


Figure 61. Open-Loop Block Diagram of a Multirate Sampling System.

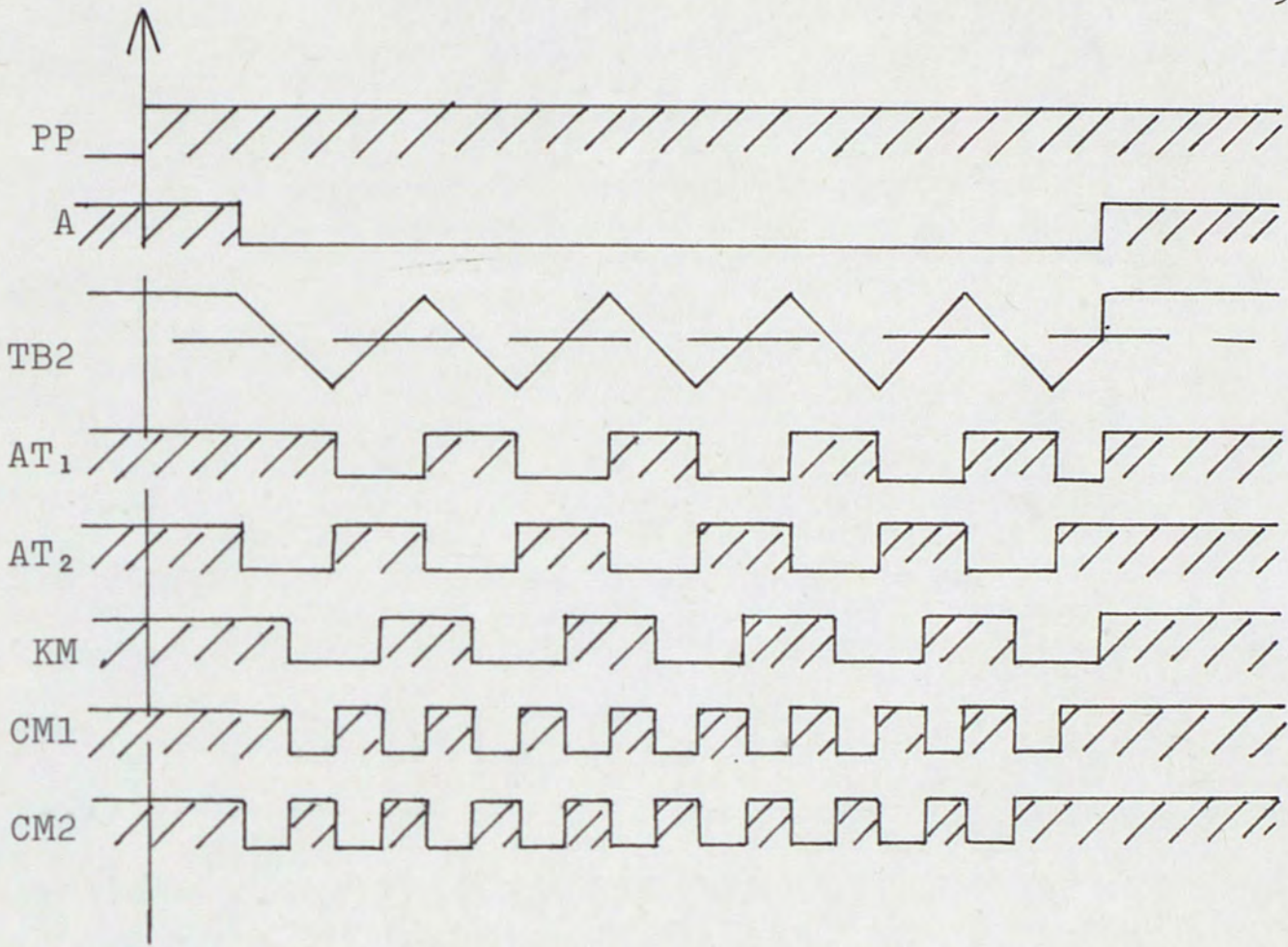
multirate sampling scheme. This multirate sampling addition is shown in Figure 62. The (a) part of the figure is the timing diagram, while the (b) part is the microprogram of the addition. COMP 1 detects when TB2 goes through zero. When TB2 is positive, the output of COMP 1 will be a logic one, and when it is negative, COMP 1 will go to a logic zero. The variable CM 1 can be defined by the following logic expression:

$$CM\ 1 = A + (KM \cdot \overline{AT2}) + (\overline{KM} \cdot AT2) \quad (31)$$

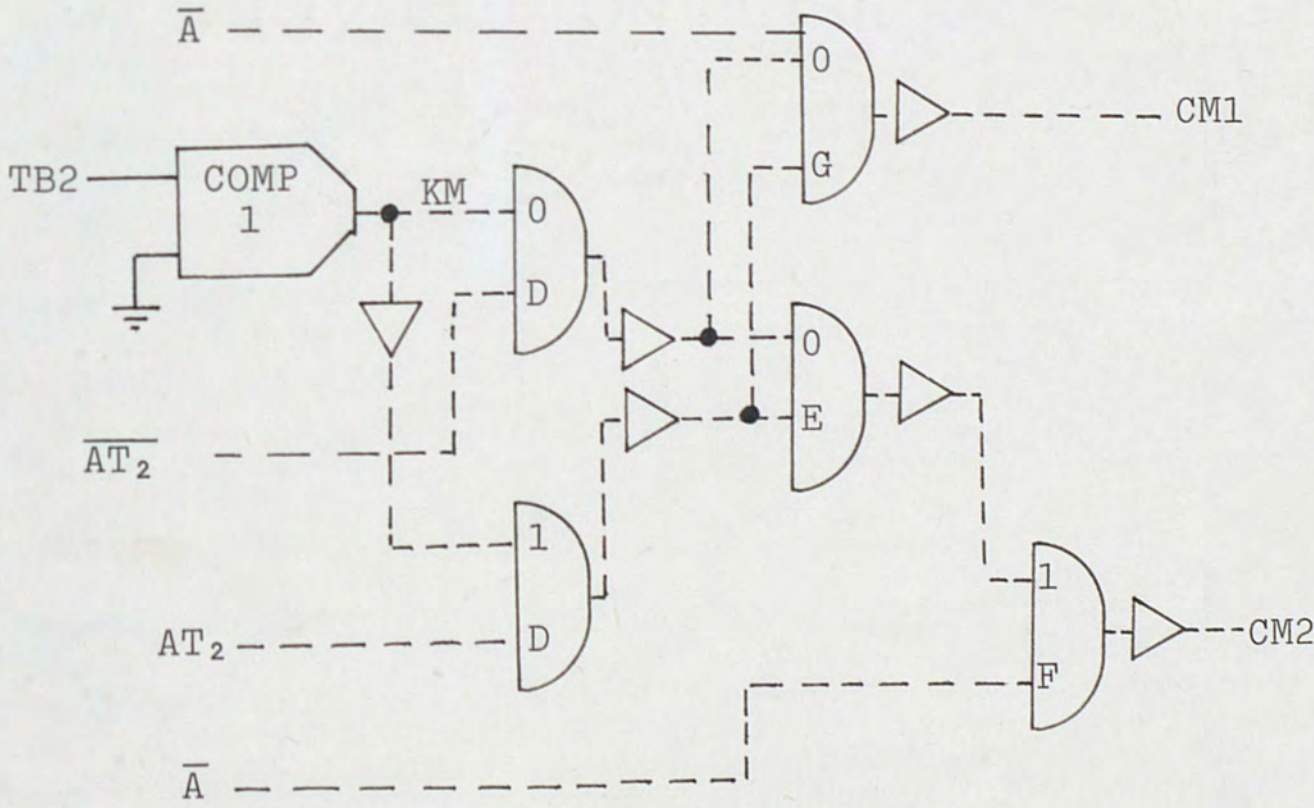
while the variable CM 2 is defined by the following:

$$CM\ 2 = \overline{A + (KM \cdot \overline{AT2}) + (\overline{KM} \cdot AT2)} \quad (32)$$

This system now has a signal, CM 1, which is twice the frequency of AT1. The same is true about CM 2's relationship with respect to AT2. Now, if the system used in the adaptive sampling portion of this chapter were to use these four logic signals as control, it would be a multirate system. This could be accomplished by allowing the error sampler to operate at the AT1 and the AT2 frequency rate while the control law sampler was operating at the CM 1 and CM 2 rate.



(a)



(b)

Figure 62. Multirate Sampling Addition to Figure 33

CHAPTER V

CONCLUSION AND RECOMMENDATIONS

This investigation has developed techniques which show some of the versatility and effectiveness of an analog/hybrid computer in the handling of various sampled-data control systems. Any skilled analog/hybrid analyst could simulate salient non-ideal characteristics of sampling devices. In fact actual sampling devices could be incorporated directly into critical simulations.

There are some good advantages to using the simulated time or update period technique that was developed. The main advantage was that computer hardware requirements are kept to a minimum but there is also a distinct disadvantage. This disadvantage is that the microprogram for updating the information in a hold device cannot be used with physical hardware. Due to the fact there are two different time bases, the result is a hardware and computer interface problem.

A parameter optimization program was conducted on Bekey and Tomovic's adaptive control law [4]. This control law was implemented with a second-order Type I control system. It was determined that the overall system was insensitive to

the K parameter of the control law but fairly sensitive to the constant, B . This sensitivity was determined using the integral squared error (ISE) as a figure of merit for the system.

Two new adaptive control laws were proposed and investigated. Although both of the laws are basically similar to Bekey and Tomovic's law, they are functions of different system variables. The first law is a function of the difference of two successive samples of the error signal--a measure of the rate of change of the error signal. The second control law is also a function of this difference of successive samples and also the output of the hold device of the error portion of the system. No improvement could be determined from the first law while the second control law indicated definite improvement when the constants K_1 and K_2 were equal to 10 and the sample period is approximately 1.6 to 2.0 seconds.

This investigation prompts several recommendations for further study. The first would be to study all of the published adaptive control laws using the simulation techniques developed here and also to utilize first-order holds instead of the conventional zero-order hold devices. The next suggestion would be to extend this method to a higher order system not only for adaptive sampling but also for the fixed frequency sampling portion. Another recommendation would be to implement the Bekey and Tomovic adaptive control

law and operate the system in a multirate sampling fashion using the multirate sampling technique proposed in this study. In this multirate sampling scheme the error sampler would operate at one rate and the control law would operate at a second rate and a parameter optimization program could be undertaken. This approach could also apply to the two control laws that were proposed. Finally it is recommended that attempts be made to modify the data reconstruction techniques proposed by Bullock and Durling [10] so they can be utilized in the simulation of sampled-data systems.

APPENDIXES

APPENDIX A

ANALOG COMPUTER MODEL

A unity-feedback, error-sampled system was chosen as the model for this particular study. The open-loop transfer function of the model is:

$$G(S) = \frac{K}{S(S+a)} \quad (1)$$

$$\text{where } K = \omega_n^2 = 1 \text{ and} \quad (1A)$$

$$a = 2\delta\omega_n = 1.0 \quad (1B)$$

The closed-loop transfer function for such a system is defined as thus:

$$\frac{C(S)}{R(S)} = \frac{G(S)}{1+G(S)} \quad (2)$$

Substitute equations (1), (1A) and (1B) into equation (2)

$$\frac{C(S)}{R(S)} = \frac{K}{S^2+aS+K} = \frac{\omega_n^2}{S^2+2\delta\omega_n S+\omega_n^2} \quad (3)$$

The differential equation put in the time domain notation, with zero initial conditions, is the familiar equation:

$$\ddot{c}(t) + 2\delta\omega_n \dot{c}(t) + \omega_n^2 c(t) = \omega_n^2 r(t) \quad (4)$$

The damping ratio (δ) was chosen to have a value of 0.5. This particular value provided a good compromise in the overall system response. The natural frequency (ω_n) then was selected to be 1 radian per second so that the X-Y plotter display could be made easily. In the time scaling of the system model, simplicity was another primary reason for this choice. It also becomes readily apparent if one thinks of time scaling a model having any other natural frequency, ω_n , by letting τ , the computer time, equal $\omega_n t$.

Then,

$$\dot{c}(t) = \omega_n \dot{c}_n(\tau), \text{ and} \quad (5)$$

$$\ddot{c}(t) = \omega_n^2 \ddot{c}_n(\tau) \quad (6)$$

and then equation (4) becomes

$$\omega_n^2 \ddot{c}_n(\tau) + 2\delta\omega_n^2 \dot{c}_n(\tau) + \omega_n^2 c_n(\tau) = \omega_n^2 r(\tau). \quad (7)$$

The ω_n^2 cancel out, leaving

$$\ddot{c}_n(\tau) + 2\delta\dot{c}_n(\tau) + c_n(\tau) = r(\tau), \quad (8)$$

which turns out to be the computer model with a natural frequency of 1 radian/second. Therefore, any "real time" time period may be found by taking the model time period and dividing it by the natural frequency of the system.

APPENDIX B

DATA RECONSTRUCTION

In most feedback control systems employing sampled-data, a data-reconstruction device must be used to smooth the digital signal before it is applied to the continuous-data components. The output of this data-reconstruction device is considered to be a function of the continuous time variable, t . A compromise between stability requirements and the desire of a close approximation of the continuous signal will usually result in the construction of such a filtering device.

Since the continuous signal is to be constructed based on information available only at past sampling instants, the reconstruction process will be an extrapolation process. Ragazzini and Zadeh [11] generated a method of obtaining this desired approximation based on the power series expansion of $e(t)$ between the sampling instants kT and $(k+1)T$.

$$e_k(t) = e(kT) + e^{(1)}(kT)(t-kT) + \frac{e^{(2)}(kT)}{2!}(t-kT)^2 + \dots \quad (1)$$

$$\text{where } e_k(t) = e(t) \text{ for } kT \leq t < (k+1)T \quad (2)$$

$$e^{(1)}(kT) \triangleq \left. \frac{de(t)}{dt} \right|_{t = kT} \quad (3)$$

$$e^{(2)}(kT) \triangleq \left. \frac{d^2 e(t)}{dt^2} \right|_{t = kT} \quad (4)$$

For a polynomial of k th order, at least $(k+1)$ prior samples are required for perfect signal reproduction. When an insufficient number of samples is used, an error results but this error is usually less than the error that results when no extrapolation is used.

Data reconstruction devices which are commonly called hold devices, can now be classified by the number or order of the terms in equation (1) which are included in the extrapolation process. If only the first term is used, it is called a zero-order hold since the polynomial used is of the zeroth order. A system using the first two terms has a first-order hold because the polynomial will be of the first-order and finally a system with $(k+1)$ terms has a k th-order hold.

Since a hold is a block within the closed-loop path of most control systems, a transfer function is required so that its effects in a dynamical system can be evaluated. The transfer function of a linear system is the Laplace transform of the impulse response of the system. With the hold transfer function and also the transfer function of the plant, the overall system response can then be determined.

Zero-Order Hold

As previously stated when only the first term of the power series is used, the reconstruction device is called a zero-order hold. Its equation would be simply:

$$e_k(t) = e(kT) \quad (5)$$

which defines the impulse response as shown in Figure 63.

Figure 64 shows that the impulse response of the zero-order hold can be broken into two unit step functions. The response then can be given by

$$g_h(t) = u(t) - u(t-T) \quad (6)$$

where $u(t)$ is the unit step function. The Laplace transform of equation (6) is then:

$$G_{ho}(S) = \frac{1}{S} - \frac{1}{S}e^{-Ts} \quad (7)$$

$$G_{ho}(S) = \frac{1-e^{-Ts}}{S} \quad (8)$$

First-Order Hold

Now if the first two terms are used to evaluate the time function (t) , a first-order hold is created. Its defining equation would be

$$e_k(t) = e(kT) + \frac{e(kT) - e(k-1)T}{T}(t-kT) \quad (9)$$

Its response to a unit impulse is shown in Figure 65. The transfer function is derived directly from the impulse response and is

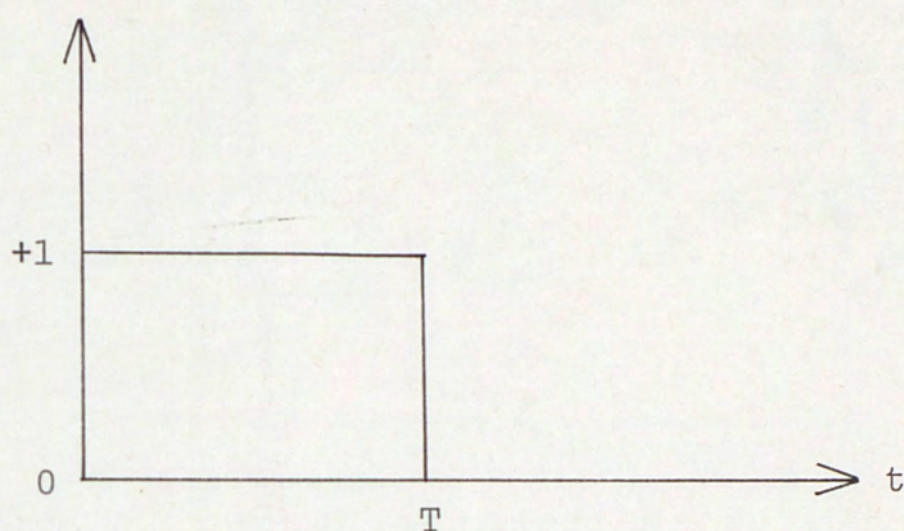


Figure 63. Impulse Response of a Zero-Order Hold

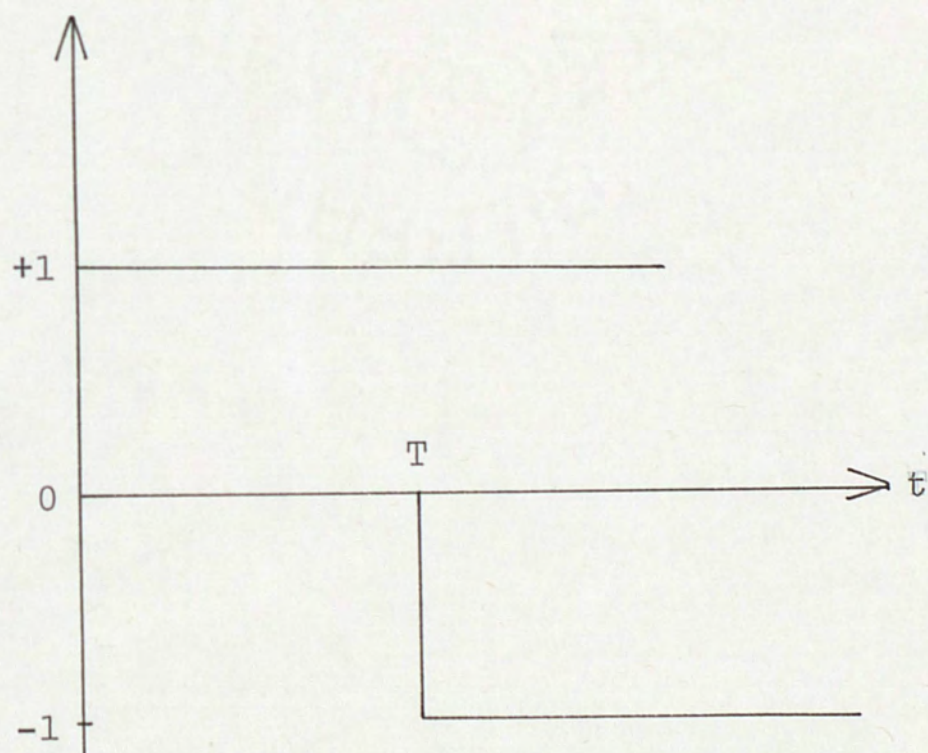


Figure 64. Decomposition of the Impulse Response of a Zero-Order Hold.

$$G_{h1}(S) = \left[\frac{1+TS}{T} \right] \left(\frac{1-e^{-TS}}{S} \right)^2 \quad (10)$$

Second-Order Hold

A second-order hold equation evolves if the first three terms of equation (1) are used. The resulting equation would be the following:

$$e_k(t) = e(kT) + [e(kT) - e(k-1)T](t-kT) + \frac{1}{2T^2} \{e(kT) - 2e[(k-1)T] + e[(k-2)T]\}(t-kT)^2 \quad (11)$$

By applying the conditions for a unit-impulse input, $e(0) = 1$, $e(-T) = 0$ and $e(-2T) = 0$ the responses for time periods $0 \leq t < T$, $T \leq t < 2T$, and $2T \leq t < T$ respectively are:

$$1 + \frac{t}{T} + \frac{t^2}{2T^2}$$

$$\frac{t}{T} - \frac{t^2}{T^2}$$

$$\frac{1}{2T^2}(t-2T)^2$$

The impulse response for all other sample periods is zero. Therefore the impulse response is shown in Figure 66 and it can be shown that its transfer function is the following expression:

$$G_{h2}(S) = \frac{1+TS+T^2S^2}{T^2} \left(\frac{1-e^{-TS}}{S} \right)^3 + \frac{4}{T} e^{-TS} \left(\frac{1-e^{-TS}}{S} \right)^2 + 5e^{-2TS} \left(\frac{1-e^{-TS}}{S} \right) \quad (12)$$

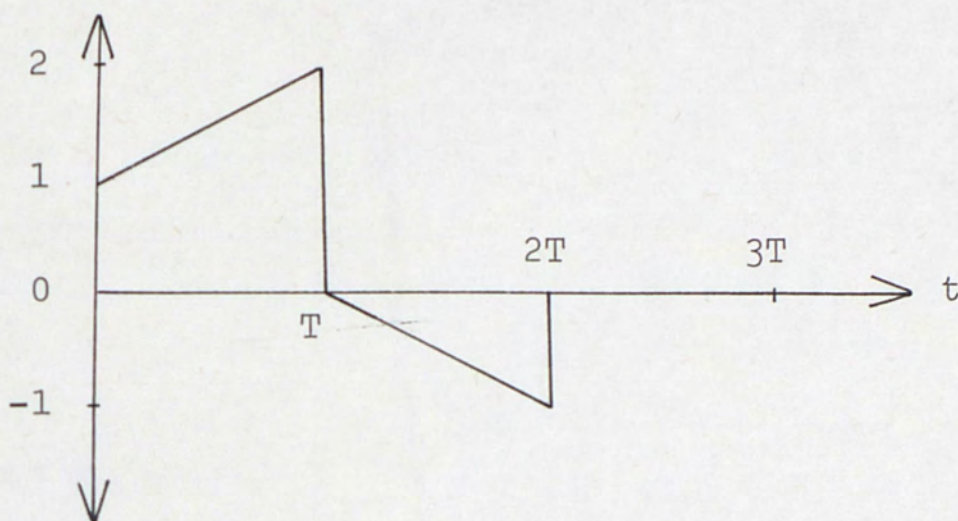


Figure 65. Impulse Response of a First-Order Hold

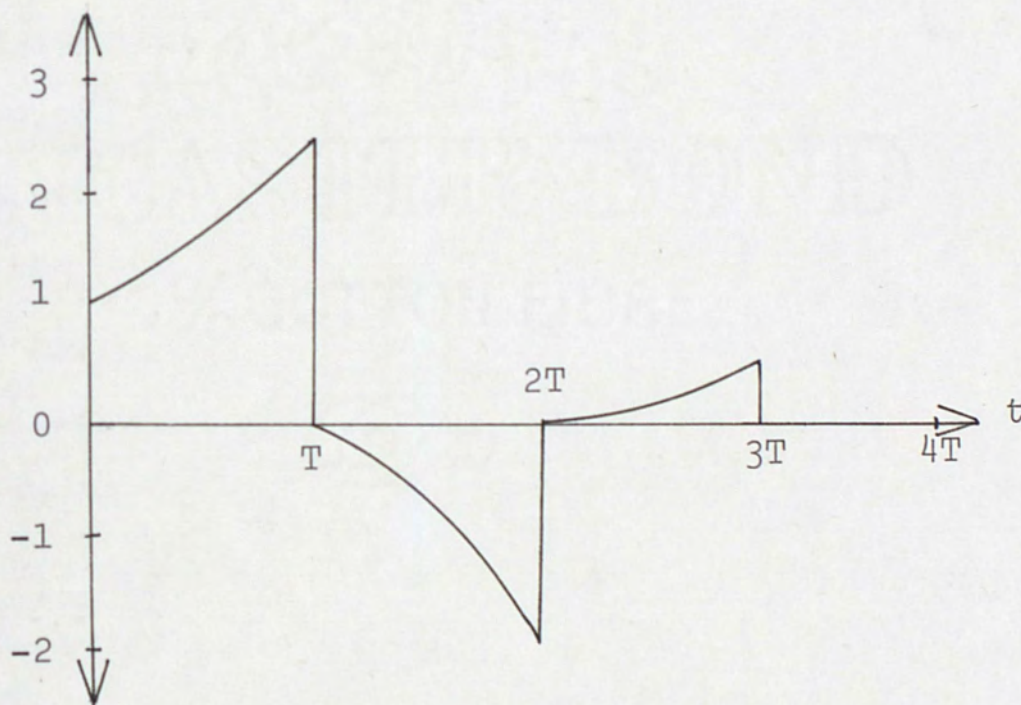


Figure 66. Impulse Response of a Second-Order Hold

APPENDIX C

STABILITY ANALYSIS

In the performance of any control system, the most important requirement is its stability. A system is said to be stable if it has a finite output response to any bounded input. This definition holds whether the system contains continuous-data, discrete(sampled)-data or a combination of the two kinds of signals.

In a closed loop sampled-data system, stability is a function of two factors--the open loop gain (K) and the sampled period (T). If the closed loop system's poles lie inside the unit circle in the z -plane, as shown in Figure 67, a sampled-data system is defined by Kuo [12] as being stable. Due to this definition one of the well-known stability analysis techniques for continuous-data system, Routh-Hurwitz criterion, cannot be applied directly to the characteristic equation. However once a bilinear transformation is performed on the characteristic equation, as shown in the example below, the interior of the unit circle in the z -plane can be mapped onto the left half of a complex variable plane. Then the Routh-Hurwitz stability criterion can be applied directly to the equation in the new variable.

From Figure 68 the following equations can be derived

$$C(S) = G(S)E^*(S) \quad (1)$$

$$E(S) = R(S) - C(S) \quad (2)$$

$$E(S) = R(S) - G(S)E^*(S) \quad (3)$$

Taking the pulse transform of (3) yields

$$E^*(S) = R^*(S) - G^*(S)E^*(S) \quad (4)$$

$$E^*(S) = \frac{R^*(S)}{1+G^*(S)} \quad (5)$$

$$C(S) = \frac{G(S)R^*(S)}{1+G^*(S)} \quad (6)$$

Pulse transform of $c(t)$ or the Laplace transform of $c^*(t)$ is obtained by taking the pulse transform of equation (6).

$$C^*(S) = \frac{G^*(S)R^*(S)}{1+G^*(S)} \quad (7)$$

The z-transform of $c(t)$ is obtained by simply changing variables of $S = \frac{1}{T} \ln z$.

$$C(Z) = \frac{G(Z)R(Z)}{1+G(Z)} \quad (8)$$

then

$$\frac{C(Z)}{R(Z)} = \frac{G(Z)}{1+G(Z)} \quad (9)$$

where

$$G(Z) = z \left[\frac{K}{S(S+a)} \right] = \frac{K}{a} \left[\frac{(1-e^{-aT})Z}{(Z-1)(Z-e^{-aT})} \right] \quad (10)$$

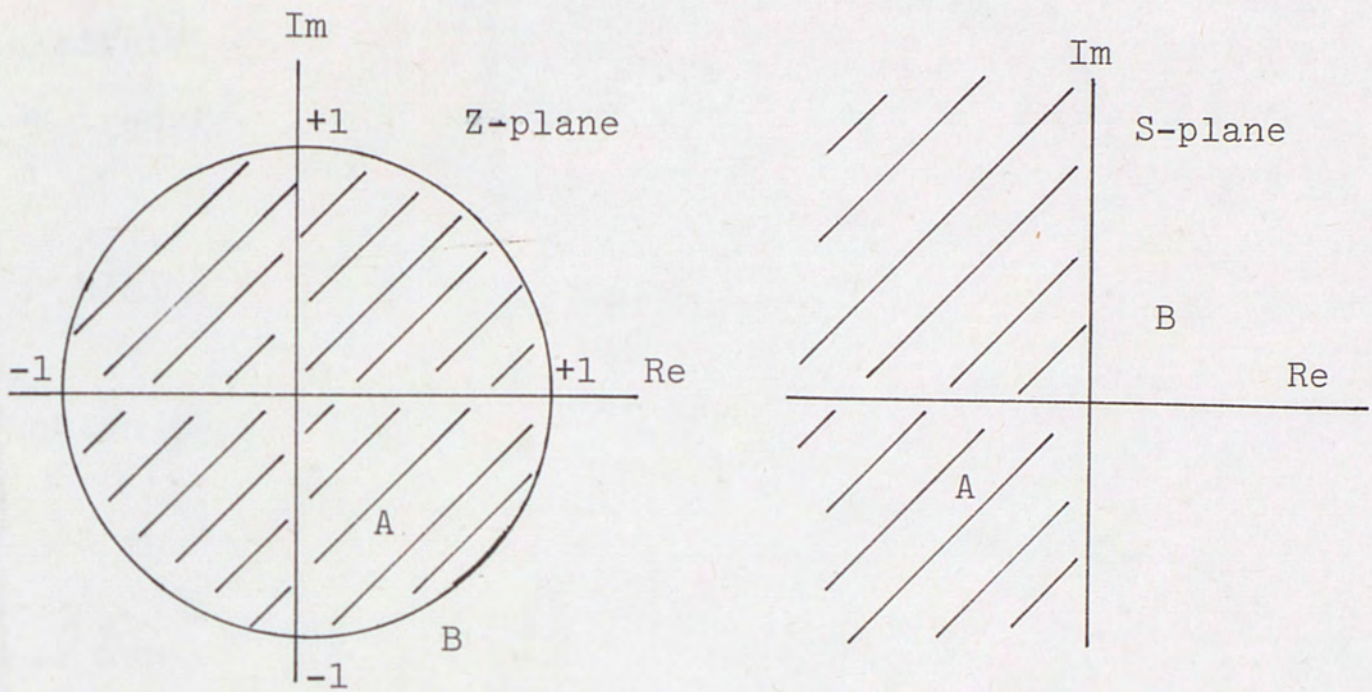


Figure 67. Corresponding Regions in Z and S Planes

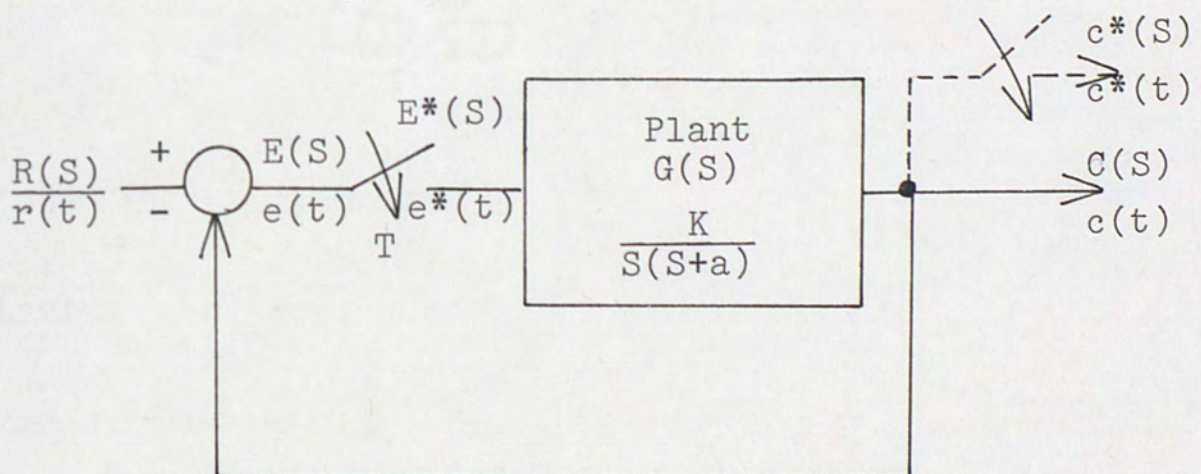


Figure 68. Second-Order Sampled-Data System With No Hold

Substitute equation (10) into equation (9) and the closed loop transfer function is then

$$\frac{C(Z)}{R(Z)} = \frac{\frac{K}{a}(1-e^{-aT})Z}{Z^2+Z\left[\frac{K}{a}(1-e^{-aT})-(1+e^{-aT})\right]+e^{-aT}} \quad (11)$$

Equation (11) has a characteristic equation of the form

$$Z^2+pZ+q = 0 \quad (12)$$

After applying the W-transform [12],

$$Z = \frac{1+W}{1-W} \quad (13)$$

The characteristic equation becomes

$$W^2(1-p+q)+W(2-2q)+(1+p+q) = 0 \quad (14)$$

The Routh criterion requirement for stability is given in D'Azzo's text [13] as

$$(1-p+q) > 0 \quad (15)$$

$$(1-q) > 0 \quad (16)$$

$$(1+p+q) > 0 \quad (17)$$

Using the values from equation (11) for p and q yields

$$2 - \frac{K}{a}(1-e^{-aT}) + 2e^{-aT} > 0 \quad (18)$$

$$1 - e^{-aT} > 0 \quad (19)$$

$$\frac{K}{a}(1-e^{-aT}) > 0 \quad (20)$$

Figure 69 shows the results of the above requirements as a sample period (T) versus gain (K) plot for a second order sampled-data control system with a plant as shown in Figure 68. For the model employed in the study, with a damping ratio (δ) equal to 0.5 and ω_n equal to one

$$K = \omega_n^2 = 1$$

$$a = 2\delta\omega_n = 1.0$$

T_{\max} was determined to be 4.32 seconds.

All of the above analysis is based on a sampled-data system with no hold device in the closed loop configuration. If a hold device was used in the system, the open loop equation would change to the following

$$C(S) = G_h(S)G(S)E^*(S) \quad (21)$$

and all other equations would change accordingly until the final closed loop equation results into a function of $GG_h(Z)$.

Using the above procedure as outlined, the stability region for the same system when a zero-order hold was used was determined and is shown in Figure 70. Both the zero-order hold and the no hold stability regions were verified on the EAI 580 experimentally and the accuracy was within 5 per cent of each other.

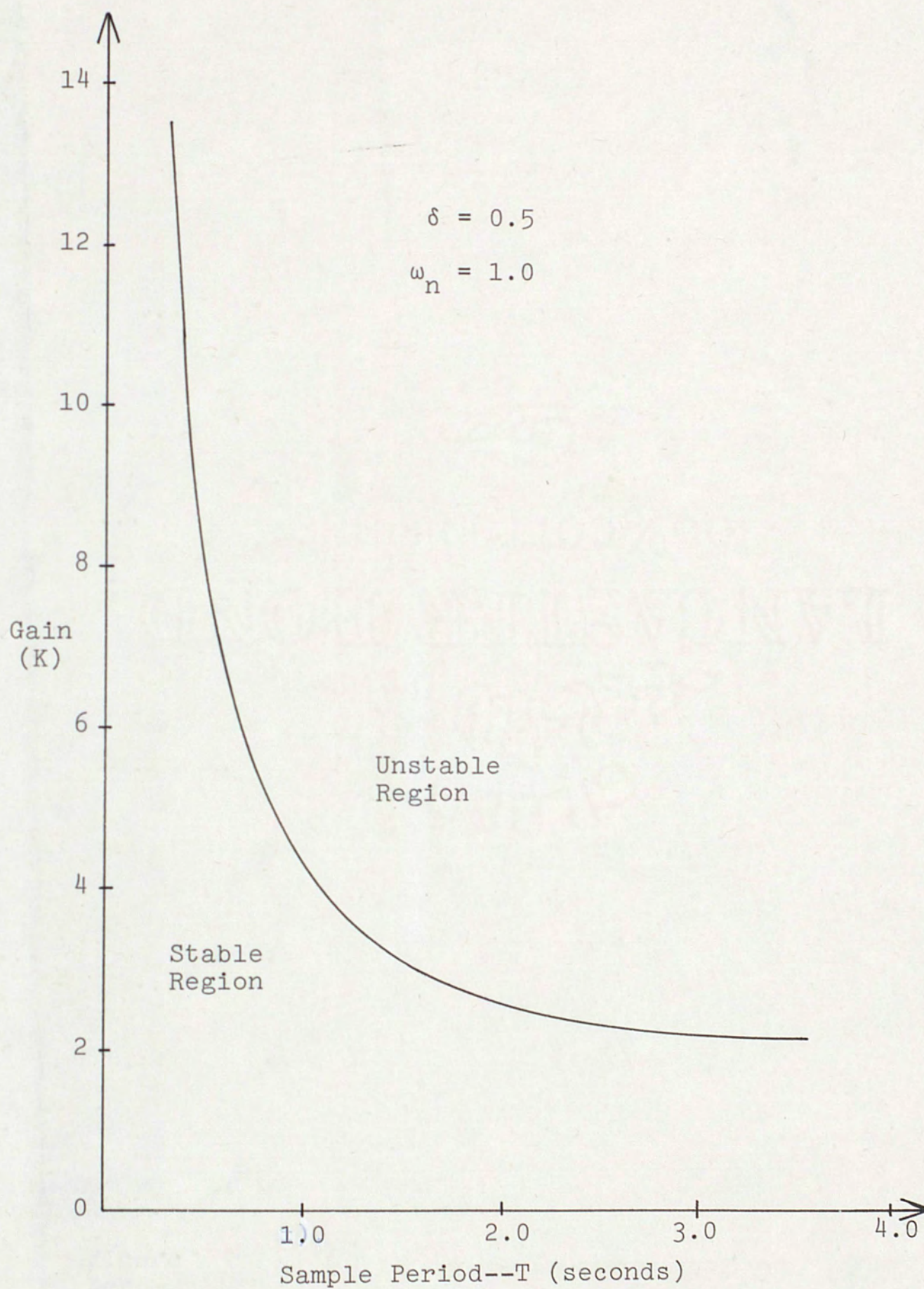


Figure 69. Stability Region for Second Order System With No Hold.

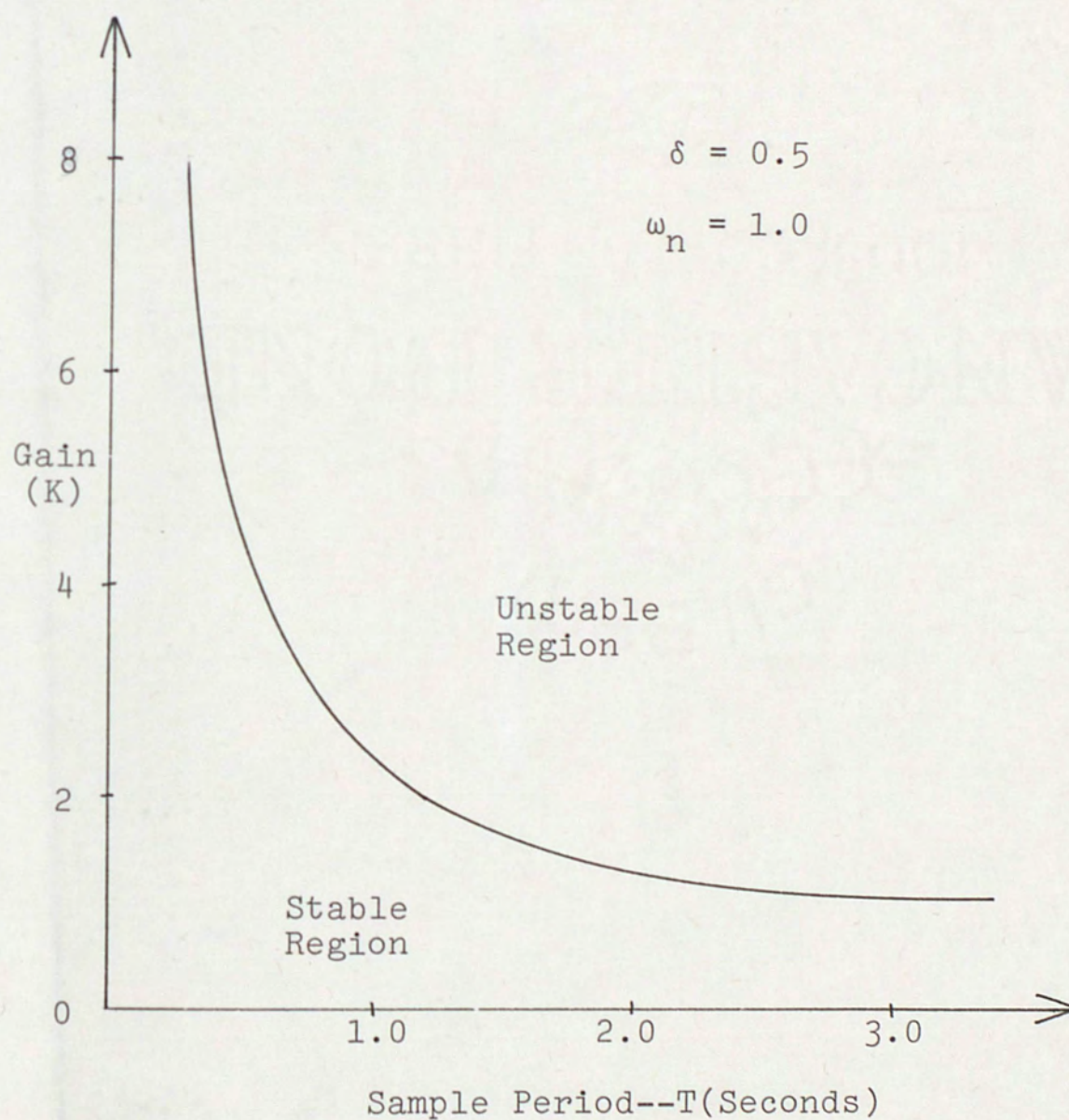


Figure 70. Stability Region for Second Order System With Zero-Order Hold.

One important factor must be kept in mind and that is as the order of the hold device increases so does the order of the characteristic equation. As can be seen in Appendix B from the transfer function of a second order hold device, the complexity of the stability requirements will also increase drastically.

REFERENCES

REFERENCES

- [1] Dorf, R. C., Farren, M. C., and Phillips, C. A. "Adaptive Sampling Frequency for Sampled-Data Control Systems." IRE Transaction on Automatic Control, Vol. AC-7 (January, 1962), 38-47.
- [2] Gupta, S. C. "Adaptive Gain and Adaptive Sampling-Data Control Systems." Paper presented at the IEEE Winter General Meeting, IEEE Conference Paper CP 63-130, January, 1963.
- [3] Tomovic, R., and Bekey, G. A. "Adaptive Sampling Based on Amplitude Sensitivity." IEEE Transaction on Automatic Control, Vol. AC-11 (April, 1966), 282-284.
- [4] Bekey, G. A., and Tomovic, R. "Sensitivity of Discrete Systems to Variations of Sampling Interval." IEEE Transaction on Automatic Control, Vol. AC-11 (April, 1966), 284-287.
- [5] Mitchell, J. R., and McDaniel, W. L., Jr. "Adaptive Sampling Technique." IEEE Transaction on Automatic Control, Vol. AC-14 (April, 1969), 200-201.
- [6] Electronic Associates, Incorporated, Technical Staff. 580 Desk-Top Analog/Hybrid Computing System, Bulletin No. 67312-1, 1967, p. 4.
- [7] Ogata, K. State Space Analysis of Control System. Englewood Cliffs, N. J.: Prentice-Hall, 1967, p. 332.
- [8] Kuo, B. C. Analysis and Synthesis of Sampled-Data Control System. Englewood Cliffs, N. J.: Prentice-Hall, 1963, p. 81.
- [9] Graeme, J. G., Tomey, G. E., and Huelsman, L. P. Operation Amplifiers, Design and Application. New York: McGraw-Hill, 1971, p. 217.

- [10] Durling, A. E., and Bullock, T. E. "A Unified Method for the Reconstruction of Sampled-Data." Submitted for publication to IEEE Transaction on Computers, Spring, 1972.
- [11] Ragazzini, J. R., and Zadeh, L. H. "The Analysis of Sampled-Data Systems." Transactions A.I.E.E., 71 Part II (1952), 225-234.
- [12] Kuo, B. C. Discrete-Data Control Systems. Englewood Cliffs, N. J.: Prentice-Hall, 1970, p. 125.
- [13] D'Azzo, J. J., and Houpis, C. H. Feedback Control System Analysis and Synthesis. 2d ed. New Jersey: McGraw-Hill, 1966, p. 122.

7MS.0 18.0.2.2