8-1-1962

Analysis and Design of Digital Control Systems

K. S. Fu
Purdue University

Raymond M. Kline
Purdue University

TR-EE 62-11
NATIONAL SCIENCE FOUNDATION CONTRACT G-14609
PRF 2991

PURDUE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

Analysis and Design of Digital Control Systems

K. S. Fu, Principal Investigator

R. M. Kline

Control and Information Systems Laboratory

June, 1962

Lafayette, Indiana

SUPPORTED BY
NATIONAL SCIENCE FOUNDATION
WASHINGTON, D.C.

TR-EE62-11

National Science Foundation Contract G-14609

PRF 2991

ANALYSIS AND DESIGN OF DIGITAL CONTROL SYSTEMS

Supported by

National Science Foundation

Washington, D. C.

by

King-sun Fu, Principal Investigator

R. M. Kline

School of Electrical Engineering

Purdue University

Lafayette, Indiana

June, 1962

ACKNOWLEDGMENTS

The author would like to express gratitude to his Major Professor, Dr. King-sun Fu, for suggestions and encouragement. Appreciation is also due Dr. Julius T. Tou under whom the initial portion of the work was accomplished. Typing and drafting assistance from the Control and Information Systems Laboratory as well as the use of equipment and facilities of the Control and Information Systems Laboratory, the Speech Laboratory, and the RPC 4000 Digital Computer Group were appreciated. Financial support from the National Science Foundation under Grant G-14609, Contract No. 2991-61-7805 is gratefully acknowledged.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF SYMBOLS

$\{\ \}$	- n x 1 matrix, vector
$[\ ]$	- square matrix
$[\ ]$	- row matrix, 1 x n
A	- magnitude of step input
a	- pole in the plant $\frac{1}{s(s+a)}$
B	- square matrix accounting for the instantaneous transitions occurring in the state variables at the closing of the sampling switch
b_{ij}	- element in the ith row, jth column of the B matrix
$b_1, b_2, \ldots$	- output magnitude of the first, second, etc. positive or negative levels of the quantizer (for case of identical negative and positive levels)
b	- magnitude of the positive and negative saturated output levels in a three level quantizer
e(t)	- system error at time t
f	- maximum fractional steady state error, $f = \frac{\delta}{A}$
G(s)	- plant transfer function
$H(\lambda)$	- square matrix giving the transition between a state vector at one time and another λ seconds later
K	- plant gain
k	- number of sampling periods
m(t)	- input signal to plant
N(kT+)	- output of a nonlinear element, usually a quantizer, at t = kT+

LIST OF SYMBOLS (Continued)

n - sampling period number during which e(t) first equals zero

Q - operator that quantizes the amplitude of the function upon which it acts

$r(t)$ - system input function

$|r|$ - magnitude of r(t) when it represents a step function

$|r|_n$ - normalized value of $|r|$, i.e., $|r|_n = \frac{|r|}{Kb}$

s - complex variable in Laplace transforms

T - sampling period (seconds)

t - time

t_o - reference time

t_o+ - the instant following t_o

t_r - time for the output to reach 95% of its final value

$u(t)$ - unit step function

$\underline{v}$ - state vector including the state variables, $x_1, x_2, \ldots, x_n$ as elements

$x_1, x_2, \ldots, x_n$ - state variables

x_1 - state variable representing the plant output

$x_1(\infty)$ - steady state value of the output

y - a constant defined by: $y = \frac{Aa^2}{Kb}$

y_1 - state variable

LIST OF SYMBOLS (Continued)

$\delta_1, \delta_2, \ldots$	-	beginning of the first, second, etc. quantization intervals in either the positive or negative direction (for case of identical positive and negative intervals)
δ_1	-	dead zone amplitude
δ	-	dead zone amplitude three level quantizer only
δ_n	-	normalized value of δ, i.e., $\delta_n = \frac{\delta}{Kb}$
λ	-	time measured from reference t_o, i.e., $\lambda = t - t_o$
ϕ	-	state transition matrix
ϕ_{ij}	-	element in the ith row, jth column of the ϕ matrix

ABSTRACT

Kline, Raymond M., Ph.D., Purdue University, August 1962. Analysis and Design of Digital Control Systems. Major Professor: King-sun Fu.

All digital control systems contain at least one signal which is sampled in time and quantized in amplitude. Design of these systems is often based on the assumption that a very large number of levels of quantization is available leading to the approximation of the basically nonlinear system by a linear one. If the actual system is constructed so that the linear assumption is satisfied, the performance may be excellent but other design factors such as reliability, cost, weight, and power consumption may be very unsatisfactory. On the other hand, if the actual system is constructed so that only a few levels of quantization are available, the other factors may be quite satisfactory but a previously well-behaved system may now possess limit cycle oscillations, large static errors, and an objectionable transient response. Thus, an important problem in the field of digital control is the development of analysis and design procedures such that the designer has the freedom to select quantization schemes not satisfying the linear approximation but producing an overall satisfactory design.

Two techniques are presented as a partial solution to the above problem. The first is based on a study of certain properties unique to quantized sampled-data system and uses Laplace transforms to carry out the analysis. It leads to closed form solutions but appears to be somewhat restricted in the class of systems to which it can be applied. The

second technique is a numerical procedure based on the state transition method and uses a digital computer to carry out the numerical calculations. It is not limited by input type, order of the plant, state variables having other than zero initial conditions, or quantizer complexity.

Using the first method, some closed form solutions are obtained for first and second order systems and the results favorably compared with the results obtained by the more general second method. Where possible the results of the second method are compared with the results of other workers. In other cases, typical results are checked by comparison with results from simulation on an analog computer. In all cases favorable comparisons are obtained. Design charts prepared by the numerical procedure are presented and examples given demonstrating their use to satisfy specifications on static accuracy, response time, and presence or absence of either overshoot or limit cycle oscillations.

A set of rules are derived describing certain properties of the system; e.g. a final value rule similar to the final value theorem for linear systems is obtained. These rules are found to be useful in both analysis and design by reducing the number of computations required to solve a given problem, by providing physical insight into system operation, and by furnishing a check on certain results.

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

The general representation for a single loop of a digital control system is shown in Fig. 1.1. In practice the complete system may consist of many loops sharing a large general purpose computer, or the complete system may consist of a single loop in which the computer has degenerated to merely a summing device. Concerning physical separation of the digital computer from the remainder of the system, again a tremendous contrast exist. For example, the computer may be adjacent to the remainder of the system in an automatic factory, or it may be separated from the remainder of the system by vast distances as happens when the plant is a part of a space vehicle controlled by a computer on the earth.

Analysis and design of digital control systems are often based on the assumption that a very large number of levels of quantization (large number of bits per computer word) are available leading to the approximation of the basically nonlinear system by a linear one. If the actual system is constructed so that the linear assumption is satisfied, the performance may be excellent but other design factors such as reliability, cost, ease of servicing, weight, and power consumption may be very unsatisfactory. On the other hand, if the actual system is constructed so that only a few levels of quantization are available, the other factors may be quite satisfactory but a previously

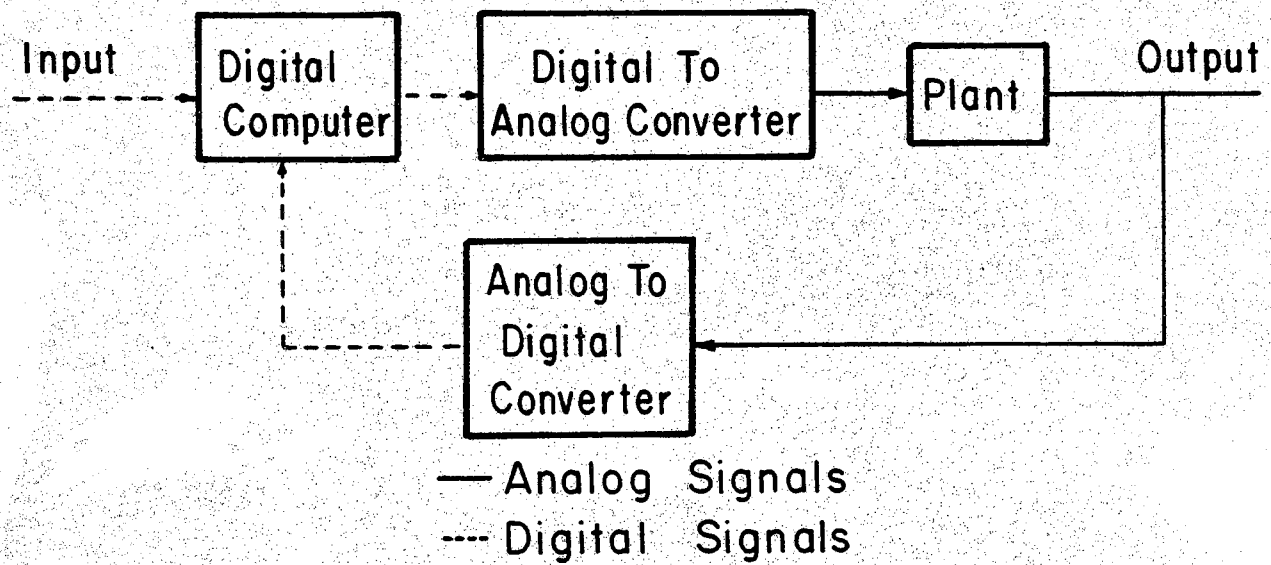


Fig. 1.1. General Representation of a Digital Control System

well-behaved system may now possess limit cycle oscillations, large static errors, and an objectionable transient response. Thus, an important problem in the field of digital control is the development of analysis and design procedures such that the designer has the freedom to select quantization schemes not satisfying the linear approximation but producing an overall satisfactory design.

The digital control system in which a relay is inserted in the error channel of a sampled-data system has been fairly extensively analyzed in the cases of second order plants and plants whose output is approximately sinusoidal. However, even here a complete design procedure does not appear to be available. For systems not fitting into one of these two cases, numerical methods appear to be the best means of analysis. However, none of the available numerical methods appear to be entirely satisfactory. The situation described above is compounded in multiple level quantized systems in that very little in the nature of analysis and design has been accomplished. These points are amplified and discussed more fully in the chapter on Literature Survey, Chapter 2.

1.2 Research Objectives and Procedures

One objective of this research is to develop a technique, which can be applied with a minimum of manual labor to the analysis of as wide a class of digital control systems as possible. The minimum permitted by this objective is the development of systematic analysis procedures not limited by the complexity of the quantizer, the order of the plant, or the input type. Another objective is the attainment of as much insight into the design of digital control systems as is possible.

These objectives are accomplished by using the following procedures:

1. Extension of the state transition method to the analysis of nonlinear systems (discussed in Chapter 3).
2. Development of the digital simulation technique, which involves implementing the state transition method on a digital computer (discussed in Chapter 4).
3. Derivation of a set of computational rules to be used in conjunction with 1 and 2 above to further reduce the work required, to provide physical insight into control system operation, and to be used as a checking method (discussed in Chapter 3).
4. Evolution of design procedures from the above analytical techniques (discussed in Chapter 5).
5. Derivation of closed form solutions, which can be used in the analysis and design of certain systems (discussed in Chapter 6).
6. Simulation of typical systems on an analog computer to provide a check on the methods of analysis and design and to provide physical insight into the operation of practical systems (discussed in Chapter 4 and Appendix B).

1.3 Terminology

Since a large class of digital control systems can either be directly reduced to the form shown in Fig. 1.2 or can be reduced to it after some minor simplifications, exclusive consideration will be given

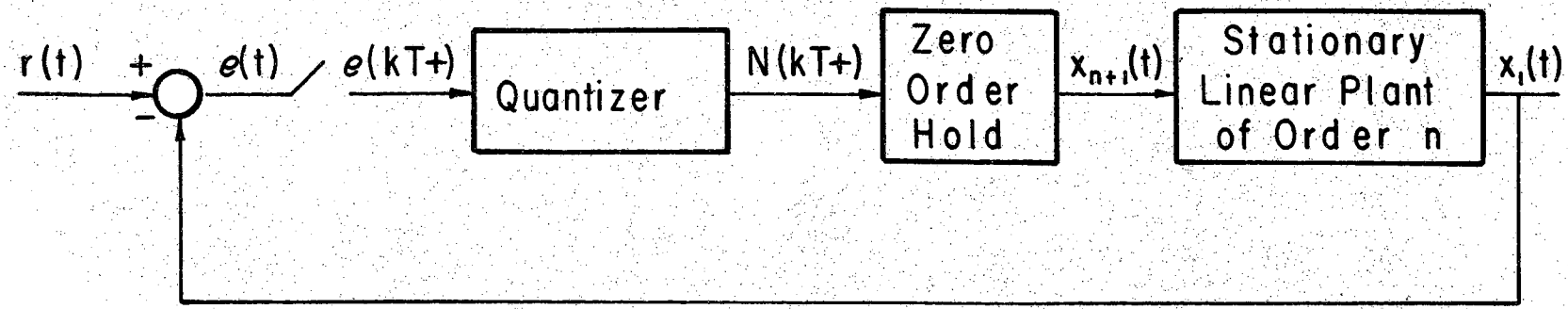


Fig. 1.2. A Typical Digital Control System

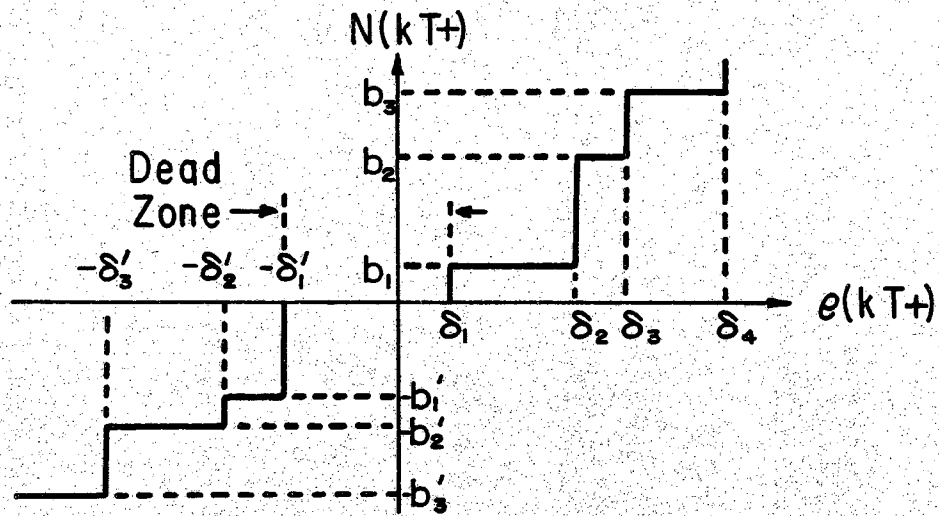


Fig. 1.3. General Description of the Quantizer

this form in presenting examples. However, it appears that most of the basic methods developed here can be extended to a wider class of systems. Throughout this report, systems are classified according to the order of the plant contained in the system; e.g. a second order system means a system of the form of Fig. 1.2 containing a second order plant.

The output of the quantizer in Fig. 1.2 is labeled N(kT+) while the input and output of the plant are labeled in accordance with the terminology of the state transition method, which is developed in Chapter 3. The remainder of the labeling on Fig. 1.2 is standard. Because the holding circuit is of zero order, it is immaterial whether this circuit appears before or after the quantizer. The quantizer itself is shown in more detail in Fig. 1.3 where δ_1, δ_2, etc. indicate the beginning of the first, second, etc. quantization intervals for positive values of quantizer input; δ_1', δ_2', etc. indicate the beginning of the first, second, etc. quantization intervals for negative values of quantizer input; b_1, b_2, etc. are the output amplitudes for the positive quantized levels; b_1', b_2', etc. the output amplitudes for the negative levels. Usually in practice $\delta_1 = \delta_1'$, $b_1 = b_1'$, $\delta_2 = \delta_2'$, $b_2 = b_2'$, etc. Although this simplification will be used henceforth, it could be eliminated in most of the work that is to follow. The interval from $-\delta_1'$ to δ_1 will be called the quantizer dead zone, and with $\delta_1 = \delta_1'$, the dead zone will be uniquely represented simply by giving its amplitude, δ_1. Quantizers are often classified by the total number of levels they contain, meaning the sum of the positive and negative levels plus one if a zero output level exist as it does in Fig. 1.3.

For three level quantizers, the subscripts on δ_1 and b_1 will be dropped and in this case the dead zone amplitude will be indicated by δ.

CHAPTER 2

LITERATURE SURVEY

2.1 General

Most portions of the field of linear sampled-data control systems are well covered by the three major texts[1,2,3] now available in English. However, as Jury[4] has stated in his review article, "In contrast to the linear theory of sampled-data, which has been thoroughly developed, the nonlinear theory has not been widely investigated and developed". Concerning the work that has been done, much of it must be classified as analysis rather than design, but even when analysis alone is considered, a fairly narrow section of the field has been studied. For purposes of classification and convenience in this discussion, the literature considered will be somewhat arbitrarily divided into the following six sections: 1) Classical Methods, 2) Numerical Methods, 3) Analog Simulation, 4) Methods of Approximation, 5) Special Techniques.

2.2 Classical Methods

The classical describing function method of analysis for continuous data systems has been extended to sampled-data systems by a number of investigators including Chow,[5] Russel,[6] and Kuo.[7,8] Chow presented a number of examples in his paper showing the predicted limit cycle amplitude and period compared to values actually obtained experimentally on the analog computer. He also predicted the dead zone amplitude required to eliminate limit cycles. Although this method is applicable to higher order systems, it says nothing about transient performance and

static errors in those case when a limit cycle does not exist. In Section 4.4 a comparison is presented between the results obtained by the method of this report and the results of Chow's method.

Phase-plane analysis of nonlinear sampled-data systems has also been developed from the corresponding classical technique for continuous systems. Some of the most important work using this method has been done by Kalman,[9] Izawa[10] and Scheidenhelm,[11] Izawa and Weaver,[12] Mullin and Jury,[13] and by Aseltine[14]. The method is applicable to all types of inputs, and transient performance as well as static error are obtainable. However, the method is very difficult to apply to systems higher than the second order. To this must be added the comment that the method is fairly time consuming in application, and for the majority of cases it does not appear to be easily adaptable to machine solution. Again results obtained by this method have been favorably compared with those of digital simulation; see Section 4.4.

2.3 Numerical Methods

Numerical methods have the characteristic in common that they are step by step calculations based on some type of recurrence relation in which the actual solutions are either carried out manually or by means of a digital computer. These methods, in general, have fewer limitations than most other methods on the type of system that can be analyzed; e.g., usually there is no limit on the order of the plant or the type of input which is permitted. On the other hand, numerical methods usually provide less physical insight into overall system behavior and they usually are more difficult to use in system design.

Tostanoski,[15] Kinnen and Tou[16,17] and Steel[18] have been some of the more prominent workers in this area. Tostanoski described how the analysis of sampled-data systems based on a z-transform approach could be carried out on the IBM Type 650 computer. He also mention the possibility of using a variation of this method for nonlinear systems. Unfortunately, no examples of the solution of either linear or nonlinear systems were given nor were any suggestions made as to how the method could be used for design.

Kinnen and Tou have used z-transforms to develop an exact method of analysis for nonlinear sampled-data systems in which the nonlinearity appears in the error channel but is not between frequency sensitive elements. This location effectively allows the nonlinear element to be separated from the linear part of the system so that a recurrence relation can be written. Moreover, Kinnen and Tou have extended their method to the approximate analysis of systems in which the nonlinearity occurs between two frequency sensitive elements through the introduction of a fictitious sampler and hold circuit preceding the nonlinearity. The method was originally developed for use with manual computation; however, this author has successfully programmed it on a digital computer for the case of a quantizer which is not between frequency sensitive elements. Although the method is a very useful one, it appears to be more time consuming for either manual or machine computation than the method of digital simulation to be presented in Chapter 3. In addition, their method does not appear to be readily adaptable to the use of non-zero initial conditions on the state variables and it does not appear to be as versatile as digital simulation.

The approach of Steel is in many way similar to that of the earlier work of Kinnen and Tou. Again the nonlinearity must not be between frequency sensitive elements and again this allows a recurrence relation to be written. The author has also successfully analyzed quantized sampled-data systems by this method. However, it was found to be less general in scope and more time consuming in application than the method of Kinnen and Tou.

2.4 Analog Simulation

Sampled-data systems have been analyzed by simulating them on the analog computer with auxiliary equipment such as a relay or electronic gate being used to perform sampling action. Similar to the numerical methods, analog simulation is applicable to a wide class of control systems but the disadvantages are also similar in that there is usually less physical insight into system behavior and there is difficulty in obtaining design information. The work of Chestnut, et al,[19] Klein,[20] Wadel,[21] and Scheidanhelm, et al,[22] have been described in the literature. Unfortunately, only Scheidanhelm has considered a quantized sampled-data system and his was an experimental model of a specific system rather than an analog computer representation which could be easily changed to simulate a wide class of system types.

As mentioned in Section 2.2 Chow has presented results of his analog simulation work for comparison with the results produced by the describing function. Although the analog simulation work of Chow appears to be of very high quality he does not give a detailed description of the equipment or the techniques used.

2.5 Methods of Approximation

Having the same basic objectives but using entirely different techniques both Bertram[23] and Tsypkin[24] have developed some useful approximate results for the performance of quantized sampled-data systems. Bertram used the state transition approach to develop a method for obtaining an upper bound on the error in the state variables caused by quantization for any number of quantizers in the system. He also showed that the introduction of quantization can not cause instability in a previously stable sampled-data system. Unfortunately, Bertram's results are quite conservative and may lead to quantizer designs which are too complicated and expensive. Moreover, his method provides no information about transient performance or about the possibility of the existence of limit cycle oscillations.

By working in terms of the impulse response of the linear portion of a multiple level quantized system, Tsypkin was able to show that the maximum upper bound on the error caused by a single quantizer is given by the sum of the absolute values of the impulse characteristic of the linear portion multiplied by the quantization interval. Tsypkin touches on the problem of limit cycle oscillation and shows that it will have a value no larger than the upper bound for system error caused by quantization. The same comments made with respect to Bertram's work concerning the conservative nature of the results and the lack of information on the transient response also apply to Tsypkin's work.

2.6 Special Techniques

The articles by Torng and Meserve,[25] Tou and Lewis,[26] and Widrow[27]

do not logically belong to any of the above groups, nor do these articles have much in common except that they all consider quantized sampled-data systems. However, they are placed together in this section as a matter of convenience.

Torng and Meserve use a difference equation approach to determine the various limit cycle modes in a relay type sampled-data system. Their method applies to systems of any order but it does not furnish information on transient response, static error, or other phases of the overall problem.

Widrow has taken a statistical approach to the study of quantized systems. He has been able to develop a quantization theorem, analogous to Shannon's sampling theorem, which determines the conditions required for recovery of certain statistical properties of the control signal. This method may be applied to systems with deterministic inputs but the results will be given in statistical terms. Moreover, the method does not provide information on limit cycle conditions or on system transient performance.

A dynamic programming approach is used by Tou and Lewis to develop a design technique for multiple level quantized systems. The designs produced by this method are optimum in the sense that certain performance criteria are minimized. Although the method appears to have a great amount of potential, it is presently limited to rather simple systems due to the complexity of the computational problem.

CHAPTER 3

ANALYSIS OF NONLINEAR SAMPLED-DATA SYSTEMS

3.1 General Considerations

Some other numerical methods[16,18] have been used successfully by the writer for the solution of nonlinear sampled-data systems, but the state transition method is especially recommended for its convenience, versatility, ability to provide information concerning the behavior of all state variables, and its apparent potential as a basis on which to build more elegant methods for analysis and design. This method has been used for synthesis[26,28] and linear analysis[29,30] of control systems and can be readily adapted to provide an exact method of analysis for nonlinear sampled-data systems if the nonlinear element can be placed adjacent to a zero order hold. (In most other cases it should be possible to at least develop approximate solutions, but these in general will be more complicated).

3.2 Review of the Analysis of Linear Time Invariant Systems by the State Transition Method

A linear time invariant control system can be described in terms of a single nth order differential equation with constant coefficients and with the input or driving function m(t).

$$\frac{d^n x_1}{dt^n} + a_n \frac{d^{n-1} x_1}{dt^{n-1}} + \cdots + a_3 \frac{d^2 x_1}{dt^2} + a_2 \frac{dx_1}{dt} + a_1 x_1 = m(t) \qquad (3.1a)$$

This equation may also be written in terms of n first order differen-

tial equations,

$$x_2 = \frac{dx_1}{dt} \quad , \quad x_3 = \frac{dx_2}{dt} \quad , \ldots \tag{3.1b}$$

$$\frac{dx_n}{dt} = m(t) - a_n x_n - \ldots - a_3 x_3 - a_2 x_2 - a_1 x_1$$

where $x_1, x_2, x_3, \ldots, x_n$ are called the state variables. The above system may be described in another but equivalent way by representing the system on a state diagram, which is similar to the flow diagram used for an analog computer. Integrators are at the heart of the state diagram and the output from the integrators may be identified with the state variables mentioned above. For example, Fig. 3.1 shows the state diagram for the differential equation

$$\frac{d^2x_1}{dt^2} + a\frac{dx_1}{dt} = m(t) = x_3(0)\, u(t) \tag{3.2}$$

which represents the plant $\frac{1}{s(s+a)}$ driven by a step input.

A vector consisting of all the state variables arranged in some convenient way is called a state vector. Since the plant is linear, one component of the state vector at some time t, $x_1(t)$, may be found by applying the superposition theorem to the individual contributions due to each of the initial conditions at time t_o in the state vector $\underline{v}(t_o)$. Using matrix notation, the relation between the state vectors at the two different instants t and t_o may be written

$$\underline{v}(t) = \phi(t - t_o)\, \underline{v}(t_o+) = \phi(\lambda)\, \underline{v}(t_o+) \tag{3.3}$$

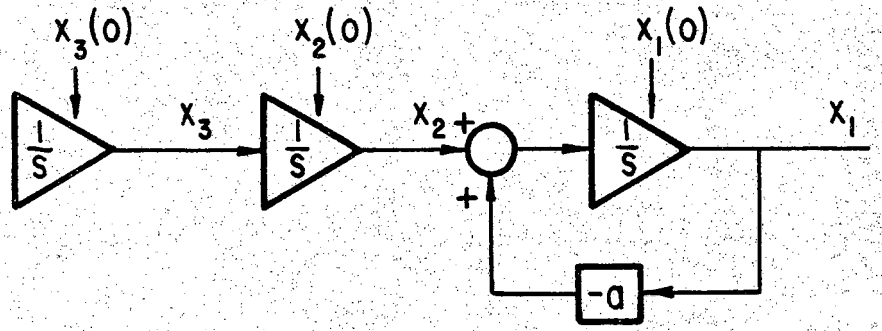


Fig. 3.1. State Diagram for the Plant $\frac{1}{s(s + a)}$
(Step Input)

where $\phi(\lambda)$ is the state transition matrix and $\lambda = t - t_o$. If the state vector is n x 1, the $\phi(\lambda)$ matrix is square and of order n. After expanding Eq. (3.3) the following equation is obtained

$$\begin{Bmatrix} x_1(t) \\ x_2(t) \\ \cdots \\ x_n(t) \end{Bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \\ \cdots & & & \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix} \begin{Bmatrix} x_1(t_o+) \\ x_2(t_o+) \\ \cdots \\ x_n(t_o+) \end{Bmatrix} \tag{3.4}$$

After matrix multiplication,

$$x_1(t) = \phi_{11}\, x_1(t_o+) + \phi_{12}\, x_2(t_o+) + \cdots \phi_{1n}\, x_n(t_o+) \tag{3.5}$$

Thus, Eq. (3.5) is exactly equivalent to the statement in words given above concerning the use of the superposition theorem to obtain $x_1(t)$. Moreover, $x_2(t)$, $x_3(t)$, etc. may be obtained in the same way. (A method for determining the ϕ matrix will be given later).

The notation t_o+ was used in the above equations in anticipation of the sampled-data case where a sampler, followed by a zero order hold, is closed at t_o and the relation between the state vector just before sampling, $\underline{v}(t_o)$, is related to the state vector just after sampling by the equation

$$\underline{v}(t_o+) = B\, \underline{v}(t_o) \tag{3.6}$$

where B is a matrix of the same order as ϕ. (A method for determining the B matrix will be given later).

Equations (3.3) and (3.6) may be combined to yield

$$\underline{v}(t) = \phi(\lambda)\, B\, \underline{v}(t_o) = H(\lambda)\, \underline{v}(t_o) \tag{3.7a}$$

where

$$H(\lambda) = \phi(\lambda)\, B \tag{3.7b}$$

The following sequence of equations can be written from Eq. (3.7a):

$$\begin{aligned}
&t_o = 0, \quad t = T \;; && \underline{v}(T) = H(T)\,\underline{v}(0) \\
&t_o = T, \quad t = 2T \;; && \underline{v}(2T) = H(T)\,\underline{v}(T) \\
&t_o = 2T, \quad t = 3T \;; && \underline{v}(3T) = H(T)\,\underline{v}(2T) \\
&\qquad\qquad \cdots \\
&t_o = \overline{k-1}\,T, \quad t = kT \;; && \underline{v}(kT) = H(T)\,\underline{v}(\overline{k-1}\,T)
\end{aligned} \tag{3.8}$$

where T is the sampling period and k is the number of sampling periods. Substituting the first equation of (3.8) into the second and then the second into the third, etc. finally the closed form expression for the state vector at the end of the kth sampling period is obtained in terms of the initial state vector $\underline{v}(0)$.

$$\underline{v}(kT) = \left[H(T)\right]^k \underline{v}(0) \tag{3.9}$$

If the initial state vector and state diagram are given, one can determine H(T) and then use Eq. (3.9) to determine the new state vector at the end of any sampling period.

A method of computing the entries in the $\phi(\lambda)$ matrix will now be established. Consider each of the initial conditions in Eq. (3.5) to be zero except one. For example, let $x_3(t_o+)$ be the only term with a nonzero value. From Eq. (3.5),

$$\phi_{13} = \frac{x_1(t)}{x_3(t_o+)} \tag{3.10}$$

Any convenient method may be used to determine $x_1(t)$, and since the system is linear, the initial condition $x_3(t_o+)$ is arbitrary and $u(t - t_o) = u(\lambda)$ may be used. As an example, consider the plant $\frac{1}{s(s + a)}$. The state diagram for this system is shown in Fig. 3.1. Since $x_3(t_o+) = u(t - t_o)$, its Laplace transform will be $X_3(s) = \frac{e^{-st_o}}{s}$. Moreover, the transfer function between $X_1(s)$ and $X_3(s)$ is $\frac{1}{s(s + a)}$; therefore,

$$X_1(s) = \frac{e^{-st_o}}{s} \left[\frac{1}{s(s + a)}\right] \tag{3.11}$$

and

$$x_1(\lambda) = \left[\frac{\lambda}{a} - \frac{1 - e^{-a\lambda}}{a^2}\right] u(\lambda) \tag{3.12}$$

Finally, substituting Eq. (3.12) and $x_3(t_o+) = u(\lambda)$ into Eq. (3.10),

$$\phi_{13} = \frac{\left[\frac{\lambda}{a} - \frac{1 - e^{-a\lambda}}{a^2}\right] u(\lambda)}{u(\lambda)} = \frac{\lambda}{a} - \frac{1 - e^{-a\lambda}}{a^2} \tag{3.13}$$

for $\lambda \geq 0$, which is the only case of interest here. The same method may be used to find the other entries in the $\phi(\lambda)$ matrix.

The entries in the B matrix are determined by first expanding Eq. (3.6) as follows:

$$\begin{Bmatrix} x_1(t_o+) \\ x_2(t_o+) \\ x_3(t_o+) \\ \cdots \\ x_n(t_o+) \end{Bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & \\ \cdots & & & \\ & & & \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix} \begin{Bmatrix} x_1(t_o) \\ x_2(t_o) \\ x_3(t_o) \\ \cdots \\ x_n(t_o) \end{Bmatrix} \quad (3.14)$$

After matrix multiplication,

$$x_1(t_o+) = b_{11}x_1(t_o) + b_{12}x_2(t_o) + \\ + b_{1n}x_n(t_o) \quad (3.15)$$

and the equations for $x_2(t_o+)$, $x_3(t_o+)$, etc. follow in the same way. Having the equations for $x_1(t_o+)$, $x_2(t_o+)$ etc. the values of the entries in the B matrix are determined by inspection of the state diagram. It should be noted that application of the B matrix in this way requires that a state variable be assigned to the output of each zero order hold in the state diagram.

3.3 Application of the State Transition Method to the Analysis of Nonlinear Systems

Consider the nonlinear sampled-data system shown in Fig. 3.2. There is no difficulty in writing the terms in the $\phi(\lambda)$ matrix which only involve the plant. However, any attempt to include the nonlinearity in either the ϕ or in the B matrix causes that matrix to become nonlinear and makes it difficult to handle analytically.

In order to see how this happens in a more specific case, consider the quantized system shown in Fig. 3.3. Inspection of the state diagram yields the following equations:

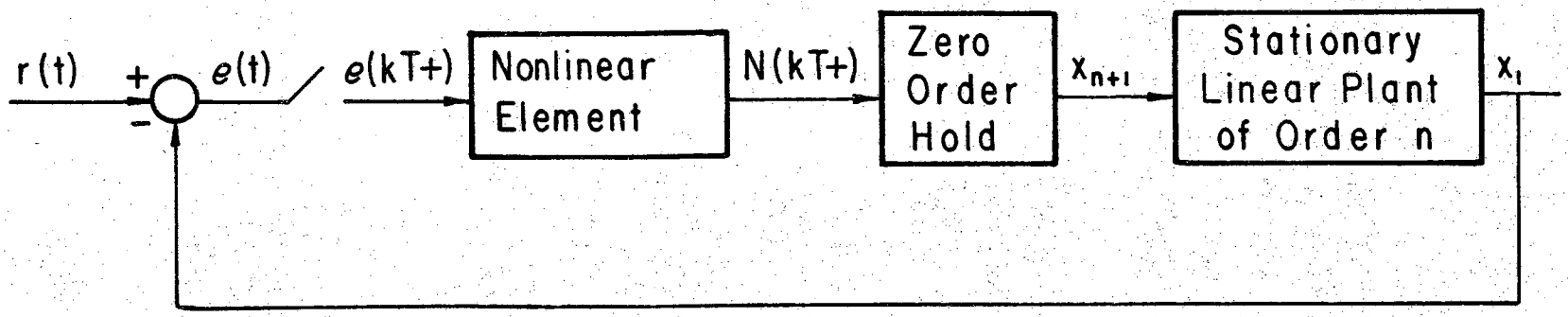


Fig. 3.2. A Typical Nonlinear Sampled-data Control System

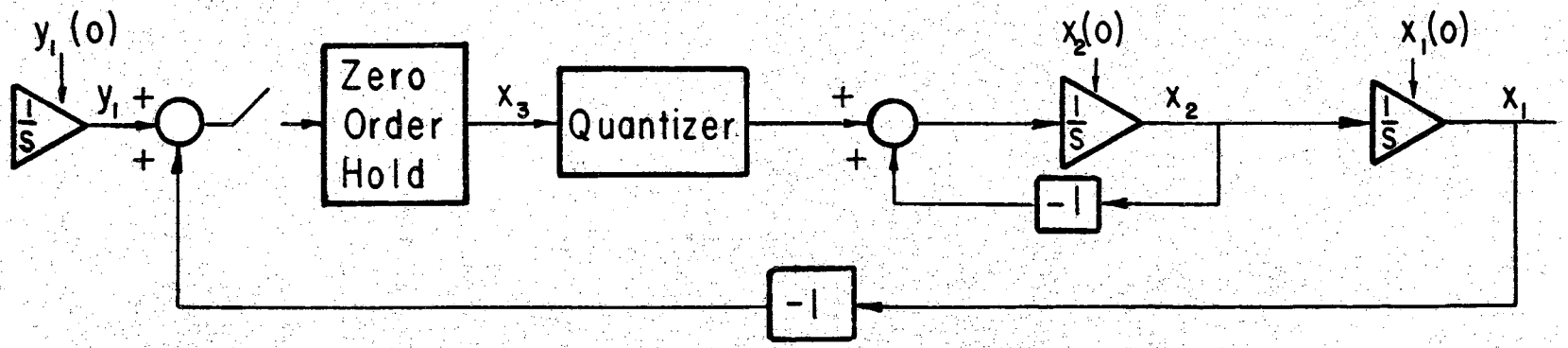


Fig. 3.3. State Diagram for a Quantized System

$$\begin{aligned} y_1(kT+) &= y_1(kT) \\ x_1(kT+) &= x_1(kT) \\ x_2(kT+) &= x_2(kT) \\ x_3(kT+) &= y_1(kT) - x_1(kT) \end{aligned} \tag{3.16}$$

where the sequence of the components in the state vector is the same as the sequence of the above equations. Therefore, the B matrix becomes:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \tag{3.17}$$

Using the method described above, the ϕ matrix is found to be:

$$\phi(\lambda) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 - e^{-\lambda} & (\lambda - 1 + e^{-\lambda})Q \\ 0 & 0 & e^{-\lambda} & (1 - e^{-\lambda})Q \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

where Q in Eq. (3.18) represents a quantization operator. (In this case, it quantizes all signals arriving at the point denoted by the state variable x_3). Thus, the $\phi(\lambda)$ matrix is nonlinear with respect to the state variables which makes it very difficult to use.

On the other hand, the fact that the holding circuit is of zero order allows the quantizer and hold to be interchanged producing Fig. 3.4. The $\phi(\lambda)$ matrix is now found to be the following linear matrix:

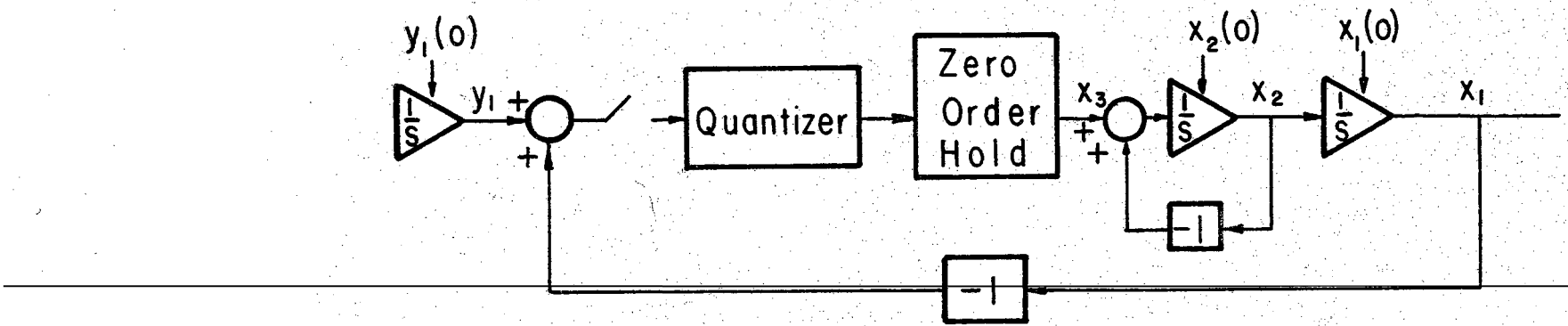


Fig. 3.4. State Diagram of Fig. 3.3 with Quantizer and Hold Interchanged

$$\phi(\lambda) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 - e^{-\lambda} & \lambda - 1 + e^{-\lambda} \\ 0 & 0 & e^{-\lambda} & 1 - e^{-\lambda} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

Inspection of the state diagram yields the following equations for the determination of the B matrix:

$$\begin{aligned} y_1(kT+) &= y_1(kT) \\ x_1(kT+) &= x_1(kT) \\ x_2(kT+) &= x_2(kT) \\ x_3(kT+) &= Q\left[y_1(kT) - x_1(kT)\right] \end{aligned} \tag{3.20}$$

However, now the quantization operator Q prevents one from writing the B matrix, because $x_3(kT+)$ can no longer be determined by a linear operation on $y_1(kT)$ and $x_1(kT)$.

On the other hand, as long as the nonlinearity is not between two frequency sensitive elements the following alternative procedure can be used. Referring to Fig. 3.2 again, the quantity e(kT+), the system error at the end of the kth sampling interval for a system with sampling period T, is computed from

$$e(kT+) = r(kT) - x_1(kT) \tag{3.21}$$

Then N(kT+), the output of the nonlinearity, is determined from e(kT+) and the characteristics of the nonlinearity. From Fig. 3.2,

$$x_{n+1}(kT+) = N(kT+) \tag{3.22}$$

and the state transition method, Eq. (3.3), can then be used to find the

state vector $\underline{v}(\overline{k+1}\ T)$ from the known value of $x_{n+1}(kT+)$ and the other state variables at kT+. The process can be repeated as often as desired to get the complete time response of the system. Of course, this method does not allow closed form solutions to be generated as was done in Eq. (3.9) for a linear sampled-data system where both the ϕ and the B matrices were used, but it does provide a convenient recurrence scheme. The method is also applicable to a large variety of other more complicated situations. For example, the system could include additional samplers and additional nonlinearities with the same general method as above being applicable. The only restriction is that if exact results are desired, the nonlinearity must not occur between frequency sensitive elements. This comment also holds for the nonlinearity between a hold circuit of order higher than zero and a frequency sensitive element, although the nonlinearity can precede a higher order hold circuit. Another complication which can be easily handled is the situation where it is desired to know the response between sampling instances. Here the input to the plant, x_{n+1}, is held at the value it had for kT+ and the parameter λ in the $\phi(\lambda)$ matrix is allowed to take on as many values as desired between 0 and T in order to generate the desired state vectors between $\underline{v}(kT+)$ and $v(\overline{k+1}\ T)$.

3.4 Illustrative Examples

In order to illustrate the above method, consider the following examples:

$$\text{Plant} = \frac{1}{s(s+1)}\ , \quad T = 1 \text{ second}, \qquad r(t) = .9u(t)$$

$$\underline{v}(0) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.23}$$

(Note that the plant gain, K, is unity in this example). The quantizer and the state diagram are shown in Fig. 3.5. Here the quantizer dead zone amplitude, δ, is 0.4 and the saturated output value, b, is 1.0. Find the state variables at the sampling instances.

From the state diagram the ϕ matrix is found to be:

$$\phi(\lambda) = \begin{bmatrix} 1 & 1 - e^{-\lambda} & \lambda - 1 + e^{-\lambda} \\ 0 & e^{-\lambda} & 1 - e^{-\lambda} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

Since only the sampling instances are of interest, $\lambda = T = 1$ second. Therefore,

$$\phi(T) = \begin{bmatrix} 1 & .632 & .368 \\ 0 & .368 & .632 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

Substituting the given values of $x_1(0)$ and $r(0)$ into Eq. (3.21), it is found that $e(0+) = .9$; the input to the quantizer has exceeded the dead zone amplitude and from Eq. (3.22) $x_3(0+) = N(0+) = 1$. Therefore,

$$\underline{v}(0+) = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \tag{3.26}$$

and

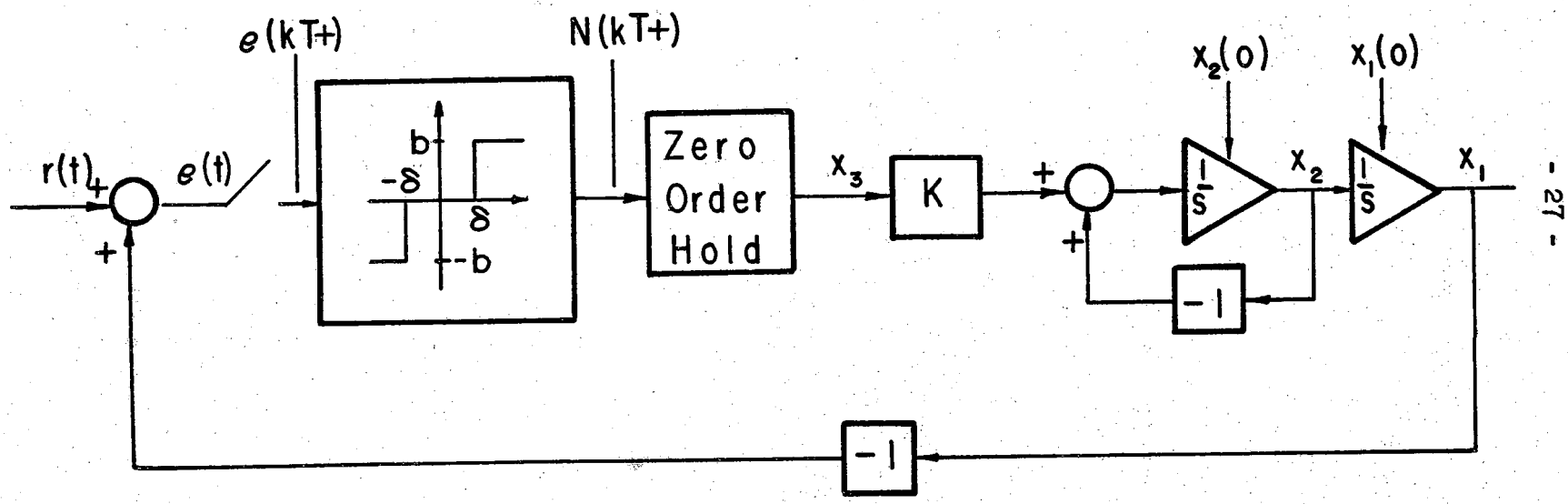


Fig. 3.5. State Diagram for a Second Order Digital Control System

$$\underline{v}(T) = \phi(T)\,\underline{v}(0+) = \begin{bmatrix} 1 & .632 & .368 \\ 0 & .368 & .632 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} .368 \\ .632 \\ 1 \end{Bmatrix} \tag{3.27}$$

Again applying Eqs. (3.21) and (3.22) it is found that $N(T+) = 1$; therefore, $\underline{v}(T+) = \underline{v}(T)$.

Matrix multiplication yields

$$\underline{v}(2T) = \begin{Bmatrix} 1.135 \\ .865 \\ 1 \end{Bmatrix} \tag{3.28}$$

In contrast to the previous two calculations, Eq. (3.21) and (3.22) now yield $N(2T+) = 0$; therefore,

$$\underline{v}(2T+) = \begin{Bmatrix} 1.135 \\ .865 \\ 0 \end{Bmatrix} \tag{3.29}$$

The above method may be repeated as often as desired. However, results for that first 14 seconds are summarized in Table 3.1.

An examination of the latter part of Table 3.1 reveals that $e(kT+)$ and the state variables have become periodic, which means that the system is in limit cycle oscillation. In this case, the period of oscillation is six seconds.

Example 2.

In this example consider the same system as in Example 1 except that the dead zone amplitude of the quantizer is 0.5 rather than 0.4. The computation proceeds using the same ϕ matrix as before,

Table 3.1

Summary of Results for Example 1

t (in sec.)	e(kT+)	x_1(kT+)	x_2(kT+)	x_3(kT+)
0	.900	0.000	0.000	1.000
1	.532	.368	.632	1.000
2	-.235	1.135	.865	0.000
3	-.782	1.682	.318	-1.000
4	-.615	1.515	-.515	-1.000
5	.078	.822	-.822	0.000
6	.598	.302	-.302	1.000
7	.421	.479	.521	1.000
8	-.276	1.176	.824	0.000
9	-.797	1.697	.303	-1.000
10	-.621	1.521	-.521	-1.000
11	.076	.824	-.824	0.000
12	.597	.303	-.303	1.000
13	.421	.479	.521	1.000
14	-.276	1.176	.824	0.000

and the results are shown in Table 3.2.

Note that the system is <u>not</u> in limit cycle oscillation, but the error appears to be approaching a steady state value of -0.1. Thus, the increase in the dead zone amplitude from 0.4 to 0.5 has been able to eliminate the limit cycle oscillations.

3.5 <u>Some Computational Rules</u>

Some computational rules have been developed which will provide insight into the operation of digital systems. These rules will help to simplify the computations in the analysis and design sections to follow.

Table 3.2

Summary of Results for Example 2

t (in sec.)	e(kT+)	x_1(kT+)	x_2(kT+)	x_3(kT+)
0	.900	0.000	0.000	1.000
1	.532	0.368	0.632	1.000
2	-.235	1.135	.865	0.000
3	-.782	1.682	.318	-1.000
4	-.615	1.515	-.515	-1.000
5	.078	.822	-.822	0.000
6	.598	.302	-.302	1.000
7	.421	.479	.521	0.000
8	.092	.808	.192	0.000
9	-.030	.930	.070	0.000
10	-.074	.974	.026	0.000
11	-.090	.990	.010	0.000
12	-.096	.996	.004	0.000

3.5.1 Dead-Zone Range Rule

Since the data in Tables 3.1 and 3.2 were for specific values of dead zone amplitude, it might at first appear that the calculations are good only for those specific values. This is not true in general as the discussion below shows.

Consider the data listed in Table 3.1 especially that of columns e(kT+) and x_3(kT+). The quantizer used to obtain these data has a dead zone amplitude of 0.4. Now consider the change in the results at t = 0 if the dead zone amplitude had been larger. It is apparent that the dead zone amplitude could be as large as 0.9 before the results would be changed. However, a dead zone amplitude of 0.9

would not leave the other points unchanged as consideration of the point at t = 1 second shows. There the dead zone amplitude could be increased to 0.532 before the results would be changed. Continuing it is found that the point at t = 2 seconds places no limit on the maximum value of the dead zone amplitude since $x_3(kT+)$ is already zero for this particular point. Thus the point t = 1 second is still the limiting point. Continuing to examine the points in Table 3.1, it is found that the point at t = 7 seconds is the ultimate limiting quantity and it establishes a limit on the maximum value which the dead zone amplitude may reach of $\delta_{max} = 0.421$ before the results of Table 3.1 are no longer valid. (There are other points, which occur every 6 seconds, having $e(kT+) = .421$ but none between .4 and .421). A similar inspection technique was used to determine the minimum value of dead zone amplitude applying to Table 3.1, and it was found to be $\delta_{min} = 0.276$, which first occurs at t = 8 seconds but again is repeated with a 6 second period. Applying the same technique to Table 3.2, it is found that $\delta_{max} = .532$, which occurs at t = 1 second, and $\delta_{min} = .421$, which occurs at t = 7 seconds.

An interesting point should now be noted. There is a common value for δ_{max} from Table 3.1 and δ_{min} from Table 3.2, i.e., both occur at $\delta = .421$; thus, this point should be the dividing line between limit cycle oscillation and completely stable behavior. If the specifications on the system, such as that shown in Fig. 3.5, where that the quantizer should not produce limit cycle oscillations for a single input of $r(t) = .9u(t)$, the design problem would be solved by using a quantizer with a dead zone amplitude of at least .421. Practical problems are

never this simple, but the use of the range rule has been demonstrated for a single point and further extension of the rule to more practical design problems will be presented later.

3.5.2 Input-Signal Range Rule

Just as the calculations in Tables 3.1 and 3.2 initially appear to apply only to a specific dead zone amplitude they also appear to apply only to a specific input magnitude, but again this is not true in general. Consider what will happen to $e(kT+)$ if the magnitude of $r(t)$ is increased. Since $e(kT+) = r(kT) - x_1(kT)$ is a linear equation, superposition applies; and if $r(kT)$ is changed by a given amount, $e(kT+)$ will be changed by the same amount. Proceeding as in the Dead Zone Range Rule then determine how much the magnitude of $r(kT)$ can increase before the results of Table 3.1 are no longer valid. By inspection it is found that $r(kT)$ can increase by .215, i.e., $r(kT)_{max} = 1.115u(t)$ and the critical point is at $t = 4$ seconds. Also $r(kT)_{min} = .879u(t)$, which first has its critical point at $t = 7$ seconds.

3.5.3 Final Value Rule

For a system without limit cycle oscillations it would be valuable to have a final value theorem or rule. However, the nonlinear nature of these systems prevent direct application of the conventional final value theorems. Consider Table 3.2; the values of the x_1 and x_2 state variables would be the same with an open loop system having the input shown in the x_3 column as it is with the closed loop system which was analyzed. Thus, if the designer has some insight as to what the $x_3(t)$ function is for a closed loop system or if he knows the functional form

from the results of analysis and wants an independent check on the system steady state value, the following method may be applied.

Since the plant itself is assumed to be linear, one can consider the output of the zero order hold to be a series of pulses which are considered separately and their individual results combined by superposition. The statement of final value rule then becomes: Given a linear plant, which is driven by a zero order hold whose output sequence is either known or assumed, the steady state value of any of the state variables associated with the plant is obtained by computing the steady state value caused by a single pulse of length T and then using superposition to determine the results for the actual pulse train.

The following is an example of the application of the final value rule. Given a plant or portion of a plant of transfer function G(s), and an input pulse of amplitude A and length T. The input function to the plant, m(t), may then be described by

$$m(t) = \left[u(t) - u(t - T)\right] \quad A \tag{3.30}$$

Taking the Laplace transform

$$M(s) = \left[\frac{1 - e^{-sT}}{s}\right] A \tag{3.31}$$

Using $x_1(t)$ as the output state variable of the plant G(s),

$$X_1(s) = M(s) \quad G(s) = \left[\frac{1 - e^{-sT}}{s}\right] \quad A\ G(s) \tag{3.32}$$

Therefore,

$$(x_1)_{ss} = (x_1)_{\text{steady state from a single pulse}}$$

$$= \lim_{s \to 0} \quad s\ X(s) = \lim_{s \to 0} \quad (1 - e^{-sT})A\ G(s) \tag{3.33}$$

Now the plant analyzed in Table 3.2 is of the form

$$G(s) = \frac{K}{s(s + a)} \tag{3.34}$$

thus

$$(x_1)_{ss} = \lim_{s \to 0} \frac{(1 - e^{-sT}) A K}{s(s + a)} \tag{3.35}$$

Direct substitution of s = 0 in Eq. (3.35) results in an indeterminant form, but by the application of L'Hospital's rule this is easily resolved as follows:

$$(x_1)_{ss} = \lim_{s \to 0} \frac{T e^{-sT} A K}{2s + a} = \frac{A K T}{a} \tag{3.36}$$

For the system analyzed in Table 3.2, T = 1 second, A = 1, and a = 1; thus, $(x_1)_{ss} = 1$. Referring to Table 3.2 it is seen that there are three positive pulses and two negative pulses each of magnitude A = 1. The result after superposition is a net steady state output of 1.0, which appears to check very well with the results of Table 3.2.

3.5.4 Quiescent Plant Rule

It is desirable to determine some conditions under which a plant can become or remain quiescent. These conditions as a group are called the Quiescent Plant Rule. The present version of the rule assumes a system having the general form of that shown on Fig. 3.2, with a quantizer of dead zone amplitude δ_1 as the nonlinear element. However, the same method can be applied to systems in other forms.

A necessary condition for the plant to become quiescent is ob-

tained as follows:

a) Unless $x_{n+1}(kT+)$ is zero for all k beyond some k_{min} the plant will receive actuating signals and cannot become quiescent.

b) By definition, $x_{n+1}(kT+)$ will be zero if

$$|e(kT+)| < \delta_1 \tag{3.37}$$

c) Now $e(kT+) = r(kT) - x_1(kT)$ (3.38)

d) After substituting Eq. (3.38) into Eq. (3.37), it is found that a necessary condition for the plant to become quiescent is that

$$|r(kT) - x_1(kT)| < \delta_1 \tag{3.39}$$

for all $k > k_{min}$.

It follows from the above reasoning that necessary and sufficient conditions for an initially quiescent plant to remain quiescent is that there be no external disturbances to the plant state variables and that Eq. (3.39) hold for all k. Moreover, Eq. (3.39) also indicates the size of the disturbance reaching x_1 or the size of the system input, r(t), required before the system will attempt to make a correction.

In many cases, time must theoretically approach infinity to have the plant state variables approach quiescence. However, for practical purposes the plant can be assumed to be quiescent after a length of time, beyond k_{min}, which is long compared to the longest time constant in the plant. In this way, the Quiescent Plant Rule can also be

applied by segments to situations where the system is actuated at widely separated intervals and approaches quiescence between these intervals.

CHAPTER 4

RESULTS FROM SIMULATION OF DIGITAL CONTROL SYSTEMS

4.1 General

Two basic method for simulating digital control systems were used to obtain the results presented in this chapter. In the first method, digital simulation, the technique explained in Chapter 3 was programmed on a small digital computer; and in the second method, analog simulation, the system was simulated in the usual sense by using an analog computer in conjunction with an experimental quantizer. The two methods of simulation were chosen to supplement one another and not to duplicate the other functions. The following are considered to be the advantages of the digital simulation approach:

1) Multiple and unusual quantizer are readily programmed on the digital computer as compared with the difficulty and expense of physically constructing them for use with the analog computer. (This is also true for other non-linearities).
2) Highly accurate, noise free performance is available.
3) Depending somewhat on the types of computers compared and the calculations required, the digital method will usually be faster.
4) The method of digital simulation, together with the computational rules presented in Chapter 3 can considerably reduce the computational time and provide greater physical insight especially in design.

5) Digital control systems involving logical decisions or complicated numerical operations can be simulated on a single machine.

These advantages for digital simulation may at first make it appear that analog simulation is unnecessary. However, the analog method was found to complement the digital method in the following ways:

1) Many of the practical problem such as drift and noise which occur in an actual system are encountered in analog simulation. These can be a "blessing in disguise" in that they give the investigator insight into the way these problems affect system design.

2) Since analog simulation is a completely independent technique, it provides an excellent method of checking digital simulation. In the work reported here, analog runs were made to "spot check" representative digital solutions for gross errors; but with the far superior accuracy of digital simulation, no attempt was made to obtain the ultimate in accuracy with the analog method.

4.2 Digital Simulation

For higher order systems and situations requiring calculations for a large number of sampling periods, hand calculations, such as those presented in Chapter 3, become very laborious. Furthermore, roundoff errors can become significant because each calculation depends on the previous values. Thus, programming of the numerical method on a digital

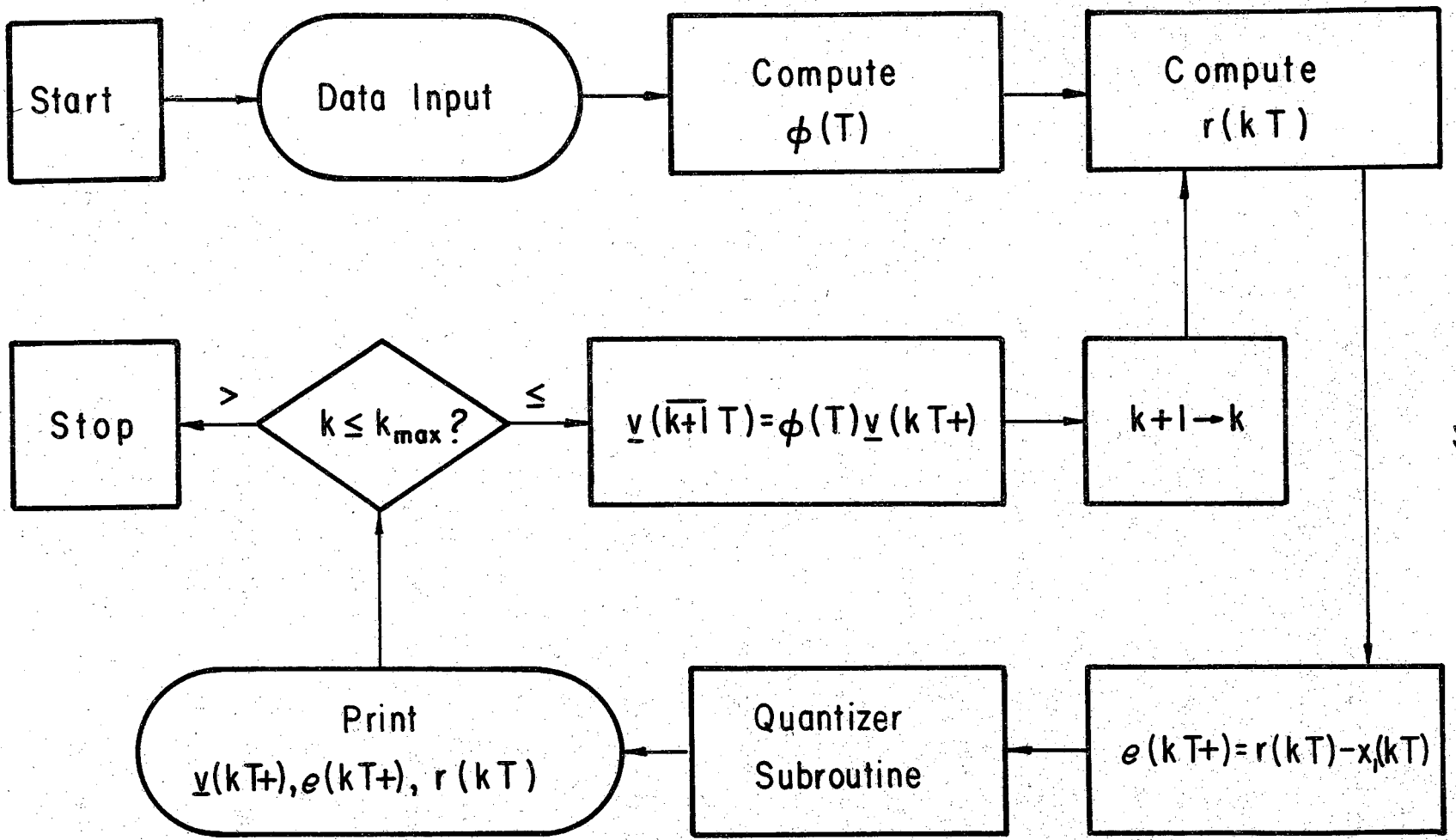


Fig. 4.1. Basic Computer Program Used for Digital Simulation

computer is advisable in many cases to satisfactorily simulate the control system. The computations reported here were performed on a Royal McBee RPC-4000 computer using the Purdue interpretive routine, PINT, although satisfactory results could be obtained on an even smaller machine. Consideration has been given to inclusion of the Dead Zone Range Rule and the Input Signal Range Rule into the computer program, and it might be necessary if a large machine were used. However, it was not done here because application of these rules is a task easily and accurately performed by the computer operator while waiting for the next series of calculations to be completed.

The basic flow diagram for the digital computer program is given in Fig. 4.1 with a complete PINT program being given in Appendix A. The diagram shown is for a single quantizer in the error channel, but more complicated systems should be capable of being analyzed by minor modifications of this basic program. The program shown in Appendix A automatically determines the $\phi(T)$ matrix for the plant $\frac{K}{s(s + a)}$ with any desired values of K, a, and T. This program is for a step input but a program to generate ramp inputs is also available. Moreover, it would not be difficult to write programs for a wide variety of other inputs, e.g. sine waves and random signals. In addition, complete freedom is allowed in the selection of initial conditions on the state variables. The quantizer subroutine permits quantizers of as many levels as desired and with any arrangement of individual levels to be simulated, also the quantizer subroutine can be bypassed so that unquantized systems can be simulated. By using the tape input feature

of the computer to supply new data, a large number of complete runs may be made automatically. For each computer run the system input, the error signal, and all of the state variables are printed, to eight significant figures, at the beginning of each sampling period. In this way a complete picture of the system performance is obtained.

It was easy to check the operation of the quantizer by comparing the value printed for the error signal at a particular time with the value printed for the output of the quantizer. Frequently it was possible to determine needed quantizer modifications for desired system performance merely by inspection of the computer's printed record alone or with the aid of one or more of the computational rules. In all cases where the system did not go into a limit cycle, the final value rule was applied to the quantizer output to verify the steady state value given on the printed record. It was interesting to watch a system begin to lock into a limit cycle in that it usually required several complete cycles before the state variables would repeat from one period to the next out to the full eight significant figures.

As an example of the application of the method of digital simulation to a fairly complicated system, the following third order system was considered:

$$\text{Plant} = \frac{s + 3}{s(s + 1)(s + 5)} \quad , \; r(t) = 1 + 0.8t, \quad T = 0.5 \text{ second}$$

$$\underline{v}(0) = \begin{Bmatrix} -.4 \\ .3 \\ .2 \\ 0 \end{Bmatrix} \tag{4.1}$$

and with the state diagram and quantizer shown in Figs. 4.2A and 4.2B

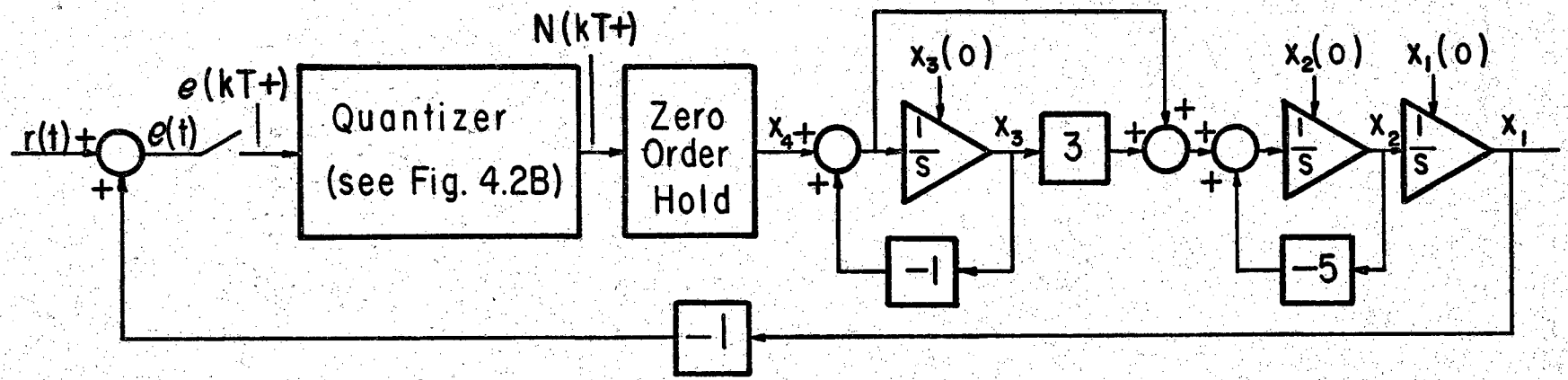


Fig. 4.2A. State Diagram of a Third Order Digital Control System

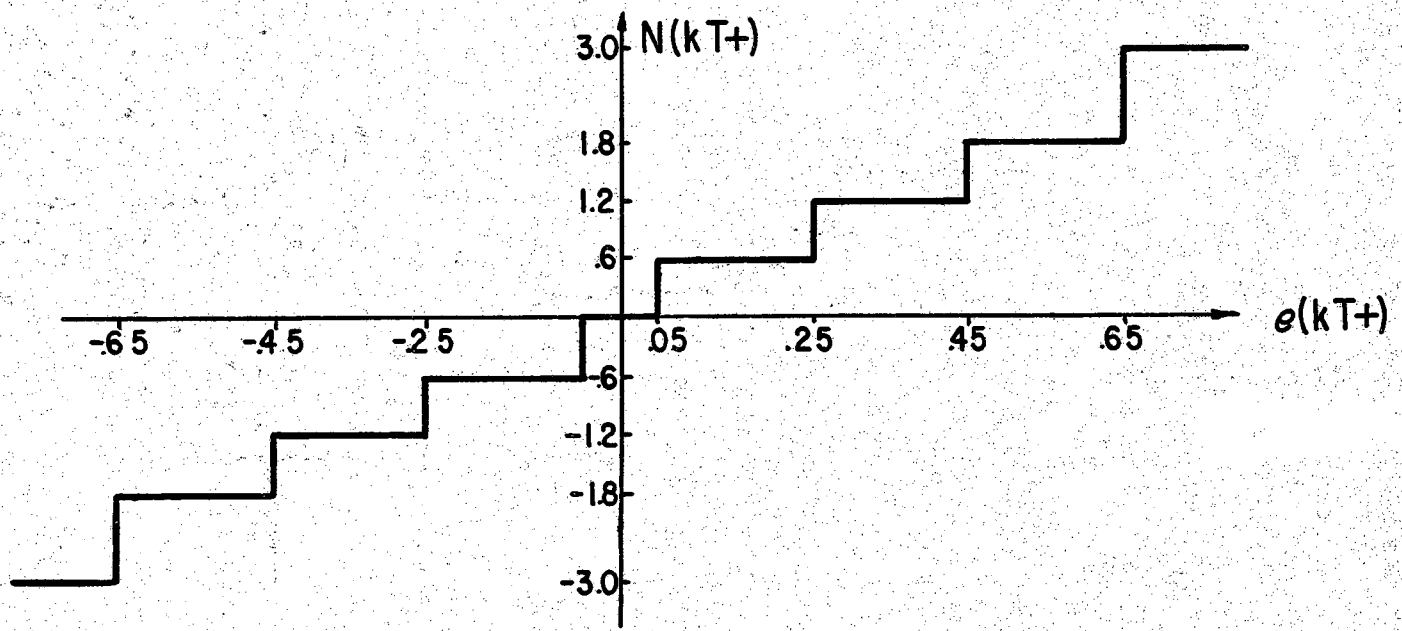


Fig. 4.2B. Quantizer Used in the System of Fig. 4.2A

respectively. Note that all state variables have nonzero initial conditions, that the input is a ramp superimposed on a unit step, and that the δ and b values of the quantizer are not all equal.

Using the method previously described the following transition matrix was found:

$$\phi(T) = \begin{bmatrix} 1 & .184 & .105 & .085 \\ 0 & .082 & .262 & .289 \\ 0 & 0 & .607 & .393 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.2)$$

These data were fed into the computer and the digital simulation carried out. The behavior of all state variables and the input for the first 15 sampling periods is shown on Fig. 4.3. Note that after the initial transient, the output follows the ramp input fairly well but there appears to be a steady state position error of approximately 0.4 unit.

4.3 Analog Simulation

A simplified block diagram of the equipment used in the analog simulation of the control systems considered here is shown in Fig. 4.4. The heart of the equipment is the quantizer, which consists of an Epsco analog to digital converter and a digital to analog converter developed by the author. With this equipment it is possible to simulate digital control systems with levels of quantization ranging all the way from very fine (10 bit quantization) to a relay without dead zone (1 bit quantization). In addition, it is possible to select a very wide range of sampling rates. (As far as the quantizer is concerned the limit is from approximately 0 to 30,000 cps).

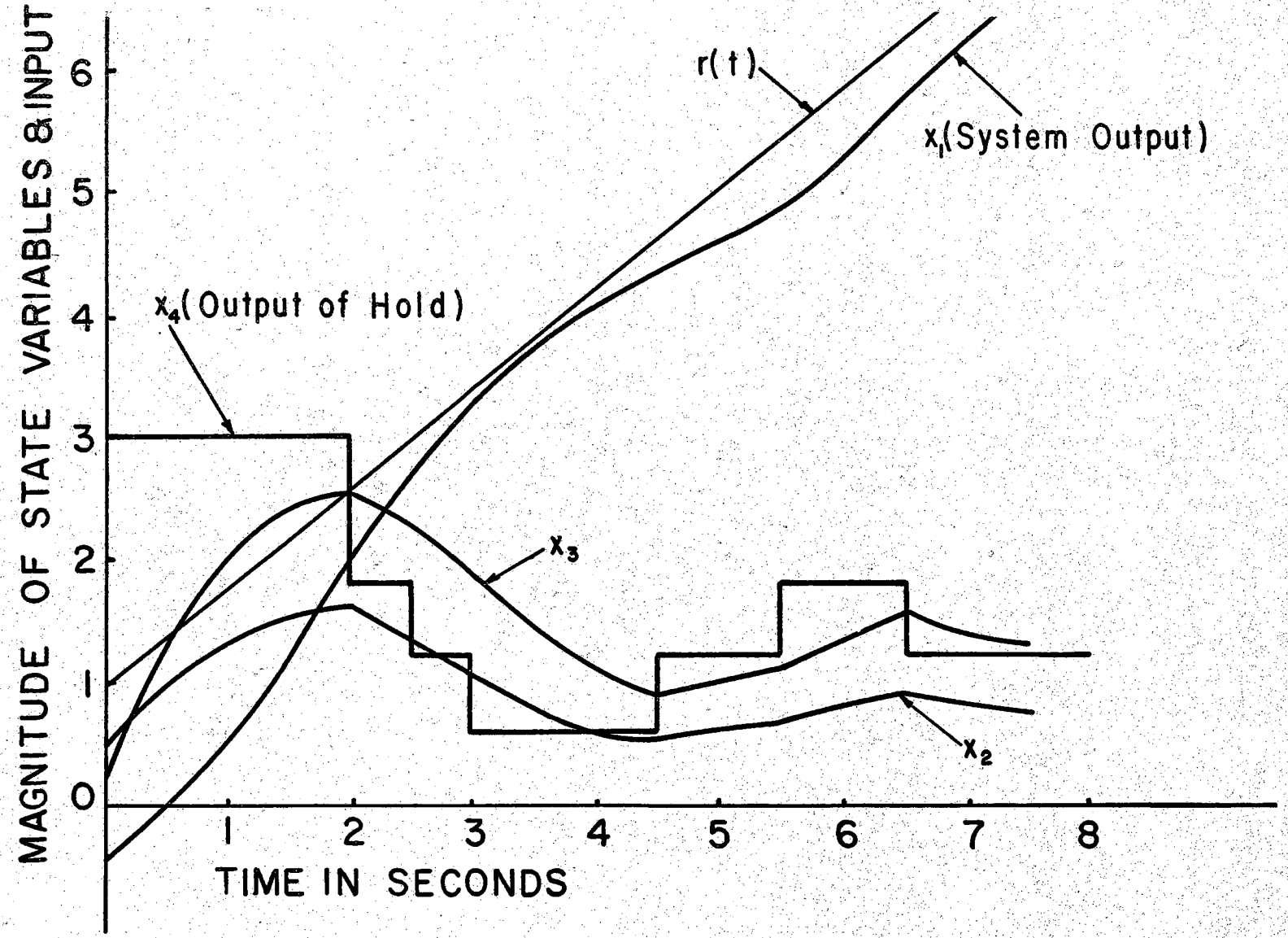
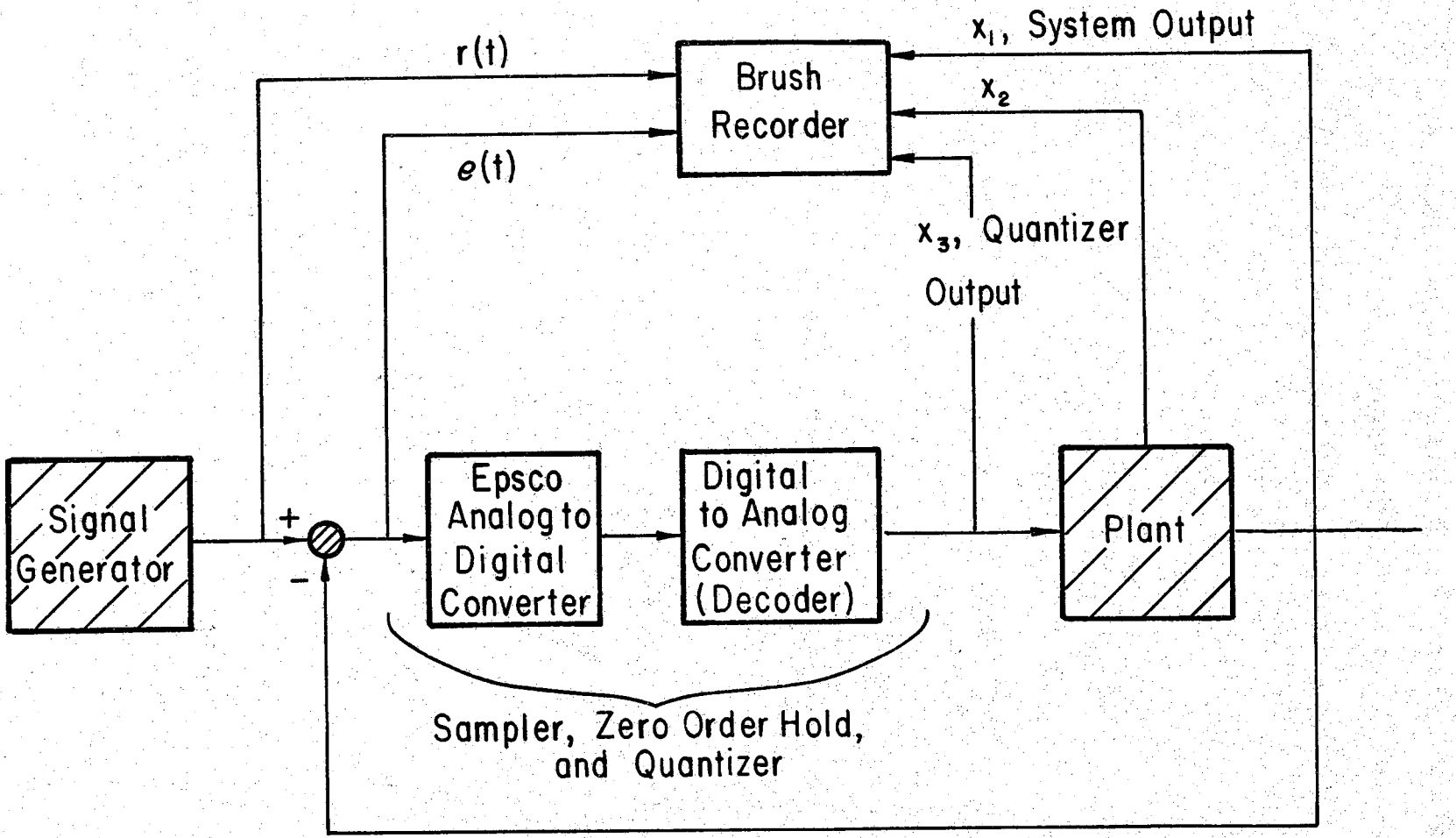


Fig. 4.3. Time Response of the System of Fig. 4.2A



Note: Shaded blocks indicate functions performed on an analog computer

Fig. 4.4. Simplified Block Diagram of Analog Simulation Equipment

As indicated in Fig. 4.4 plant simulation, signal generation, and certain other functions such as summing are performed on an analog computer. A six channel Brush recorder is used to record the same variables as are printed in digital simulation discussed in Section 4.2; thus, a direct comparison can be made between the results of the two methods of simulation and this is done in Section 4.5 below. A detailed discussion concerning the techniques used in the analog simulation work and complete circuit diagrams are presented in Appendix B.

4.4 Comparison with Results Presented in the Current Literature

In order to test the validity of the method of analysis developed here, a comparison will be made in this section between the results produced by digital simulation and those presented in the literature for the same system. A further validity test will be presented in Section 4.5 where a comparison will be made between results from digital simulation and analog simulation of the same system.

Consider a system having a sampling period of one second, a three level quantizer, and the plant $\frac{1}{s(s+1)}$. By changing the dead zone amplitude and varying the magnitude of the step input to the system, different modes of limit cycle oscillation can be produced. Both Kuo[8] and Chow[5] have analyzed such systems and the results for four modes of oscillation found by them are summarized in the first few columns of Table 4.1. Note that Kou's results are for the apparent amplitude of oscillation, which is computed by knowing the system output at the sampling points only, while those of Chow are for the

Table 4.1

COMPARISON OF ANALYSIS RESULTS FOR A SYSTEM IN LIMIT CYCLE OSCILLATION

Mode of Oscillation	Period (in Sec.)	Apparent Amplitude from Kuo[8]	True Amplitude from Chow[5]	Using Methods of this Report	
				Apparent Amplitude	True Amplitude
1	4	.30	.308	.295	.302
2	4	.352	.435	.3519	.430
3	5	.525	.571	.5253	.568
4	6	.80	.84	.799	.851

true* amplitude of oscillation. Using digital simulation, the system was analyzed and the apparent amplitude of oscillation computed for the same cases as presented by Kuo. (See the second last column in Table 4.1). Note that very good agreement is obtained with Kuo's results but that a significant differences appears between these results and those of Chow especially for the mode of oscillation labeled 2. The apparent reason for this is that the maximum and minimum of the output do not occur at the sampling instances. This appears to be physically reasonable because the acceleration can change instantaneously but the velocity cannot for this particular system; thus, the command to the plant may be to change the direction of motion but the output will continue in the original direction for an interval after the sampling instance. A crude approximation for determining the true amplitude is to plot the output against time for the sampling instances and draw a smooth curve by eye through the points. A better method is to calculate and plot several intersampling points and from these determine the amplitude of oscillation. It is not necessary to determine the intersampling points within all sampling periods. As a matter of fact, data for one sampling period in which the maximum occurs and one in which the minimum occurs are all that is necessary so that intersampling points are needed during only two periods. Although it would not be difficult to modify the computer program to accomplish this automatically,

---

*Actually Kuo has also calculated the true amplitude by another version of the same method used by Chow and obtained essentially the same results as Chow.

the following alternate manual method will also be satisfactory in many instances. Recall from Eq. (3.3) that the state vector at any time, t, is given by

$$\underline{v}(t) = \phi(t - t_o)\, \underline{v}(t_o) \tag{4.3}$$

To obtain intersampling points, let $t = t_o + pT$ in Eq. (4.3)

$$\underline{v}(t_o + pT) = \phi(t_o + pT - t_o)\, \underline{v}(t_o) = \phi(pT)\, \underline{v}(t_o) \tag{4.4}$$

where $0 \leq p \leq 1$.

Since only the output is of interest, Eq. (4.4) may be simplified as follows

$$x_1(t_o + pT) = \left\lfloor \phi_1(pT) \right\rfloor \underline{v}(t_o) \tag{4.5}$$

where $\left\lfloor \phi_1(pT) \right\rfloor$ is the row vector formed by taking the first row of $\phi(pT)$ matrix. Often it is not difficult to determine $\phi_1(pT)$ to moderate accuracy with a slide rule for four or five values of p. The computer record is then examined to determine $\underline{v}(t_o)$ and the operation indicated by Eq. (4.5) carried out. This method was used to determine the true oscillation amplitudes presented in the last column of Table 4.1. Note that these results are quite close to those presented by Chow especially when it is recalled that the author's work is to slide rule accuracy and the describing function, which was used by Chow, is an approximate method.

As a further check, the results of digital simulation were compared with the phase-plane results presented by Mullin and Jury[13] for the plant $\frac{1}{s(s + 1)}$ with a unit step input and for sampling periods of 0.5

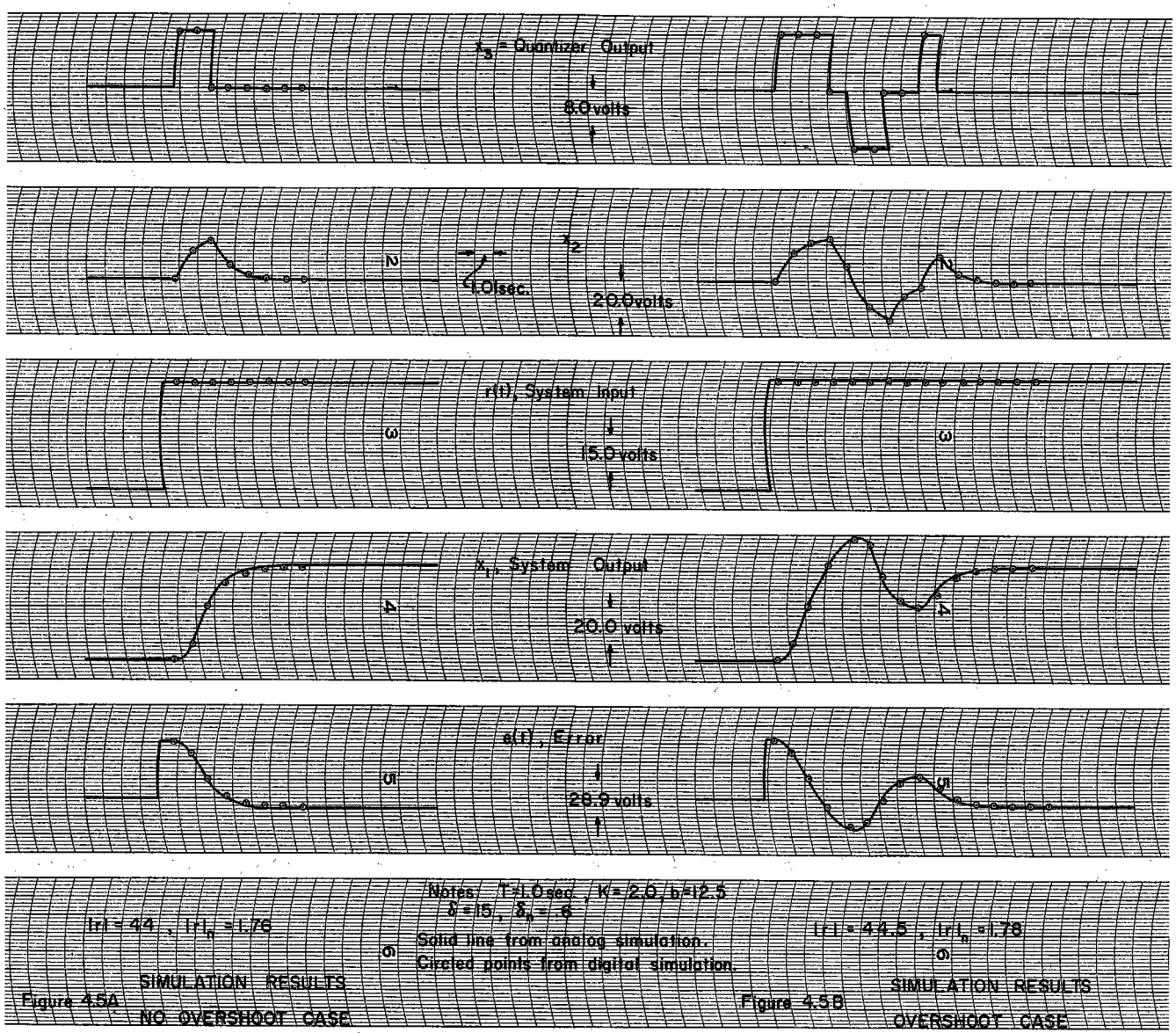


Figure 4.5A

Figure 4.5B

and 1.0 second. In the case of the 0.5 second sampling period, the system output was found to overshoot but to finally settle to a value of 1.0. On the other hand, the 1.0 second sampling period caused a limit cycle with a 6.0 second period to exist. The results of digital simulation were found to be in good agreement with those of Mullin and Jury in both cases.

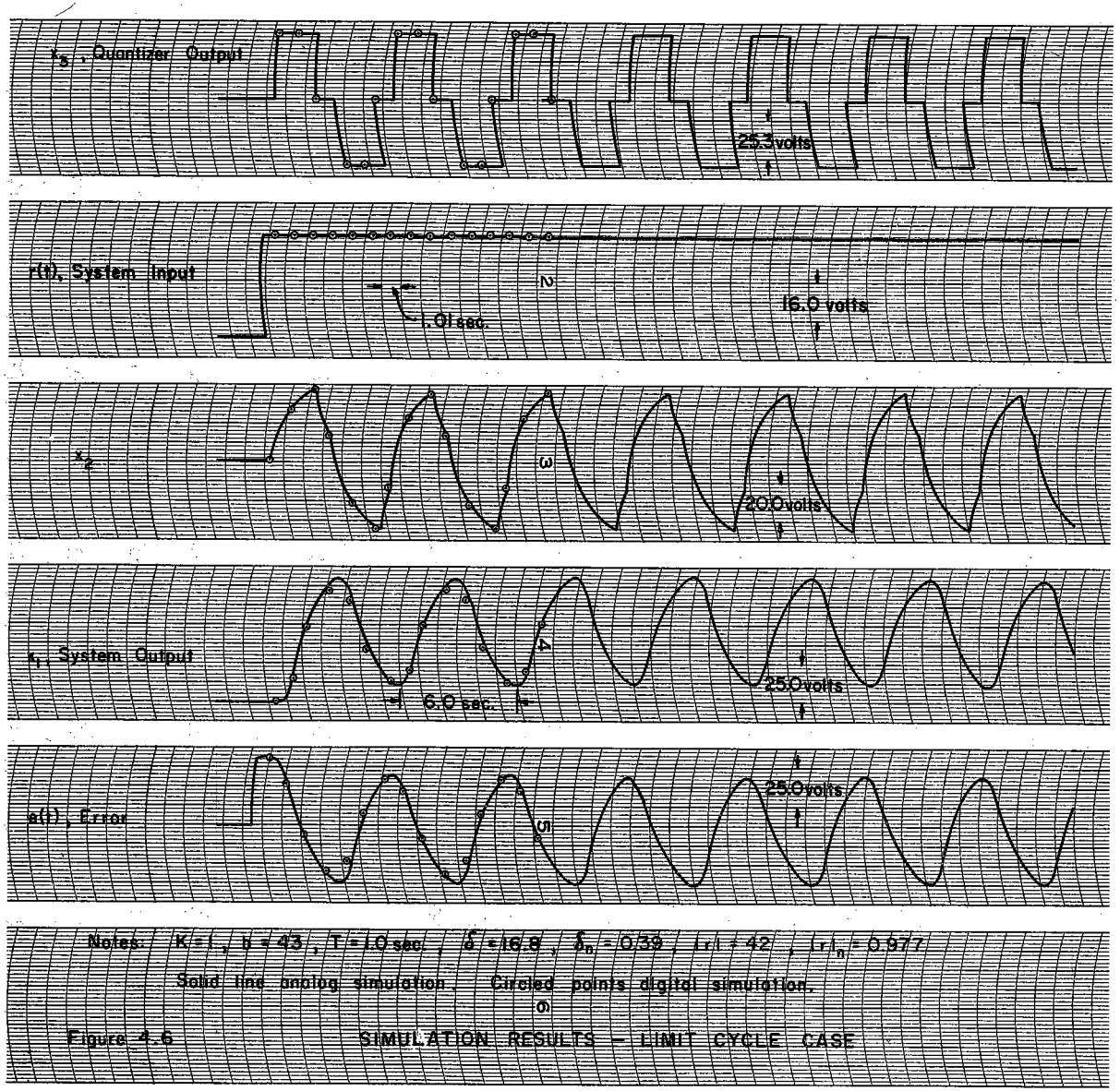
4.5 Comparison of Results from Digital with Analog Simulation

In order to provide an experimental check on the method of digital simulation, representative systems containing the plant $\frac{1}{s(s+1)}$ and subject to step inputs have been analyzed by the method of analog simulation described in Section 4.3. Typical results* obtained are presented here in Figs. 4.5, 4.6, and 4.7. with some additional results being presented, where appropriate, in other parts of this report. Also shown on these figures are the results of digital simulation under the same conditions. In plotting the digital simulation data, minor corrections were made to take into account small imperfections in the recordings; e.g. the fact that the chart speed is slightly less than 1.0 division per second and the fact that the arc made by the pens do not always agree perfectly with the arcs printed on the recording paper were taken into account.

Results for analog simulation of a system with a three level quantizer are shown by the solid line in Figs. 4.5 and 4.6, also shown

---

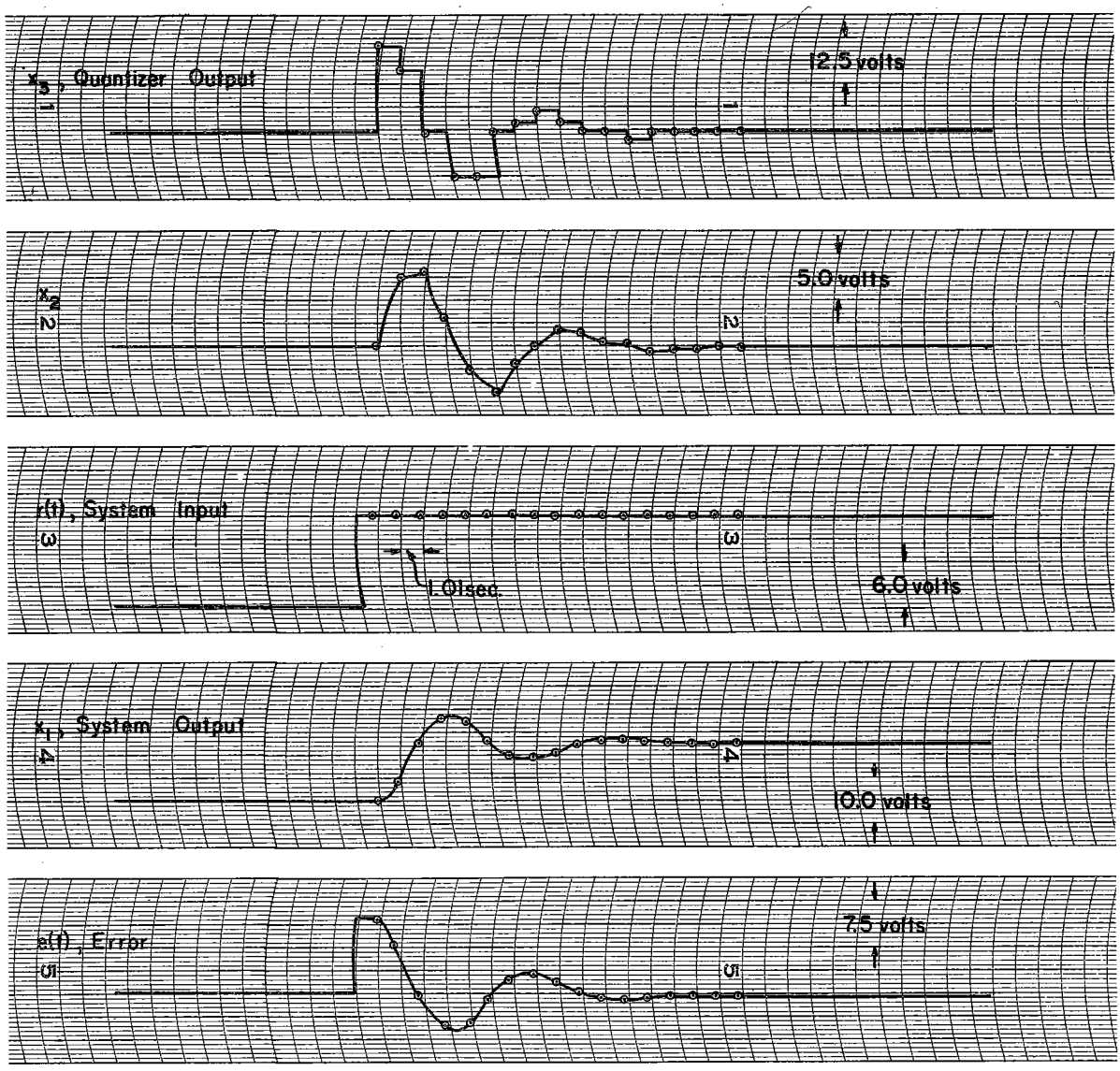
*The original graphs from the Brush Recorder were retouched to insure that good reproductions would be obtained.



Notes: $K = 1$, $b = 43$, $T = 1.0$ sec., $\delta = 16.8$, $\delta_n = 0.39$, $|r| = 42$, $|r|_n = 0.977$

Solid line analog simulation. Circled points digital simulation.

Figure 4-6 SIMULATION RESULTS -- LIMIT CYCLE CASE

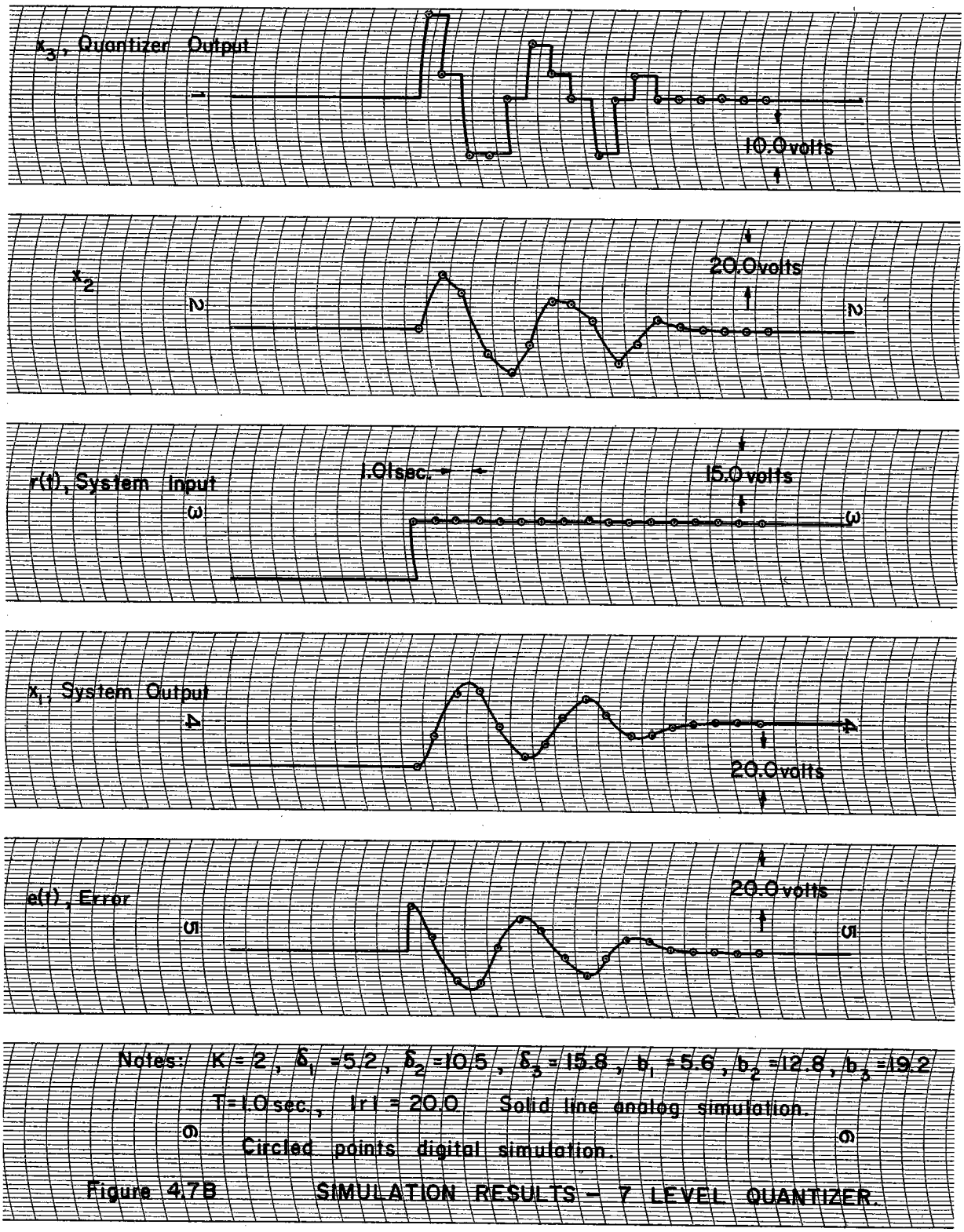


Notes: $|r| = 12$, $T = 1.0$ sec., $\delta_1 = 1.3$, $\delta_2 = 2.6$, $\delta_3 = 3.9$, $\delta_4 = 5.2$, $\delta_5 = 6.5$, $\delta_6 = 7.8$, $\delta_7 = 9.1$

$b_1 = 2.5$, $b_2 = 5.9$, $b_3 = 9.3$, $b_4 = 12.35$, $b_5 = 17.3$, $b_6 = 20.7$, $b_7 = 23.9$, $K = 0.5$

Solid line analog simulation. Circled points digital simulation.

Figure 4.7A SIMULATION RESULTS - 15 LEVEL QUANTIZER



Notes: $K = 2$, $\delta_1 = 5.2$, $\delta_2 = 10.5$, $\delta_3 = 15.8$, $b_1 = 5.6$, $b_2 = 12.8$, $b_3 = 19.2$
$T = 1.0$ sec., $|r| = 20.0$ Solid line analog simulation.
Circled points digital simulation.

Figure 4.7B SIMULATION RESULTS – 7 LEVEL QUANTIZER.

in these figures are points at the sampling instances, which were computed from digital simulation. Note that the results by the two methods of simulation give results which are in good agreement. The only change between Fig. 4.5A and Fig. 4.5B is that the normalized input has increased by .02. This causes a considerable difference in the transient response but the final output value is essentially the same. The difference in performance is easily explained in terms of the design graph presented and explained in Chapter 5. The fact that a limit cycle exists for the conditions of Fig. 4.6 is also easily explained in terms of the design graphs of Chapter 5. Using only the quantizer output and the Final Value Rule, the steady state value of output was predicted for both Figs. 4.5A and 4.5B and the predictions were found to be in agreement with the actual steady state output values obtained. A check of the steady state error in both figures reveals that both are well within $\pm\delta$, which is as it should be.

In all three figures the plot of x_2, which is a general state variable but corresponds to velocity in a position control system, against time contains discontinuities in its slope; this is reasonable in that $\dot{x}_2 = x_3 - x_2$ and x_3, the quantizer output, is a discontinuous function. Since the initiation of the input is not synchronized with the sampler, a random delay is observed in the first nonzero output from the quantizer on all figures. The delay in the output from the quantizer in turn causes a random delay in the overall system response by up to one sampling period, but it is the most common way such a system would operate in practice.

Essentially the same procedure described above for the three level quantizers was used to analyze multiple level quantizers; the result from two different cases are shown in Figs. 4.7A and 4.7B. Again the results of analog and digital simulation are in good agreement. Although the quantizer producing the results of Fig. 4.7A contains 15 levels, only 7 of these actual show in the figure. Naturally levels not shown can be made to appear and those appearing can be made to disappear if either the input magnitude or sampling period are changed in the proper way. It was shown by both methods of simulation that small changes in either the b values or the δ values of the quantizer due to noise or other causes can significantly influence the transient performance of the system; but of course, steady state response will always lie within $\pm\delta$, as long as a limit cycle is not produced.

As expected, it was observed that the system reponse appears to more and more resemble the response of a system without quantization as the number of levels increases. However, it appears that many systems could be designed to have quite satisfactory performance with only a few levels of quantization; this will be placed in quantitative terms in Chapter 5 where the design of a three level quantizer is considered.

In all of the analog simulation runs, the results observed were either predictable by digital simulation or definitely caused by system malfunction, drift, or noise. Drift in the decoder and in the network preceding Amplifier 7 were the most troublesome effects.

Although it was not difficult to control these by frequent checks and adjustments during the experimental work, careful redesign would be required in both of these areas in commercial equipment.

CHAPTER 5

DESIGN OF DIGITAL CONTROL SYSTEMS BY DIGITAL SIMULATION

5.1 General

This chapter presents design methods which have evolved from the analysis techniques of Chapters 3 and 4. First, the design graphs for a three level quantizer used in a specific second order system are presented and their salient features explained. The method of constructing design graphs applicable to any second order system is then presented. Finally, more detailed consideration is given to systems possessing overshoot.

5.2 Design of a Three Level Quantizer for a Second Order System

In order to see how digital simulation can be applied to the design of digital control systems, consider the three level quantizer system shown in Fig. 3.5. Interest in the three level quantizer stems from the fact that it is the simplest quantizer which can be used in the systems to be described without causing limit cycle oscillations. Moreover, it exhibits most of the same phenomena as do quantizers with a larger number of levels. For purposes of this example, assume that all the parameters of the system are fixed except δ, b, and the gain of the plant K. Assuming step inputs, it is the purpose of the design to satisfy specifications on allowable static error, on maximum settling time, and there are to be no limit cycle oscillations. Static error or accuracy is a function of input magnitude, but assuming it is desired to

operate the system in response to a fairly wide range of input magnitudes and that noise may be present to perturb the output, application of the Quiescent Plant Rule leads to the conclusion that the error will be limited to a band of $\pm \delta$ about the desired value in steady state. On the other hand, there are bounds to how small δ can be reduced before limit cycle oscillations occur. One example of this was presented in Chapter 3, but a much more complete presentation of the influence of δ on system performance is shown in Fig. 5.1. This figure was developed by using the computer program shown in Fig. 4.1 with the various computational rules helping to reduce the computing load. The following several points should be observed concerning Fig. 5.1.

1. The parameter b of the quantizer has the same effect on the output of the plant as does the gain of the plant K; thus, both parameters may be combined into a single quantity Kb.
2. The fact that the plant itself is linear allows a single set of universal boundaries normalized in terms of Kb to be plotted in Fig. 5.1 rather than having to make a new curve for each value of Kb. (See Section 5.7 for a more detailed explanation).
3. Depending on the value of normalized dead zone, $\delta_n = \frac{\delta}{Kb}$, and the value of normalized input magnitude $|r|_n = \frac{|r|}{Kb}$, the system response will fall into one of four regions: no response, limit cycle oscillations, overshoot, or no

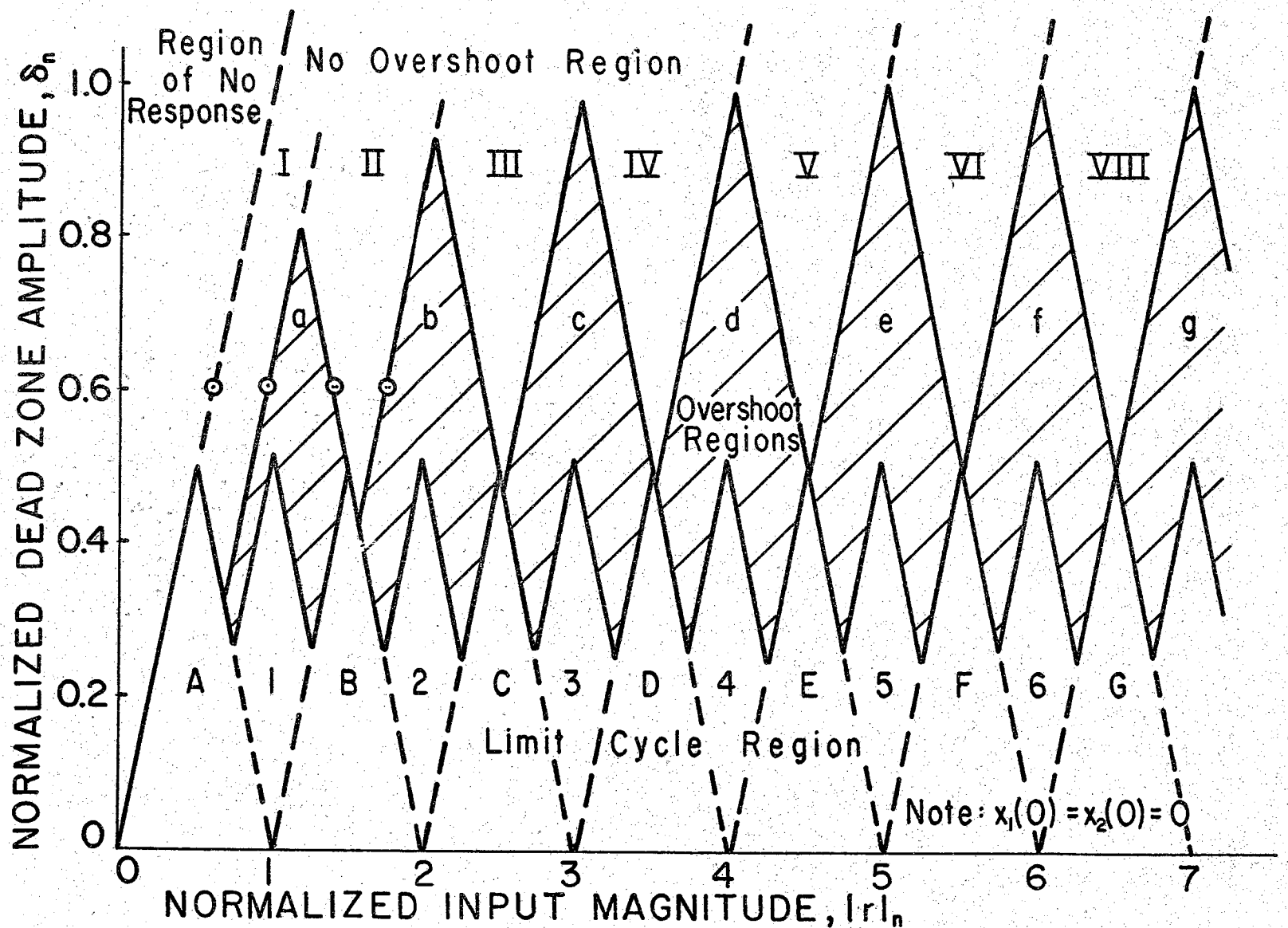


Fig. 5.1. Normalized Design Boundaries for the System of Fig. 3.5

overshoot. (The subregions set off by the dashed lines will be discussed starting in Section 5.3).

4. Although both the limit cycle as well as the overshoot boundaries are a function of input magnitude, little advantage can be taken of this fact unless the system is to be used with a very narrow range of input magnitudes. In addition, the overshoot boundaries are somewhat lower for small inputs, but again this is a rather special case. For most practical purposes, Fig. 5.1 shows that a minimum value of $\delta_n = 1.0$ is required to avoid overshoot and a minimum value of $\delta_n = .515$ is required to avoid limit cycle oscillations.

5. There appears to be approximately a factor of two to be gained in the minimum value of δ_n by allowing overshoot. However, in many cases this advantage can not be exploited because the long settling time produced by some inputs are objectionable. There is a band of values of $|r|_n$ along the right side of the overshoot regions for which long settling times are produced. For example, with $\delta_n = .58$ and $|r|_n$ between 1.3 and the overshoot boundary at 1.425 the minimum value of settling time (to within 5°/o of the final value) is 8.6 seconds. Moreover, narrowing the input range by allowing the left edge of the range to approach the overshoot boundary increases the minimum settling time until infinite settling time is reached at the boundary itself.

Initially, it will be assumed that the possibility of long settling times or just the fact that an overshoot is present rules out consideration of the overshoot case. However, system design with overshoot allowed will be considered in Section 5.9.

6. Since Fig. 5.1 is normalized in terms of Kb, it is seen that any numerical value of δ may be selected provided that it is used with the proper value of Kb. (Here $\delta_n = 1$ (δ = Kb) leads to no overshoot independent of input magnitude). It is thus seen that high static accuracy may be obtained by using a small value of Kb. However, settling time is again increased but not to the extent it is in some instances when overshoot is allowed.

An analog computer verification of the location of some of the boundaries shown on Fig. 5.1 was attempted. The results are indicated on the figure by means of the circled points at $\delta_n = 0.6$ and values of $|r|_n$ near 0.6, 0.96, 1.4 and 1.75. Each circle encloses two points although they usually are too close to be resolved on the figure. One point indicates the position of the left edge of the boundary and the other indicates the right edge as determined by analog simulation. Again note the close check between the results of digital and analog simulation. Incidentally, the complete analog record for the two points near $|r|_n = 1.75$ are shown on Fig. 4.5.

Fig. 5.2 shows the relationship between settling time and input magnitude for the smallest value of δ_n consistent with no overshoot

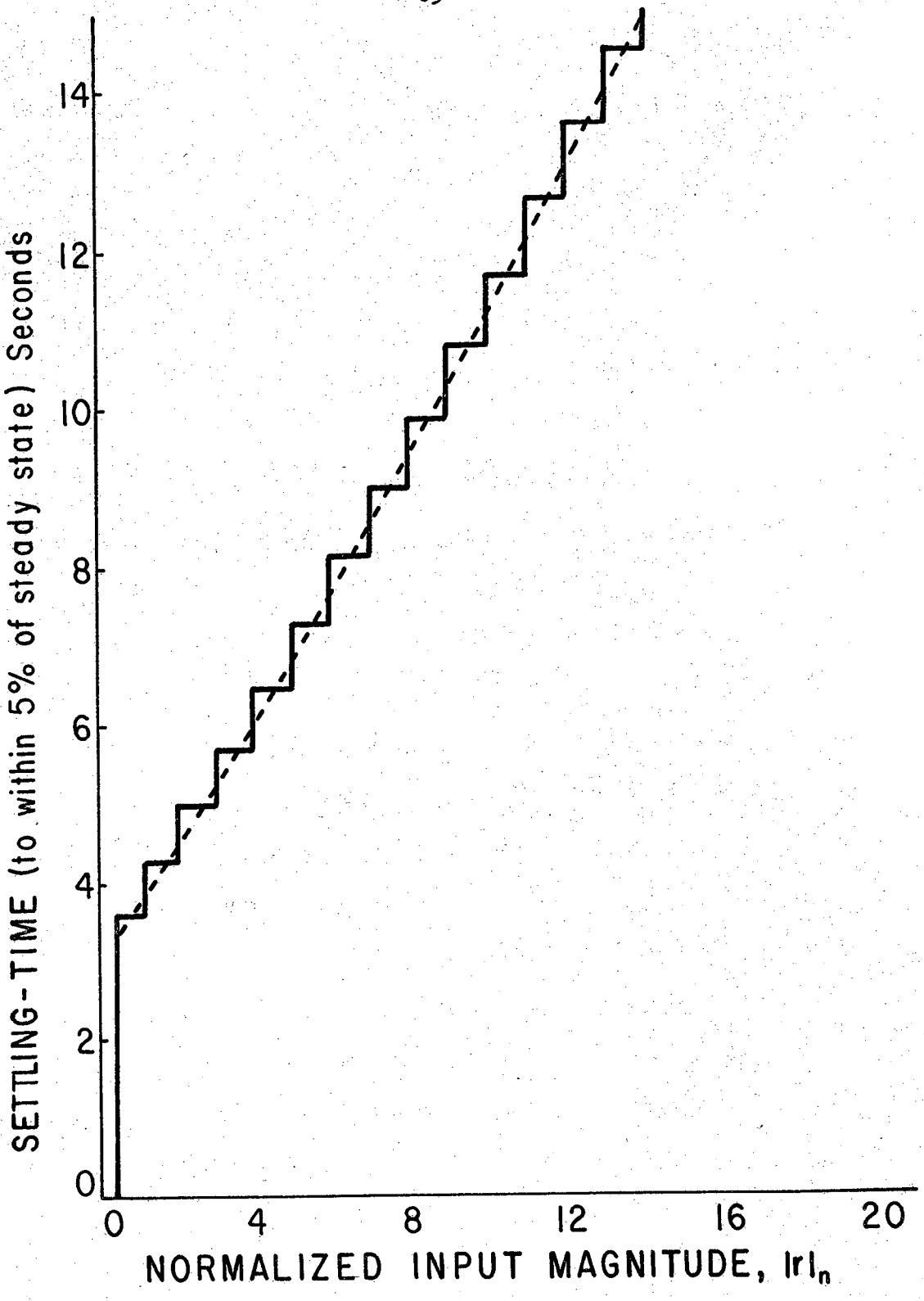
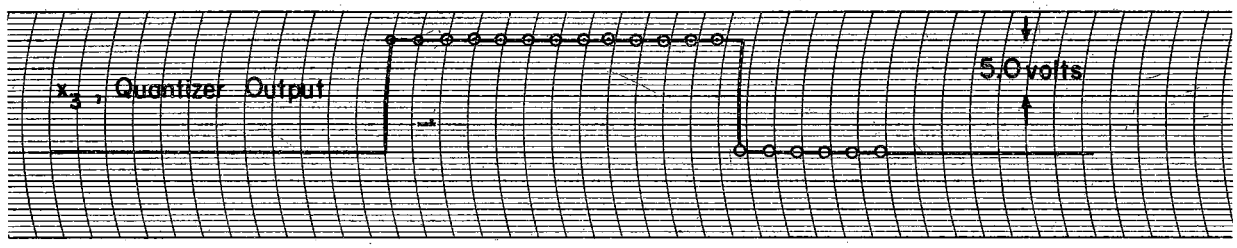
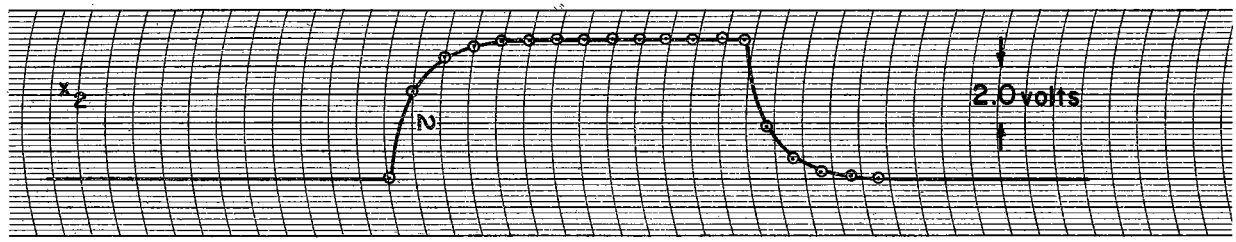
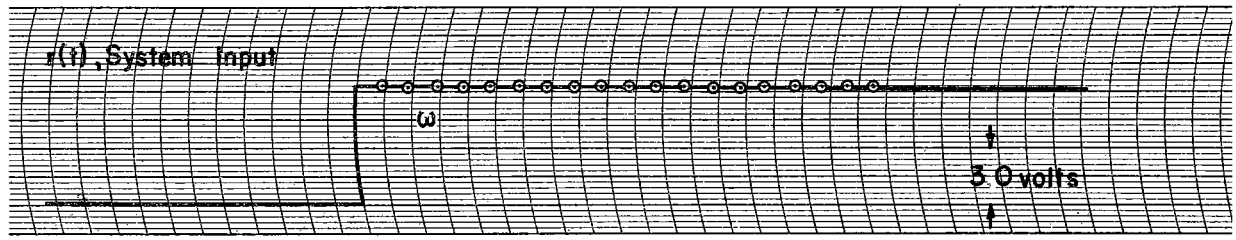


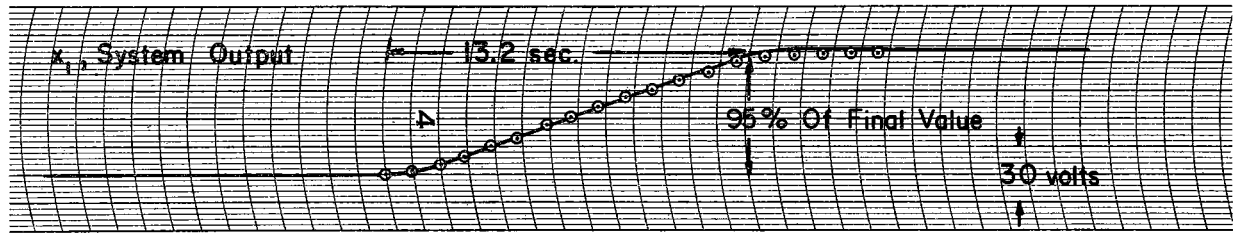
Fig. 5.2. Settling Time Design Curve for the System of Fig. 3.5

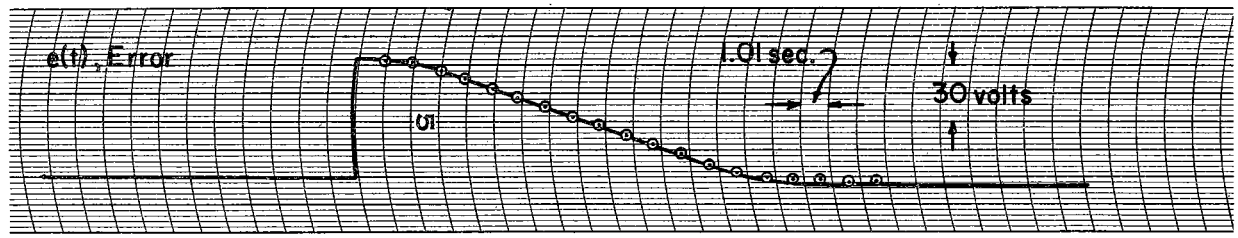
for any member of the ensemble of inputs up to the value of $|r|_n$ selected from the graph. Note that this figure is also normalized in units of Kb. Making use of Figs. 5.1 and 5.2 it is possible to establish a quantizer design which satisfies a set of specifications on settling time and maximum static error. For example, assume that the specification requires a system accuracy of at least .24 unit and the settling time should be no more than 14 seconds for an ensemble of inputs of magnitude no greater than 3.0 units. Select a value of $\delta = .24$ in order to satisfy the accuracy specification and initially assume the smallest value of Kb which would ever be required in this system, i.e., $Kb = \delta = .24$. The normalized dead zone amplitude and normalized input magnitude then becomes $\delta_n = 1$ and $|r|_n = \frac{3}{.24} = 12.5$. From Fig. 5.1 it is seen that the original assumption of $Kb = \delta$ is justified in order to prevent overshoot. Going to Fig. 5.2, it is seen that $|r|_n = 12.5$ leads to a settling time of 13.6 seconds. Thus, it is seen that a quantizer design having $Kb = \delta = .24$ will satisfy the original specifications. As a verification of this design, the system was simulated using the analog computer method described in Chapter 4 and Appendix B. A copy of the five channel recording obtained is shown in Fig. 5.3. Note that the settling time of 13.2 seconds is in good agreement with the value obtained from the design graph and that the curves from analog simulation are accurately verified by the points determined from digital simulation. As expected a steady state error exists in the output but this is within $\pm \delta$.











Notes: $|r| = 62.5$, $|r|_{ss} = 12.5$, $\delta = 5.0$, $\phi_n = 1$, $T = 1.0$ sec, $K = 0.5$, $b = 10$

Solid line from analog simulation. Circled points from digital simulation

Figure 5.3 SIMULATION RESULTS – DESIGN EXAMPLE

Of course, there will be situations for which the specifications on accuracy and settling time can not be satisfied simultaneously. When this occurs some compromise must be made. Either the specifications must be relaxed or some change must be made within the system; e.g. a quantizer with an additional set of levels may satisfy the specification. Another possibility that becomes more attractive for larger values of $|r|_n$ is to allow overshoot. This possibility will be explored further in Section 5.9.

5.3 Use of Computational Rules in Developing Design Graphs

The results presented in Section 5.2 and especially those of Fig. 5.1 could have been obtained by using the barrage technique of computing a multitude of points in the regions of interest and then trying to find some systematic behavior in the results. Of course this was not done, but as was mentioned before, the computational rules were employed to make the procedure more systematic and to reduce the number of computer runs needed. The reasoning used to develop the various boundaries and regions on Fig. 5.1 will now be presented. Initially it will be assumed that Kb = 1, but the normalization of both δ and $|r|$ will be discussed later. The Region of No Response is the simplest region to consider but the reasoning used is similar to that employed on the other regions. Analysis of this region proceeds as follows:

1. If $x_1(0) = 0$ and the plant was initially quiescent, the Quiescent Plant Rule, Eq. 3.39, yields

$$\delta > |r(kT)| \tag{5.1}$$

as the condition for the plant to remain quiescent.

2. Thus, Eq. (5.1) describes a region in which an initial quiescent plant with $x_1(0) = 0$ will remain quiescent. This region is called the Region of No Response on Fig. 5.1.

3. Note that the conclusion in 2 above holds not only for step inputs but for any function whose values are defined at the sampling points.

5.4 Limit Cycle Region

As may be seen from Fig. 5.1, the Limit Cycle Region is divided into a number of smaller regions. These will be discussed separately beginning with Region A.

1. In the discussion of the Final Value Rule, Section 3.5.3, it was found that the output in steady state, $x_1(\infty)$, for the system under consideration will be an integer, which is a special case since the steady state output of systems of this type will be quantized but not necessarily quantized to an integer value. (However, the following reasoning can also be used in the case of noninteger values).

2. If $0 < r < 1$ and $\delta < 1$ (where r represents the magnitude and sign of a step input), the only steady state values of x_1 which will satisfy the Quiescent Plant Rule, Eq. 3.39, are 0 or +1. (If $-1 < r < 0$, x_1 can only be 0 or -1).

3. With $x_1(\infty) = 1$, the Quiescent Plant Rule will not be

satisfied during any sampling period for which

$$\delta < |r(kT) - 1| \quad (\text{for } 0 < r < 1) \tag{5.2}$$

at the beginning of the sampling period and if the input is a step the fact that Eq. (5.2) is not satisfied for k = 0 means that it will not be satisfied for any sampling period.

4. For $x_1(\infty) = -1$, the equation corresponding to Eq. (5.2) is

$$\delta < |r(kT) + 1| \quad (\text{for } -1 < r < 0). \tag{5.3}$$

5. Equations (5.2) and (5.3) may be combined into the following equation which covers the range $-1 < r < 1$:

$$\delta < \Big|\, |r| \qquad -1 \,\Big| \tag{5.4}$$

6. With $x_1(\infty) = 0$ and a step input, reasoning similar to that in 3 leads to

$$\delta < |r| \tag{5.5}$$

as the condition for which the system can never be quiescent.

7. The above discussion leads one to the conclusion that any input within the region defined by Eqs. (5.4) and (5.5) cannot lead to a quiescent system. Since Bertram[23] has shown that quantization cannot cause a previously stable* sampled-data system to be unstable, then the above must be

---

*The definition of stability used here is that the output must be bounded for a bounded input.

a limit cycle region. This region has been labeled A on Fig. 5.1. (Note that $\delta < 1$ for the entire Region A; thus, the assumption in 2 above is justified).

Reasoning similar to that used on Region A will now be applied to Region B.

1. Assume $1 < |r| < 2$, $\delta < 1$, and $x_1(0) = 0$.
2. From the Quiescent Plant Rule, Eq. (3.39), the plant may become quiescent if

$$|r(kT) - x_1(kT)| < \delta \tag{5.6}$$

for $k > k_{min}$.

3. Recalling that $x_1(\infty)$ must be an integer for this particular system and using the assumptions of 1, it is found that $x_1(\infty) = 1$, $x_1(\infty) = 2$ or in some cases both are the only values of $x_1(\infty)$ which satisfy Eq. (5.6). ($x_1(\infty) = -1$ and $x_1(\infty) = -2$ for negative inputs).
4. Assuming $x_1(\infty)$ is +1 or -1, the Quiescent Plant Rule will not be satisfied if

$$\delta < \left| |r| - 1 \right| \tag{5.7}$$

5. Assuming $x_1(\infty)$ is +2 or -2, the Quiescent Plant Rule will not be satisfied if

$$\delta < \left| |r| - 2 \right| \tag{5.8}$$

6. From 3 the plant can only be quiescent if $x_1(\infty) = \pm 1$ or ± 2 but even if this is true the plant will not be quiescent if both Eqs. (5.7) and (5.8) are satisfied.
7. Therefore, Eqs. (5.7) and (5.8) define a region (Region B

on Fig. 5.1) where the plant is never quiescent. Since the system is stable this region must be part of the Limit Cycle Region.

By arguments similar to those used on Region B, it may be shown that Regions, C, D, E, etc. are also part of the Limit Cycle Region.

It is desirable to find a way of proving that Regions 1, 2, 3, etc. are also regions of limit cycle oscillation. Although only a few computer runs are required to establish that the boundary satisfies the conditions for limit cycle oscillations and although it seems intuitively clear that the entire region is one of limit cycle oscillation, the best that the writer has been able to do thus far is to make the best use possible of the Input Signal Range Rule and the Dead Zone Range Rule to systematically show that every point in these regions is one of limit cycle oscillation. On the other hand, once it has been shown that the outline of these regions is correct and that a sizable portion of the area around their boundaries lead to limit cycle oscillations it is immaterial for most engineering purposes if the regions contained a few holes in which the conditions for limit cycle oscillations are not satisfied.

5.5 No Overshoot Region

Like the Limit Cycle Region, the No Overshoot Region is divided into a number of smaller regions which will be discussed separately. First consider Region I.

1. From Table 3.2 at t = 1 second, $x_1(T) = .368$. This value will be obtained for the system under consideration irrespective of the magnitude of r as long as $x_1(0) = 0$,

the input does not fall in the Region of No Response, and r is positive. (For r negative, $x_1(T) = -.368$).

2. Since

$$e(kT+) = r - x_1(kT) \tag{5.9}$$

$$e(T+) = r - .368 \qquad r > 0 \tag{5.10}$$

and

$$e(T+) = r + .368 \qquad r < 0 \tag{5.11}$$

3. The output of the quantizer at t = 1 second will be zero as long as $\delta > |e(T+)|$. Substituting Eqs. (5.10) and (5.11) into this inequality and writing the result as a single inequality, which is good for all values of r, the following expression results:

$$\delta > \left| \, |r| - .368 \, \right| \tag{5.12}$$

4. Starting with a quiescent plant and allowing a single pulse one sampling period long to come from the quantizer, it is found* that x_1 monotonically approaches $x_1 = 1$ ($x_1 = -1$ for negative pulse). In this case, Eq. (5.9) yields the following monotonic behavior for the system error:

$$e(kT+) \longrightarrow r - 1 \qquad r > 0 \tag{5.13}$$

or

$$e(kT+) \longrightarrow r + 1 \qquad r < 0 \tag{5.14}$$

5. As long as δ satisfies the equation

---

*See Eq. (6.32).

$$\delta > \left| |r| - 1 \right| \qquad \text{(any } r\text{)} \tag{5.15}$$

obtained from Eqs. (5.13) and (5.14) and the requirement that $\delta > |e(kT+)|$ leads to zero quantizer output, the output of the quantizer will always be zero after the first sampling period.

6. Eq. (5.12), Eq. (5.15) and the boundary of the Region of No Response define a region labeled I in Fig. 5.1 in which x_1 monotonically approaches $x_1 = 1$ (or $x_1 = -1$); therefore, it is part of the No Overshoot Region.

Region II was found in a manner very similar to that used for Region I.

1. If the input does not satisfy Eq. (5.12), the quantizer output will maintain its maximum (or minimum) value for at least the first two sampling periods.
2. From Table 3.2 at t = 2 seconds, $x_1(2T+) = 1.135$.
3. Now the output of the quantizer at t = 2 seconds will be zero as long as $\delta > |e(2T+)|$. Therefore, the requirement for zero output at t = 2 second is

$$\delta > \left| |r| - 1.135 \right|, \tag{5.16}$$

which is valid for both positive and negative r.

4. For a pulse two sampling periods long from the quantizer, x_1 monotonically approaches $x_1 = 2$.* If Eq. (5.16) is satisfied the following equation gives the condition under which the output from the quantizer will always be zero

---

*See Eq. (6.32).

after the second sampling period:

$$\delta > \left| |r| - 2 \right| \qquad \text{(for all r)} \tag{5.17}$$

5. Eqs. (5.12), (5.16), and (5.17) define a region (II on Fig. 5.1) for which x_1 monotonically approaches $x_1 = 2$ (or $x_1 = -2$); therefore it also is part of the No Overshoot Region.

The same type of argument used to determine Regions I and II was used to determine all of the other regions (III, IV, V, etc.) which make up the No Overshoot Region. Now Region I required a pulse of length T to bring the system to the steady state value of 1 and Region II required a pulse of length 2T to bring the system to the steady state value of 2. By the same reasoning it is found that a pulse 3T long is required to bring the system to the steady state value of 3 in Region III, and the values for Regions IV, V, VI, etc. follow in the same way.

5.6 Overshoot Region

Like the other major regions the Overshoot Region is also broken into smaller regions. These will be considered separately beginning with Region a.

1. Again assuming that $x_1(0) = 0$, then at t = 1 second, $x_1(T) = \pm .368$ depending only on the sign of r as long as r does not fall into the region of no response.

2. Using reasoning similar to that used previously it is found that

$$\delta < \left| \; |r| - .368 \; \right| \tag{5.18}$$

is the condition for the quantizer output to continue to

maintain its maximum (or minimum) output during the second sampling period.

3. With $-2 < r < 2$ and $\delta < 1$, a system satisfying the equation

$$\delta < \left| \, |r| - 2 \, \right| \tag{5.19}$$

cannot be quiescent with $x_1 \geq 2$ ($x_1 \leq -2$ for $r < 0$).

4. A system satisfying Eq. (5.18) will settle with $x_1(\infty) \geq 2$ unless the quantizer output changes sign for some later sampling period ($x_1(\infty) \leq -2$ for negative inputs). But a system which also satisfies Eq. (5.19) cannot be quiescent with $x_1(\infty) \geq 2$ (or $x_1(\infty) \leq -2$).

5. Therefore, any system satisfying both Eqs. (5.18) and (5.19) must overshoot if it is to become quiescent.

6. The above discussion does not prove that all points bounded by Eqs. (5.18) and (5.19) settle out to a steady state value of $x_1(\infty) = 1$ because it has already been proved that Region B in Fig. 5.1, which is a subregion of this, is one of limit cycle oscillations. However, systematic use of the Dead Zone Range Rule and the Input Range Rule reveals that Region a in it entirety is an overshoot region. Details concerning the settling time required by the subregions making up Region a are presented in Section 5.9.

Arguments similar to those used above show that Regions b, c, d, etc. have the same properties as that of Region a, i.e., they can either be overshoot or limit cycle regions but actual calculations reveal that they are overshoot regions only.

5.7 The Significance of Kb as a Normalizing Parameter

For the systems discussed here, the plant itself always is assumed to be linear. Thus, the value of the system output at any time and for that matter the value of all of the plant state variables at any time is linearly related to the size of the quantity Kb. Thus, if Fig. 5.1 is considered to be for Kb = 1 for the moment and then inquiry is made (by reviewing the arguments of Sections 5.3 through 5.6) as to what happens to the regions in the figure as Kb is increased (or decreased) it is seen that the Kb acts as a magnification factor which is applied to both axes. Thus, one can obtain a universal or normalized set of curves by plotting $\delta_n = \frac{\delta}{Kb}$ and $|r|_n = \frac{|r|}{Kb}$ instead of δ and $|r|$ themselves, which is what was done in Fig. 5.1.

5.8 Application to Other Systems

The general method described in this section can be used to develop quantizer design curves for other systems. Moreover, depending on the system specifications, it may not always be necessary to make the curves as detailed as that shown in Fig. 5.1. For example, if the design specifications require no overshoot, only the boundary of the no overshoot region needs to be drawn; this saves a considerable amount of work and computing time.

5.9 System Design under the Condition that Overshoot is Allowed

In order to obtain maximum steady state accuracy with minimum settling time, it is tempting to try to operate the system in the overshoot region. Some of the consequences of this will now be discussed.

Figure 5.4 shows some of the structure occurring in and around the first overshoot region. Basically, it is an enlargement of Regions a, I, II and III of Fig. 5.1 with the settling times to within 5°/o of the final value indicated on each of the subregions. Note that very long settling times are produced if the system operating point on Fig. 5.4 happens to lie inside but very close to the right side of the overshoot boundary. Table 5.1, which was prepared from Figs. 5.2 and 5.4, presents a comparison of the settling times obtained with and without overshoot, i.e., $\delta_n = .6$ and $\delta_n = 1.0$ respectively. This example uses the same system accuracy, $\delta = .24$, as was used in the example given in Section 5.2. For inputs within certain ranges the settling time with overshoot is equal to or less than that with no overshoot; and in other ranges the settling time is slightly longer with overshoot; and in other ranges it is considerably longer with overshoot. These results are typical of those obtained for all of the overshoot regions. However, as larger inputs are considered it appears that a greater percentage of the total range of inputs lead to shorter settling times in the overshoot case. For example, with inputs in the range 2.54 to 3.0, with $\delta = .24$, and again comparing the results for $\delta_n = .6$ with those for $\delta_n = 1.0$, approximately 70°/o of the cases result in a shorter settling time for the overshoot case and only approximately 6°/o of the inputs have settling times with overshoot which are more than 2.2 seconds longer than without overshoot. Moreover, in those instances where the very long settling times occur, the output is approximately within $\pm\delta$ for the entire latter portion of the settling period. Thus, in many

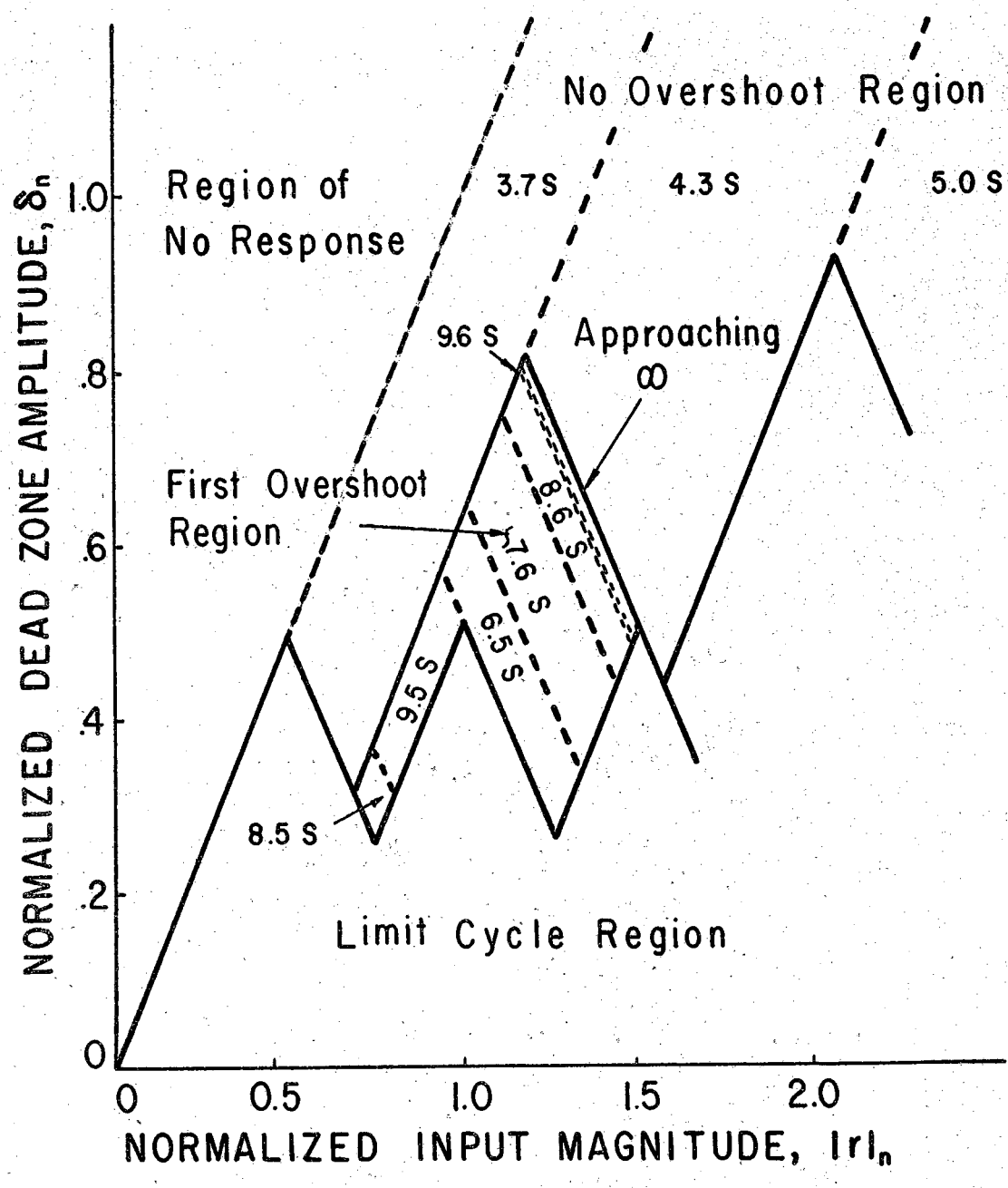


Fig. 5.4. Structure within and around the First Overshoot Region
(Settling-times, in seconds, to within 5°/o of steady state indicated)

Table 5.1

Comparison of Settling Times Between the Overshoot and No Overshoot Cases

Range of Input Magnitudes, $\lvert r \rvert$	0 to .24	.24 to .32	.32 to .38	.38 to .43	.43 to .51	.51 to .54	.54 to .56	.56 to .60
Settling Time, Sec. $\delta_n = 0.6$ (Overshoot)	No Resp.	3.7	3.7	6.5	7.6	8.6	≥ 9.6	4.3
Settling Time, Sec. $\delta_n = 1.0$ (No Overshoot)	No Resp.	3.7	4.3	4.3	4.3	5.0	5.0	5.0

designs it may be desirable to allow overshoot in order to decrease the average settling time of the system in response to an ensemble of inputs and accept the few cases where the settling times are long.

CHAPTER 6

CLOSED FORM SOLUTIONS

6.1 General

The development of some closed form solutions for quantized systems will now be presented. Closed form solutions, which are universally applicable to all input types, multiple level quantizers, etc., are desired; but, at the present time, their development appears to be an almost impossible task. However, solutions for certain special cases even though they may be only approximate in some respects, are believed to be very valuable in the following aspects:

1. They provide general physical insight into the operation of quantized systems.
2. In those cases where available closed form solutions are exact or where approximate solutions are satisfactory, system design or analysis can be completed in a short time. (Approximate solutions should be particularly valuable in preliminary design and feasibility studies).
3. In those cases where the approximate closed form solutions are not good enough as final results, they can provide a starting point for more accurate numerical methods.

6.2 First Order System with a Step Input

6.2.1 Conditions for Elimination of Limit Cycle Oscillations

The closed form solution for the first order system shown in

Fig. 6.1, with a step input, will now be developed. It is seen from Fig. 6.1 that if $e(kT+) \geq \delta$, $N(kT+) = b$. At $t = 0$ let $e(0+) \geq \delta^*$, then $N(0+) = b$ and it will continue that way through succeeding sampling intervals until the end of the sampling interval, n at which $e(nT+) < \delta$. Thus, the system output, $x_1(t)$, for $0 \leq t \leq nT$ may be obtained by allowing a pulse of length nT and amplitude b to exist at the input to the plant. In this case the input to the plant is

$$x_2(t) = b\left[u(t) - u(t - nT)\right] \tag{6.1}$$

Taking the Laplace transform of Eq. (6.1),

$$X_2(s) = b\left[\frac{1 - e^{-snT}}{s}\right]; \tag{6.2}$$

therefore,

$$X_1(s) = \frac{Kb}{s}\left[\frac{1 - e^{-snT}}{s}\right] + \frac{x_1(0)}{s} \tag{6.3}$$

and

$$x_1(t) = Kb\left\{ t\left[u(t) - u(t - nT)\right] + nT\, u(t - nT)\right\} + x_1(0) \tag{6.4}$$

where $x_1(0)$ is the initial value of the output.

Case I: The situation where $|e(nT+)| < \delta$ will be called Case I. For this case, $N(nT+) = 0$, but the output of an integrator (such as the present plant) will remain constant at the value it had when its input went to zero. Now the overall system input r(t) is constant and if the $x_1(t)$ does not change, e(t) can not change. Therefore, the

---

*The results for $e(0+) \leq -\delta$ throughout this chapter follow in a similar way to those for $e(0+) \geq \delta$.

system has reached steady state. Note that under these conditions, Eq. (6.4) is valid for all $t \geq 0$, and from Eq. (6.4) the steady state value of the output, $x_1(\infty)$, is

$$x_1(\infty) = KbnT + x_1(0) \tag{6.5}$$

The steady state system error, $e(\infty)$, becomes

$$e(\infty) = A - KbnT - x_1(0) \tag{6.6}$$

where A is the amplitude of the step input.

Case II: The situation were $e(nT+) < -\delta$ will be known as Case II. (By definition it is impossible for $e(nT+) > \delta$). Then $x_2(t) = -b$ for $nT < t \leq (n + 1)T$ and the input to the plant for $0 \leq t \leq (n + 1)T$ becomes

$$x_2(t) = b\Big[u(t) - u(t - nT)\Big] - b\Big\{u(t - nT) - u\Big[t - (n + 1)T\Big]\Big\} \tag{6.7}$$

Proceeding in the same way as previously it is found that

$$x_1(t) = Kb\Big\{t\Big[u(t) - 2u(t - nT) + u(t - \overline{n + 1}\ T)\Big]\Big\} + KbT\Big\{n\Big[2u(t - nT) - u(t - \overline{n + 1}\ T)\Big] - u(t - \overline{n + 1}\ T)\Big\} + x_1(0) \tag{6.8}$$

which is valid for $0 \leq t \leq \overline{n + 1}\ T$.

From Eq. (6.8)

$$x_1(\overline{n + 1}\ T) = Kb\ T(n - 1) + x_1(0) \tag{6.9}$$

and

$$x_1(\overline{n - 1}\ T) = Kb\ T(n - 1) + x_1(0) \tag{6.10}$$

The output, the error, and hence the input to the plant are the same for the (n - 1) and the (n + 1) sampling instances, but the response of

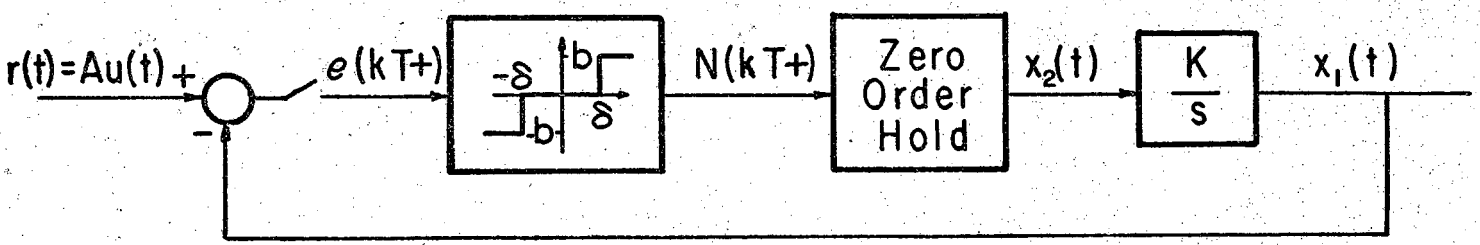


Fig. 6.1. A First Order Digital Control System

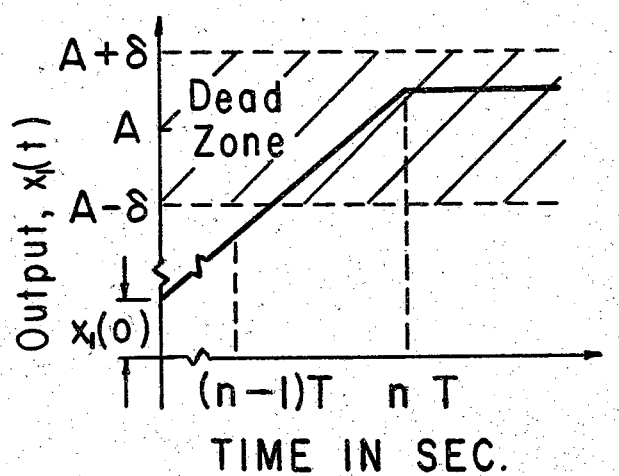


Fig. 6.2. Typical Response
(System of Case I)

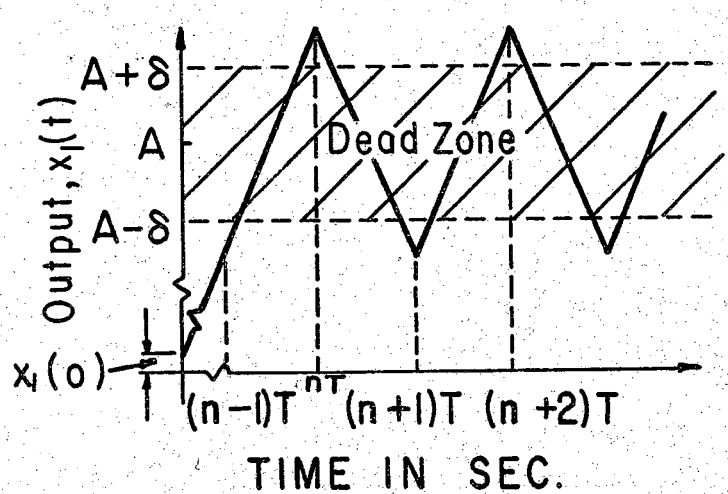


Fig. 6.3. Typical Response
(System of Case II)

an integrator plant is determined only by its input and the initial condition on the integrator. It thus appears that the system is in limit cycle oscillation with period 2T. Typical results for Cases I and II are shown in Figs. 6.2 and 6.3 respectively.

The condition under which a limit cycle will not exist is

$$-\delta < e(nT+) < \delta \tag{6.11}$$

Since the error is greater than or equal to the dead zone amplitude at $t = \overline{n-1}\,T$,

$$e(\overline{n-1}\,T) \geq \delta \tag{6.12}$$

Now

$$e(\overline{n-1}\,T) = r(\overline{n-1}\,T) - x_1(\overline{n-1}\,T) = A - x_1(\overline{n-1}\,T) \tag{6.13}$$

After substituting Eq. (6.10) into Eq. (6.13) and the resulting equation into Eq. (6.12), it is found that

$$A - KbT(n-1) - x_1(0) \geq \delta \tag{6.14}$$

or

$$A - KbT(n-1) - x_1(0) - h = \delta \tag{6.15}$$

where h is a positive quantity or zero, which will be determined later.

Writing $e(nT+)$ in Eq. (6.11) in terms of $r(nT)$ and $x_1(nT+)$ and substituting for $x_1(nT+)$ from Eq. (6.4), the following equation is obtained:

$$-\delta < A - KbT\,n - x_1(0) < \delta \tag{6.16}$$

Now solve Eq. (6.15) for A and substitute into Eq. (6.16)

$$-\delta < \delta - KbT + h < \delta \tag{6.17}$$

Solving the left portion of this inequality for δ

$$\delta > \frac{KbT - h}{2} \tag{6.18}$$

From the right portion of Eq. (6.17),

$$h < KbT \tag{6.19}$$

but $h \geq 0$; therefore,

$$0 \leq h < KbT \tag{6.20}$$

From Eqs. (6.18) and (6.20) the values of δ which will prevent limit cycle oscillations range from $\delta > 0$ to $\delta > \frac{KbT}{2}$ depending on the values of A and $x_1(0)$ which in turn control the location of $x_1(nT)$ within the band $\pm\delta$ about A at $t = nT$. This may be seen on Fig. 6.2 where the condition $\delta > 0$ corresponds to a value of A and $x_1(0)$ which leads to $x_1(nT)=A$, and the condition $\delta > \frac{KbT}{2}$ corresponds to values of A and $x_1(0)$ which lead to $x_1(nT) = A - \frac{KbT}{2}$ or $x_1(nT) = A + \frac{KbT}{2}$. If the system is to be designed to operate with a range of values of A and $x_1(0)$ at least as wide as $2\delta = KbT$, then the condition $\delta > \frac{KbT}{2}$ must be maintained. Since the great majority of the cases of interest in practice, result in ranges in either A or $x_1(0)$ of at least KbT, the following will be adopted as a system design criterion:

$$\delta > \frac{KbT}{2} \tag{6.21}$$

or

$$T < \frac{2\delta}{Kb} \tag{6.22}$$

Moreover, a value of δ satisfying Eq. (6.21) or a value of T satisfying Eq. (6.22) also assures that the system will not go into a limit cycle after a noise disturbance because a noise disturbance can merely be

considered to establish a new value of $x_1(0)$, and Eqs. (6.21) and (6.22) were derived without restrictions on $x_1(0)$.

6.2.2 System Time Response

Define the system response time t_r as the time required for the output to reach 95°/o of the final value. Assuming that it is necessary to have the final value equal to the desired value, A, the value of the output when the time response is measured becomes $x_1(t_r) = .95A$. From Eq. (6.4) with $x_1(0) = 0$, it is found that

$$t_r = \frac{.95A}{Kb} \tag{6.23}$$

or

$$Kb = \frac{.95A}{t_r} \tag{6.24}$$

Sampling at T intervals causes the above equations to be approximate in some instances; this is considered further in the next paragraph. After substituting Eq. (6.24) into Eq. (6.21) the following equation is obtained:

$$\delta > \frac{.475AT}{t_r} \tag{6.25}$$

or

$$T < \frac{\delta\, t_r}{.475A} \tag{6.26}$$

From the Quiescent Plant Rule, δ is the maximum steady state error for this system; Eq. (6.26) then becomes

$$T < 2.105\, f\, t_r \tag{6.27}$$

where $f = \frac{\delta}{A}$ is the maximum fractional steady state error. Equations

(6.24) and (6.27) are the design equations for a single integrator system using a three level quantizer.

If it is desired that t_r as calculated from Eq. (6.23) be the exact time response for a particular A, it is necessary that n, determined from Eq. (6.5) by letting $x_1(\infty) = A$ and $x_1(0) = 0$, be an integer. This equation is

$$n = \frac{A}{KbT} \tag{6.28}$$

If n as calculated from Eq. (6.28) is not an integer, the system response will be that for the next highest integer. To find the values of A and t_r which corresponds to the new value of n, it is merely necessary to substitute the new value of n into Eq. (6.28) and solve for the new value of A. By using the new value of A in Eq. (6.23) the true value of t_r can be determined. Variations of this procedure may be required depending on which system parameters are part of the specification and which are being determined. Moreover, if the minimum δ allowed by Eq. (6.25) is not used the system output may not reach A in which case the settling time as calculated from Eq. (6.23) will be too long.

6.2.3 Example

The following example demonstrates the use of the above equations.

Given: The following specifications are placed on the system shown in Fig. 6.1: $f \leq .04$, $K = 2$, $t_r \leq 0.1$ second and the system is to operate with an ensemble of inputs up to $A = 2$ without going into limit cycle oscillation. Find the parameters T, b and δ of the system.

Solution: From Eq. (6.27), $T < 0.00842$ second and from Eq. (6.24),

Kb = 19. Since $f = \frac{\delta}{A}$, $\delta = (.04)(2) = 0.08$. Checking the value of n by using Eq. (6.28), it is found that n = 12.5. Thus, the actual system response will have n = 13 which would then make the settling time slightly larger than the specified value. If necessary, a correction can be made by selecting n = 13 but keeping the same nT product; thus, the new value of T becomes $T = \frac{(12.5)(.00842)}{13} = 0.00809$ second. In this case all the other system parameters will remain unchanged except that δ can now be slightly smaller. From Eq. (6.25) the new δ is found to be $\delta > 0.0768$.

6.3 Second Order System with a Step Input

6.3.1 Conditions Insuring Response without Overshoot

A closed form solution for the second order system with a step input, shown in Fig. 6.4, will now be developed under the restriction that the system output does not overshoot. Similar to what was done in Section 6.2, at t = 0 let $e(0+) \geq \delta$; then, $N(0+) = b$ and it will continue that way until the end of the sampling interval, n, during which $e(t) < \delta$. The output, $x_1(t)$, for $0 \leq t \leq nT$ is obtained by considering a pulse of length nT and amplitude b to exist at the input to the plant. Thus

$$x_3(t) = b\left[u(t) - u(t - nT)\right] \tag{6.29}$$

Taking the Laplace transform

$$X_3(s) = \frac{b(1 - e^{-snT})}{s} \tag{6.30a}$$

Therefore,

$$X_1(s) = G(s)\, X_3(s) + \frac{x_1(0)}{s} = \frac{K}{s(s + a)} \cdot \frac{b(1 - e^{-snT})}{s} + \frac{x_1(0)}{s} \tag{6.30b}$$

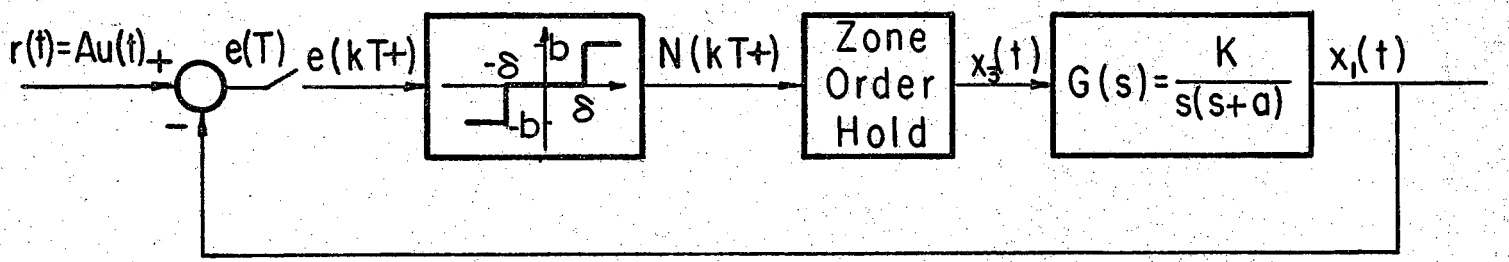


Fig. 6.4. A Second Order Digital Control System

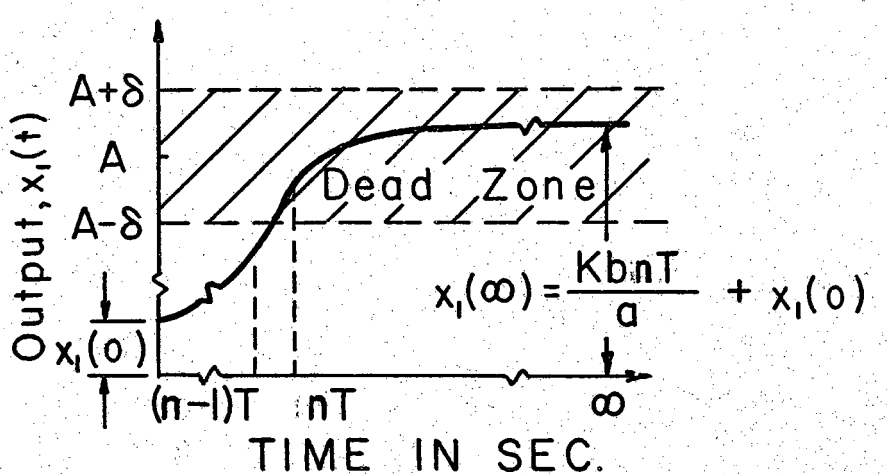


Fig. 6.5. Typical Response for the System shown in Fig. 6.4.

and

$$x_1(t) = \frac{Kb}{a}\left\{\begin{array}{l} t\left[u(t)-u(t-nT)\right] - \frac{1}{a}\left[u(t)-u(t-nT)\right] + nT\,u(t-nT) \\ \\ + \frac{e^{-at}}{a}\left[\,u(t) - e^{anT}\,u(t-nT)\,\right] \end{array}\right\} + x_1(0) \qquad (6.31)$$

For $t \geq nT$, Eq. (6.31) yields

$$x_1(t) = \frac{Kb}{a}\left[nT + \frac{e^{-at}}{a}\left(1 - e^{anT}\right)\right] + x_1(0) \qquad (6.32)$$

and in steady state, $t \longrightarrow \infty$,

$$x_1(\infty) = \frac{KbnT}{a} + x_1(0) \qquad (6.33)$$

For $t \leq nT$, Eq. (6.31) yields

$$x_1(t) = \frac{Kb}{a}\left[t - \frac{1}{a} + \frac{e^{-at}}{a}\right] + x_1(0) \qquad (6.34)$$

Assume that it is necessary to design the system in such a way that the output $x_1(t)$ does not overshoot its steady state value for any value of A or $x_1(0)$. In this case a typical system response will be as shown in Fig. 6.5.

Examination of Fig. 6.5 reveals that

$$A + \delta > x_1(\infty)^* \qquad (6.35)$$

and

$$A - \delta = x_1(\overline{n-1}\,T) \qquad (6.36)$$

---

*Implied in this inequality is the requirement that $x_3(t) = 0$ $t \geq nT$.

are the two extreme conditions which satisfy this assumption. The requirement of Eq. (6.36) arises from the fact that $x_1(t) = A - \delta$ at sometime within the interval $\overline{n-1}\,T \leq t < nT$, depending on the value of A and $x_1(0)$. However, to include all possible values of A and $x_1(0)$ the extreme condition on t is used, i.e., $t = (n-1)T$. (Note that A in Fig. 6.5 controls the position of the dead zone with respect to the response curve). Solve Eq. (6.36) for A; substitute into Eq. (6.35); and solve the resulting equation for δ

$$\delta > \frac{x_1(\infty) - x_1(\overline{n-1}\,T)}{2} \tag{6.37}$$

After substituting for $x_1(\infty)$ and $x_1(\overline{n-1}\,T)$ in Eq. (6.37) by using Eqs. (6.33) and (6.34) respectively, it is found that

$$\delta > \frac{Kb}{2a}\left[T + \frac{1}{a} - \frac{e^{-a(n-1)T}}{a}\right] \tag{6.38}$$

Note for $a(n-1)T \gg 1$, $e^{-a(n-1)T} \ll 1$ and in this case Eq. (6.38) can be written in the simpler form

$$\delta > \frac{Kb}{2a}\left[T + \frac{1}{a}\right] \tag{6.39}$$

If it is desired to solve for T, Eq. (6.39) can be written as

$$T < \frac{2a\delta}{Kb} - \frac{1}{a} \tag{6.40}$$

An independent verification of Eq. (6.38) will now be demonstrated. Divide Eq. (6.38) through by Kb and recall that in Chapter 5 the definition $\frac{\delta}{Kb} = \delta_n$ was used

$$\delta_n = \frac{\delta}{Kb} > \frac{1}{2a}\left[T + \frac{1}{a} - \frac{e^{-a(n-1)T}}{a}\right] \tag{6.41}$$

Now the values given by Eq. (6.41) are the minimum values of δ_n required to prevent overshoot for that particular value of n. However, a plot of the value of δ_n for the no overshoot case was obtained by an independent method in Chapter 5 and this plot is found in Fig. 5.1. It was shown in Chapter 5 that the peaks in this figure at the separation of the No Overshoot Region from the other regions occur for n = 1,2,3, etc. Now in Fig. 5.1 the following constants are used a = T = 1 second. These values were then substituted into Eq. (6.41) along with first n = 1 then n = 2, etc. The results are shown in Table 6.1.

Table 6.1

Values of δ_n from Eq. (6.41), a = T = 1

n	1	2	3	4	∞
δ_n	$>.5$	$>.816$	$>.932$	$>.975$	>1

Comparing these values with the values of δ_n for the first, second, etc. peaks on Fig. 5.1, it is found that Eq. (6.41) exactly predicts the peaks within the accuracy of the graph. It is thus seen that Eq. (6.41) is a valuable addition to the techniques for producing design graphs already discussed in Chapter 5.

6.3.2 System Time Response

The time response of this system will now be determined. In contrast to the first order system considered in Section 6.2, the output

of the system presently under consideration does not remain constant when the input to the plant goes to zero but continue to increase, at an ever slower rate, until t = ∞. However, as long as Eq. (6.38) is satisfied, the final error of the output from the desired value of A will always be less than δ.

The same definition will be used for the settling time as was used in Section 6.2 and in Chapter 5, i.e., the settling time is the time required for the output to reach 95°/o of its final value. Taking 95°/o of the value calculated from Eq. (6.33) and letting $x_1(0) = 0$, it is found that

$$x_1(t_r) = .95x_1(\infty) = .95\,\frac{KbnT}{a} \tag{6.42}$$

The value of t_r can then be obtained from either Eq. (6.32) or Eq. (6.34) depending on whether $x_1(t_r) = .95x_1(\infty)$ occurs before or after t = nT. For the condition $t_r \leq nT$, substitute Eq. (6.42) into Eq. (6.34) with $x_1(0) = 0$ to obtain

$$.95nT = \left[t_r - \frac{1}{a} + \frac{e^{-at_r}}{a}\right] \tag{6.43}$$

Now assume that it is desired to have $x_1(t)$ equal the desired value, A, in steady state. From Eq. (6.33), with $x_1(0) = 0$,

$$A = x_1(\infty) = \frac{KbnT}{a} \tag{6.44}$$

or

$$nT = \frac{A\,a}{Kb} \tag{6.45}$$

Substitute Eq. (6.45) into Eq. (6.43) and solve for $\frac{Aa^2}{Kb} = y$ to obtain

$$y = 1.053 \left[at_{r_1} - 1 + e^{-at_{r_1}} \right] \tag{6.46}$$

where the subscript 1 in t_{r_1} indicates that t_r is calculated under the condition that $t_r \leq nT$. If $at_{r_1} \gg 1$, Eq. (6.46) may be written in the simpler form

$$y = 1.053 \left[at_{r_1} - 1 \right] \tag{6.47}$$

For the condition $t_r \geq nT$, substitute Eq. (6.42) into Eq. (6.32) to obtain

$$.95nT = nT + \frac{e^{-at_{r_2}}}{a} (1 - e^{anT}) \tag{6.48}$$

Substitute Eq. (6.45) into Eq. (6.48) and solve for $e^{at_{r_2}}$ to obtain

$$e^{at_{r_2}} = \frac{20}{y} (e^y - 1) \tag{6.49}$$

where the subscript 2 in the t_{r_2} indicates that t_r is calculated under the condition that $t_r \geq nT$. The solution for at_{r_2} from Eq. (6.49) is

$$at_{r_2} = \log_e \left[\frac{20}{y} (e^y - 1) \right] \tag{6.50}$$

Eqs. (6.46), (6.47) and (6.50) are plotted in Fig. 6.6. The regions of applicability of these equations within the figure will now be determined. Now Eq. (6.46) is applicable for $t_r \leq nT$ or $at_r \leq anT$, but from Eq. (6.45),

$$at_r \leq \frac{Aa^2}{Kb} = y \tag{6.51}$$

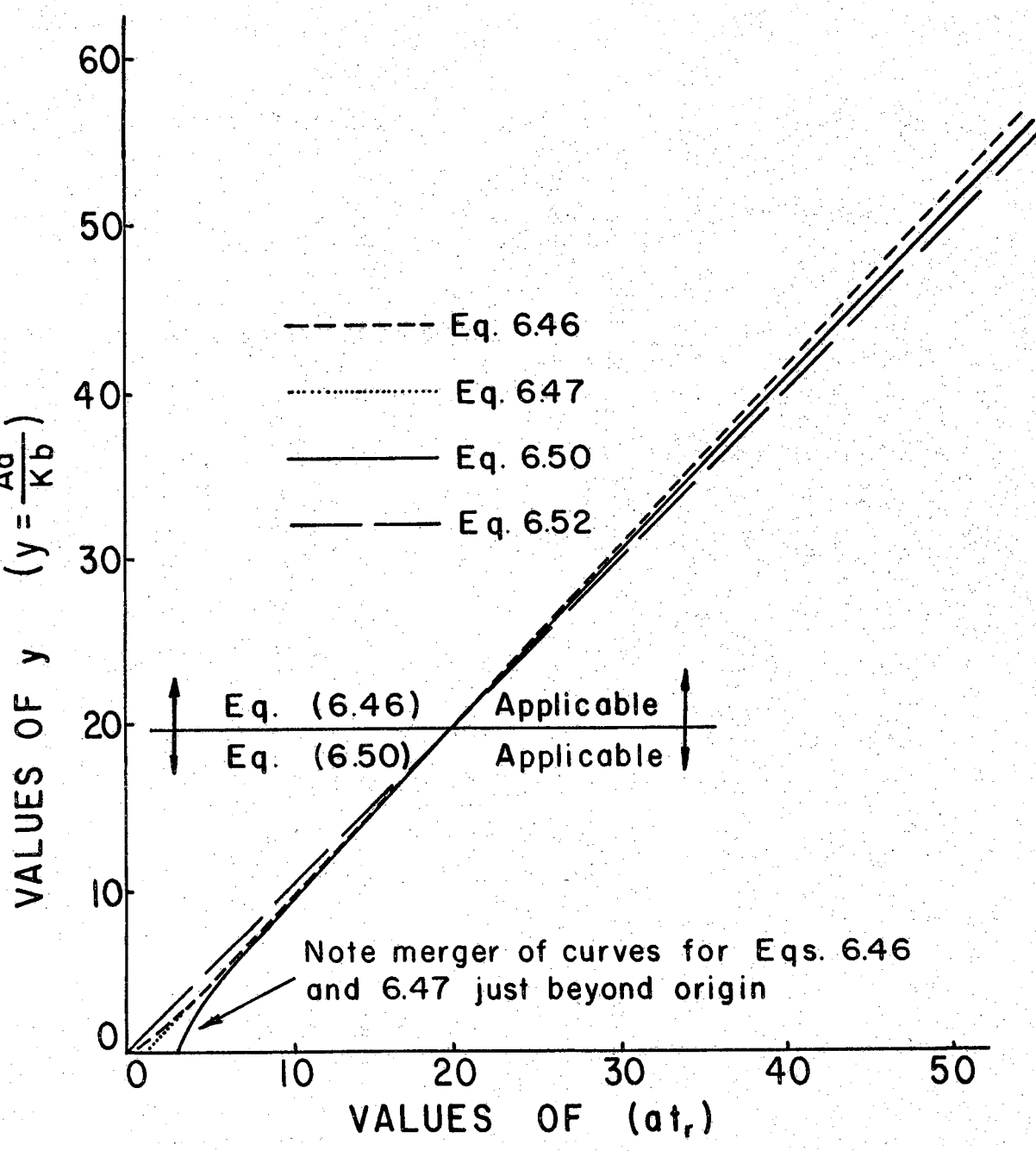


Fig. 6.6. Three Level Quantizer Design Curves
(Closed form solution, second order plant, no overshoot)

Thus, Eq. (6.51) describes the half plane in which Eq. (6.46) is applicable with the equation

$$at_r = y \tag{6.52}$$

being the line of separation. Similarly, $at_r \geq y$ describes the half plane in which Eq. (6.50) is applicable, with Eq. (6.52) again being the line of separation. Equation (6.52) has also been plotted in Fig. 6.6. Note that the intersection of all of the equations occur at $at_r = 19.7$. Thus, Eq. (6.46) is applicable above the intersection and Eq. (6.50) is applicable below the intersection. Further, Fig. 6.6 shows that the dotted line representing Eq. (6.47) is practically identical with the dashed line representing Eq. (6.46) beyond approximately $at_r = 3$. Thus, the simpler equation, Eq. (6.47), can be used over the whole region in which Eq. (6.46) is applicable. In addition, Fig. 6.6 shows that for $8 \leq at_r \leq$ (the intersection) the difference between Eq. (6.47) and Eq. (6.50) is never more than 5°/o, which means that Eq. (6.47) is valid within 5°/o for $at_r \geq 8$. This includes a large portion of the situations occurring in practice, but of course Fig. 6.6 or even Eqs. (6.46) and (6.50) may be used to satisfy more stringent requirements.

Similar to the results found for the first order system in Section 6.2, here it is found that if the system parameters are such that n obtainable from Eq. (6.45) is not an integer, the final value, $x_1(\infty)$, will not equal A but will be larger due to the fact that the system will use the next highest integer as its value of n. This is

the reason for the step-like nature of Fig. 5.2. Thus, the value of t_r from Fig. 6.6 may be somewhat lower than the actual value. However, the difference will be small for values of A requiring large values of n; and a correction can always be made by first determining n from Eq. (6.45), rounding this value to the next highest integer, using Eq. (6.45) to find a new value of A, and finally using Fig. 6.6 to find the corrected value of t_r. On the other hand, the value for t_r obtained from Fig. 6.6 may be larger than the actual t_r if the minimum allowable value for δ, calculated from Eqs. (6.38) or (6.39), is not used. Here again a correction can be made if necessary.

Recalling the definition of the normalized input presented in Chapter 5, i.e., $|r|_n = \frac{|r|}{Kb} = \frac{A}{Kb}$, the quantity $y = \frac{Aa^2}{Kb}$ becomes $y = a^2 |r|_n$. With $a = 1$, $y = |r|_n$ and at $t_r = t_r$. Although they were produced by different methods, Figs. 5.2 and 6.6 are now found to be in agreement. (An explanation for the step nature of Fig. 5.2 has already been made).

A design method using the above results proceeds in the following way:

1. System specifications give the value of a, the maximum value of A, and the maximum allowable value of t_r.
2. Either Fig. 6.6 or one of the equations represented on that figure is used to determine y and from this Kb.
3. The values of K and b may be apportioned according to other requirements or in an arbitrary manner as long as the proper Kb product is maintained.

4. If the value of T has been specified, the value of δ is determined from either Eq. (6.38) or (6.39). Alternatively, if the value of δ has been specified the value of T may be determined from the same equations.

6.3.3 Example and Concluding Remarks

A system designed by the methods of this section will now be compared with a design obtained using the design graphs of Figs. 5.1 and 5.2.

Given: A three level quantizer is to be designed for a system having the block diagram of Fig. 6.4 with K = 2, T = 1 second, a = 1, $t_r \leq 12.2$ seconds, the system error must be less than 2.5 units and the system must operate with an ensemble of inputs up to 22.0 units without overshoot of the final value.

Solution: Here at $t_r = 12.2$ and from Fig. 6.6 $y = 11.6 = \frac{Aa^2}{Kb} = \frac{22}{Kb}$; therefore, Kb = 1.9. From Eq. (6.45), n = 11.6. Since the specification requires $t_r \leq 12.2$ seconds, the new value of n is n = 11. From Eq. (6.45), Kb = 2 and b = 1. Now, y = 11 and from Fig. 6.6, $t_r = 11.6$. Now a(n - 1)T = 10 thus the approximate equation, Eq. (6.39), may be used to find δ; the result is

$$\delta > \frac{2}{2}\left[1 + 1\right] = 2$$

From the specifications on system error and from the Quiescent Plant Rule, δ can be as large as 2.5 units. Therefore, the specifications can be satisfied by choosing the parameters $\delta = 2.0$ and b = 1; the resulting settling time will be 0.7 second less than required by the specification.

From the design graph of Fig. 5.2, it is found that the largest value of t_r which will satisfy the specification is $t_r = 11.7$ seconds and the largest value of $|r|_n$ in this case is $|r|_n = 11.0$. Thus $Kb = \frac{|r|}{|r|_n} = \frac{22}{11.0} = 2$. With $|r|_n = 11$, Fig. 5.1 yields $\delta_n > 1$ as the requirement for no overshoot. From this and the fact that $Kb = 2$, it is found that $\delta > 2$. The close agreement of the results from these two methods should be noted.

It appears to be possible to extend the closed form solutions to more complicated systems, e.g., ramp inputs and higher order plants; but it is likely that the solutions will become much more complicated and if approximations are used they may be less accurate than those already presented.

CHAPTER 7

CONCLUSION

7.1 Summary of Results

The state transition method has been extended to the analysis of digital control systems in which a quantizer is in the error channel and the plant itself is linear and time-invariant. After adapting the method for programming on the digital computer, examples were solved for second and third order systems. Ramp and step inputs, multiple level quantizers, and initial conditions on the state variables other than zero were some of the features of these examples. Where possible these results were compared with the results of other research workers and in other cases they were compared with results from systems simulated on the analog computer. In all cases favorable comparisons were obtained.

A design technique was developed from the analysis method and charts prepared for the design of a second order system containing a relay with dead zone as the quantizer. Using these charts an example of system design to given specifications was completed. Simulation of the system on the analog computer verified the design. Some closed form solutions for first and second order systems subject to step inputs were derived and their results favorably compared with the design techniques mentioned at the beginning of this paragraph.

A set of computational rules were derived. These rules were found to be helpful in both analysis and design by providing information con-

cerning the properties of the systems, by reducing the computational load, by furnishing a check on results, and by providing physical insight into system operation.

7.2 Areas for Future Study

There appear to be a number of possibilities which can be suggested as areas for future study through extension or application of the techniques developed here. One of the most challenging areas for future study is that of the development of closed form solutions. A general method is desirable but it is doubtful whether one will be forthcoming in the near future. On the other hand, the method presented here can probably be extended and there is a possibility that a closed form method can be developed based on the state transition technique.

Perhaps a less challenging area but certainly a promising one is that of extending the methods developed here, especially that of digital simulation, to digital and other nonlinear sampled-data control systems of other forms. Typical examples of this are systems containing quantizers in both the feedback path as well as in the error channel and systems in which a nonlinear element appears between two frequency sensitive elements.

There are a number of possible paths to be investigated which may be called application of the present results. Such things as consideration of other types of nonlinearities, writing and studying the results of more general computer programs, and developing design graphs for additional plants fall into this category.

7.3 Conclusions

Digital simulation, in conjunction with the computational rules, has been shown to be a powerful and versatile method for the analysis and design of digital control systems. In contrast to many other methods it is not limited by input type, order of the plant, state variables having other than zero initial conditions, and quantizer complexity. Not only does digital simulation possess the distinct advantages of accuracy and noise free performance over analog computer simulation, but in most cases it also is faster, more versatile and easier to use. On the other hand, simulation on the analog computer tends to complement digital simulation in that it presents more of the practical problems of control system operation and is a good means of spot checking digital computer results.

Of course closed form solutions are more concise and much quicker to apply than the other design and analysis methods considered. However, they are difficult to develop and presently are available for only a few cases.

From the work on quantizers containing a relay with a dead zone, it appears that this type of quantizer can be used in a large variety of systems having step inputs to satisfy specifications on static accuracy, response time, and on absence of overshoot and limit cycle oscillations. However, there will be cases where a compromise must be made between fast response time and high static accuracy in that these two quantities are somewhat opposed to each other.

BIBLIOGRAPHY

1. Tou, J. T., Digital and Sampled-data Control Systems, McGraw-Hill Book Company, Inc., New York, N. Y., 1959.

2. Ragazzini, J. R. and Franklin, G. F., Sampled-data Control Systems, McGraw-Hill Book Company, Inc., New York, N. Y., 1958.

3. Jury, E. I., Sampled-data Control Systems, John Wiley and Sons, Inc., New York, N. Y., 1958.

4. Jury, E. I., "Recent Advances in the Field of Sampled-data and Digital Control Systems," IFAC Proceedings, Vol. I, pp. 262-269, Butterworths, London, England, 1961.

5. Chow, C. K., "Contactor Servomechanisms Employing Sampled-Data," AIEE Transactions, Vol. 73, pt. II, pp. 51-62, March, 1954.

6. Russel, F. A., Discussion of Ref. 5; AIEE Transactions, Vol. 73, pt. II, p. 62, March, 1954.

7. Kuo, B. C., "A Z-Transform-Describing Function for On-Off Type Sampled-Data Systems," Proc. IRE, Vol. 48, pp. 941-942, May, 1960.

8. Kuo, B. C., "Nonlinear Sampled-data Control Systems," University of Illinois Ph.D. Thesis, 1958.

9. Kalman, R. E., "Nonlinear Aspects of Sampled-Data Control Systems," Proc. Symposium on Nonlinear Circuit Analysis, pp. 273-313, Polytechnic Institute of Brooklyn, N. Y., 1956.

10. Izawa, K., "Discontinuous Feedback Control Systems with Sampling Action," IFAC Proceedings, Vol. I, pp. 321-327, Butterworths, London, England, 1961.

11. Scheidenhelm, R., "The Analysis and Design of Digitally-Controlled Instrument Servos," Technical Report 7890-TR-1, MIT Servomechanicms Laboratory, June, 1959.

12. Izawa, K., and Weaver, L. A., "Relay-Type Feedback Control Systems with Dead Time and Sampling," AIEE Transactions, Vol. 78, pt. II, pp. 49-53, May, 1959.

13. Mullin, F. J., and Jury, E. I., "Phase-Plane Approach to Relay Sampled-Data Systems," AIEE Transactions, Vol. 77, pt. II, pp. 517-524, January, 1959.

14. Aseltine, J. A., "Non-linear Sampled-data System Analysis by the Incremental Phase-Plane Method," IFAC Proceedings, Vol. I, pp. 295-304, Butterworths, London, England, 1961.

15. Tostanoski, B. M., "The Analysis of Sampled-Data Servomechanisms Performed on the IBM Type 650," AIEE Transactions, Vol. 75, pt. I, pp. 446-450, September, 1956.

16. Kinnen, E. and Tou, J., "Analysis of Nonlinear Sampled-Data Control Systems, Part I," AIEE Transactions, Vol. 78, pt. II, pp. 386-390, January, 1960.

17. Kinnen, E. and Tou, J., "Analysis of Nonlinear Sampled-Data Control Systems, Part II," AIEE Transactions, Vol. 78, pt. II, pp. 390-394, January, 1960.

18. Steel, G. K., "Analysis of Nonlinear Sampled-Data Control Systems by a Method of Linear Response Correction," Journal of Electronics and Control, Vol. IX, pp. 309-320, October, 1960.

19. Chestnut, H., Dabul, A., and Leiby, D., "Analog Computer Study of Sampled-Data Systems," AIEE Transactions, Vol. 77, pt. II, pp. 634-640, January, 1959.

20. Klein, R. C., "Analog Simulation of Sampled-Data Systems," IRE Transactions of Telemetry and Remote Control, Vol. TRC-1, pp. 2-7, May, 1955.

21. Wadel, L. B., "Analysis of Combined Sampled and Continuous Data Systems on an Electronic Analog Computer," IRE Convention Record, pt. 4, pp. 3-7, 1955.

22. Scheidenhelm, R. and Yngvar, L., "A Simulator Study of Two Digitally-Controlled Instrument Servos," Technical Report 7890-TR-2, MIT Servomechanisms Laboratory, June, 1959.

23. Bertram, J. E., "The Effect of Quantization in Sampled Feedback Systems," AIEE Transactions, Vol. 77, pt. II, pp. 177-182, September, 1958.

24. Tsypkin, Y. Z., "Estimating the Effect of Quantization by Level on the Processes in Automatic Digital Control Systems," Automation and Remote Control, Vol. 21, pp. 195-197, November, 1960.

25. Torng, H. C., and Meserve, W. E., "Determination of Periodic Modes in Relay Servomechanisms Employing Sampled Data," IRE Transactions, Vol. AC-5, pp. 298-305, September, 1960.

26. Tou, J. T., and Lewis, J. B., "A Study of Nonlinear Digital Control Systems," Technical Report 201, Purdue University School of Electrical Engineering, July, 1961.

27. Widrow, B., "Statistical Analysis of Amplitude-Quantized Sampled-Data Systems," AIEE Transactions, Vol. 79, pt. II, pp. 555-568, January, 1961.

28. Tou, J. T., and Vadhanaphut, B., "Optimum Control of Nonlinear Discrete-Data Systems," AIEE Transactions, Vol. 80, pt. II, pp. 166-171, September, 1961.

29. Kalman, R. E., and Bertram, J. E., "A Unified Approach to the Theory of Sampling Systems," Jour. Franklin Inst., Vol. 267, pp. 405-436, May, 1959.

30. Gilbert, E. O., "A Method for the Symbolic Representation and Analysis of Linear Periodic Feedback Systems," AIEE Transactions, Vol. 78, pp. 512-523, January, 1960.

31. Susskind, A. K., "Notes on Analog-Digital Conversion Techniques," John Wiley and Sons, Inc., New York, N. Y., 1957.

APPENDIX A

PROGRAM FOR THE RPC 4000 COMPUTER

The following is a program that computes the state vector at the sampling points for the plant $\frac{K}{s(s + a)}$ and a system input which is a step.* The quantizer can contain as many levels as desired but the magnitude of the output levels and the quantization intervals must be the same for both positive and negative signals. (This is a working program and has not been reduced to the minimum possible orders). The coding is that of the PINT routine, a detailed explanation of which may be obtained from the School of Electrical Engineering, Purdue University.

Input Data Storage Locations:

194 = a, 195 = K, 196 = T, 197 = number of positive levels in the quantizer, 198 = 1.0, 199 = number of points to be computed, 200 = order of the $\phi(T)$ matrix, 501 = δ_1, 502 = $\delta_2 - \delta_1$, 503 = $\delta_3 - \delta_2$, ..., 600 = 0., 601 = b_1, 602 = b_2, ..., 899 = r(0), 900 = $x_1(0)$, 902 = $x_2(0)$, 903 = $x_3(0)$, ... (For an explanation of the above symbols, refer to the List of Symbols).

Summary of the Function of Program Orders:**

023-026	data input
027-048	computation of ϕ matrix entries

---

*Results for more complicated plants and system inputs can be obtained by modifications to this program.

**Also see Fig. 4.1, which is a basic flow diagram for the program.

049-070	preparing conditions for both matrix multiplication and for quantizer subroutine
071-073	compute and store e(kT+)
074-087	quantizer subroutine
088-090	preparation for transfer of state vector (see order 101)
091-098	subroutine for printing data
099	jump to location 134 for next instruction
100	jump to location 112 for next instruction
101-103	transfer of state vector from temporary storage to operating position
104-111	end computation, input new data, and begin new computation
112-132	matrix multiplication subroutine
133	temporary storage
134-138	calling sequence for matrix multiplication

Program Proper:

CLEAR* LOAD 023*

023	INM194*	INM501*	INM600*	INM899*	CCF198*
028	CCI201*	CCI205*	CZI204*	CZI206*	CZI207*
033	CCF196*	MUL194*	NEG000*	EXP000*	CCI209*
038	NEG000*	ADD198*	DIV194*	CCI203*	MUL195*
043	CCI208*	CCF196*	SUB203*	MUL195*	DIV194*
048	CCI202*	7LDA201*	6LDA399*	4LDI001*	3LDI001*
053	2LDI001*	1LDI001*	2LDA000*	CCF199*	CCI397*
058	CCF200*	ADD198*	CAI061*	2LDC004*	2CCF899*
063	2CCI399*	2CIJ062*	1LDA000*	3LDA000*	4LDA000*
068	CCF197*	CAI070*	1LDC001*	CCF399*	SUB400*
073	CCI398*	JIN081*	1SUB501*	JIN078*	1CIJ075*
078	1CCF600*	CCI401*	JMP088*	POS000*	1SUB501*

```
083      JIN085*    1CIJ082*    1CCF600*     NEG000*     CCI401*
088      CCF200*     CAI090*    3LDC003*     CCF200*     ADD198*
093      ADD198*     CAI095*    4LDC005*    4PRM398*    4CIJ096*
098      CAR000*     JMP134*     JMP112*    3CCF800*    3CCI400*
103    3CIJ101*      CCF397*     SUB198*     CCI397*     JOS109*
108      JIP065*     HLT110*     INM899*     JMP055*     SRA132*
113      CAI119*     CAI123*     CAI130*    6CXI125*    X5LDI001*
118   X6LDI001*     5LDC003*     CZI133*    7CXI124*    X6LDA000*
123    6LDC003*     6CCF207*    6MUL400*     ADD133*     CCI133*
128    6CIJ124*    X5CCI000*    7AXA003*    5CIJ120*     JMP101*
133 Temp. Storage   CCF200*     7LDA201*    6LDA400*    5LDA800*
138      JMP100*
```

```
BEGIN023*
```

APPENDIX B

SYSTEM SIMULATION BY MEANS OF THE ANALOG COMPUTER

The reasons for making analog computer runs, their advantages and disadvantages, and a summary of the results obtained have already been presented in Chapter 4. However, the actual technical details of the simulation equipment have not been previously covered and will be discussed here.

A simplified block diagram of the overall experimental equipment was presented in Fig. 4.4. The experimental equipment is built around the Epsco Model B-611 analog to digital converter and a digital to analog converter. The Epsco is constructed so that a semistatic binary coding of the analog input signal, at the sampling instant, is available within a few microseconds of the sampling instant and lasting until the next sampling instant. Moreover, a parallel output is available from the Epsco so that each binary digit is represented by a separate output terminal. The digital to analog converter, basically a decoder, acts on the semistatic output of the Epsco to provide an analog output. For this purpose a relay type and an electronic type decoder were designed and tested.

In the relay type decoder each bit in the parallel output from the Epsco controls a separate relay in the decoder. The relays in turn control the resistance in the feedback path of an operational amplifier so that the voltage output from the decoder is a quantized analog representation of the original input to the Epsco. Such decoders are

described in detail in References 1 and 31. Although the relay type decoder is rather simple in construction, it was found to be unsatisfactory in that a ramp input to the Epsco was not found to give a perfect "stair case type" output. Instead, the output had spikes riding on the expected wave form at several points. The trouble was traced to the fact that at several times during the duration of the ramp input some of the relays in the decoder are required to open simultaneously with the closing of others. Since the relays do not have the same pull-in and drop-out times, an erroneous relay combination exists for a short time causing the voltage spikes.

The circuit for the electronic type decoder is shown in Fig. B.1. It is connected to the Epsco in the same way as the relay type, but in the electronic type the Epsco's parallel outputs control vacuum tubes, which are connected so as to approximate constant current generators. The vacuum tubes in turn are connected to a resistance network* which combines the individual signals in a weighted fashion according to the significance of the bit which they represent. The output from this network is a quantized analog representation of the original continuous signal which was impressed on the Epsco. The single pole double throw switches in the grid circuits of all channels except the first determine the number of decoder channels receiving inputs from

---

*The resistors in the network which connect the plates of the tubes directly to the +200 v. supply are 6.8k except those of the end channels. All other resistors in the network are 3.3k.

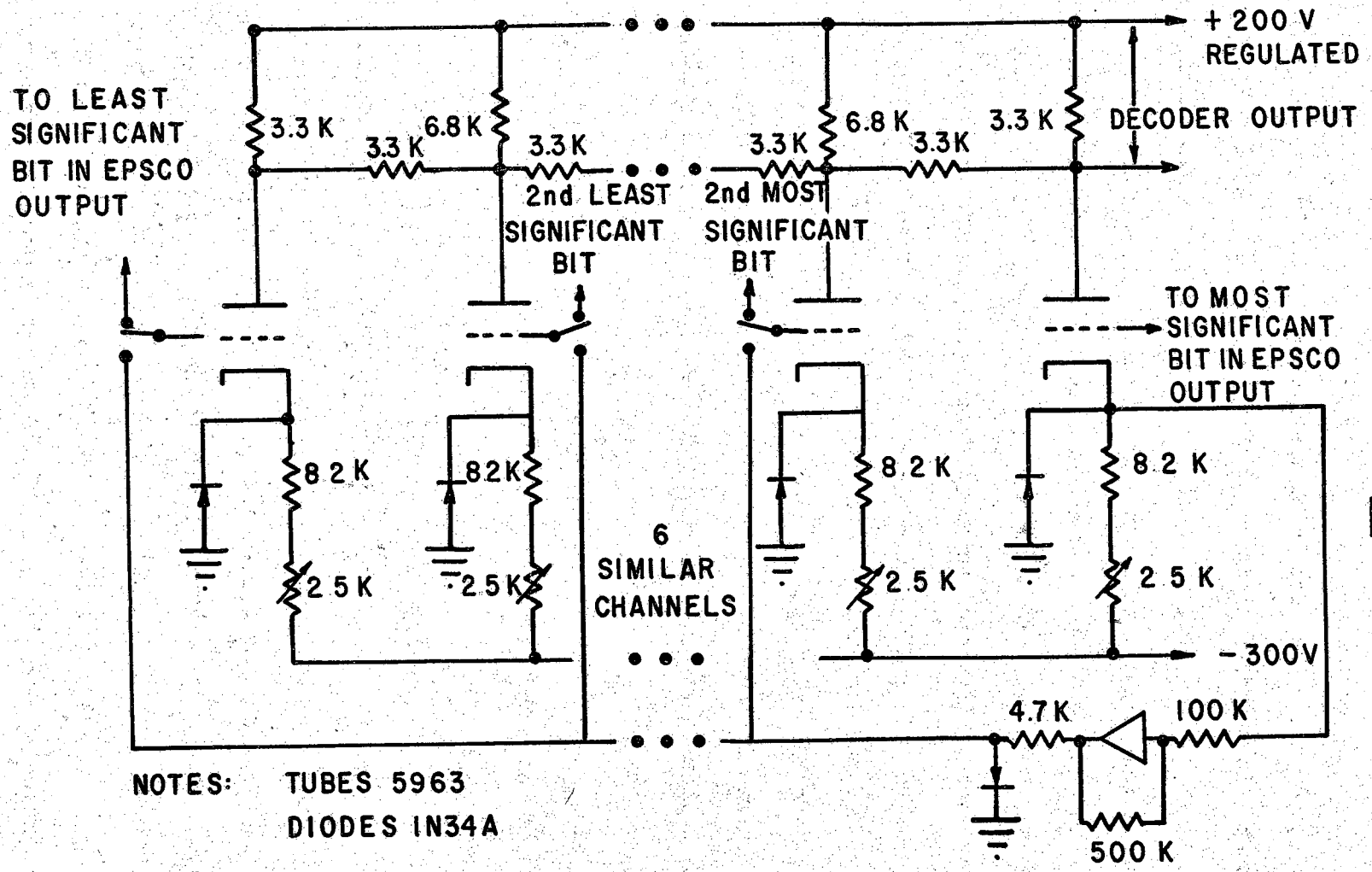


Fig. B-1. Circuit Diagram of the Electronic Type Decoder

the Epsco; hence they determine the number of levels in the decoder output. In the down position of these switches, instead of receiving signal from the Epsco, the grids receive an amplified signal from the channel representing the most significant bit, the sign bit, causing the channels whose switches are down to be in the opposite state as the sign bit. This forces the first quantization interval, δ_1, to be of the same width for both negative and positive inputs. Without this circuit the first quantization interval would be some nonzero value to inputs of one sign and zero to inputs of the other sign.

Like the relay type decoder, the electronic decoder is also rather simple in construction, yet open loop tests as well as later closed loop tests have revealed a large improvement in performance for it over the relay type decoder. Because of its superior performance, the electronic type decoder has been used in all the closed loop system simulations. Additional information concerning his decoder is found in Reference 31.

A detailed analog simulation flow diagram for a typical computer run is shown in Fig. B.2 with the function of each part of the circuit being indicated adjacent to that part. Note that Amplifier 3 performs the dual function of forming the error signal as well as providing an adjustment on the overall scale of the quantization intervals. One way of looking at this is that with a gain of unity in Amplifier 3, δ_1, δ_2, etc. occur at voltages fixed by the Epsco, but when Amplifier 3 possesses the gain K each of the δ's possessed by the Epsco alone are divided by K which effectively produces a new scale for the quantizer.

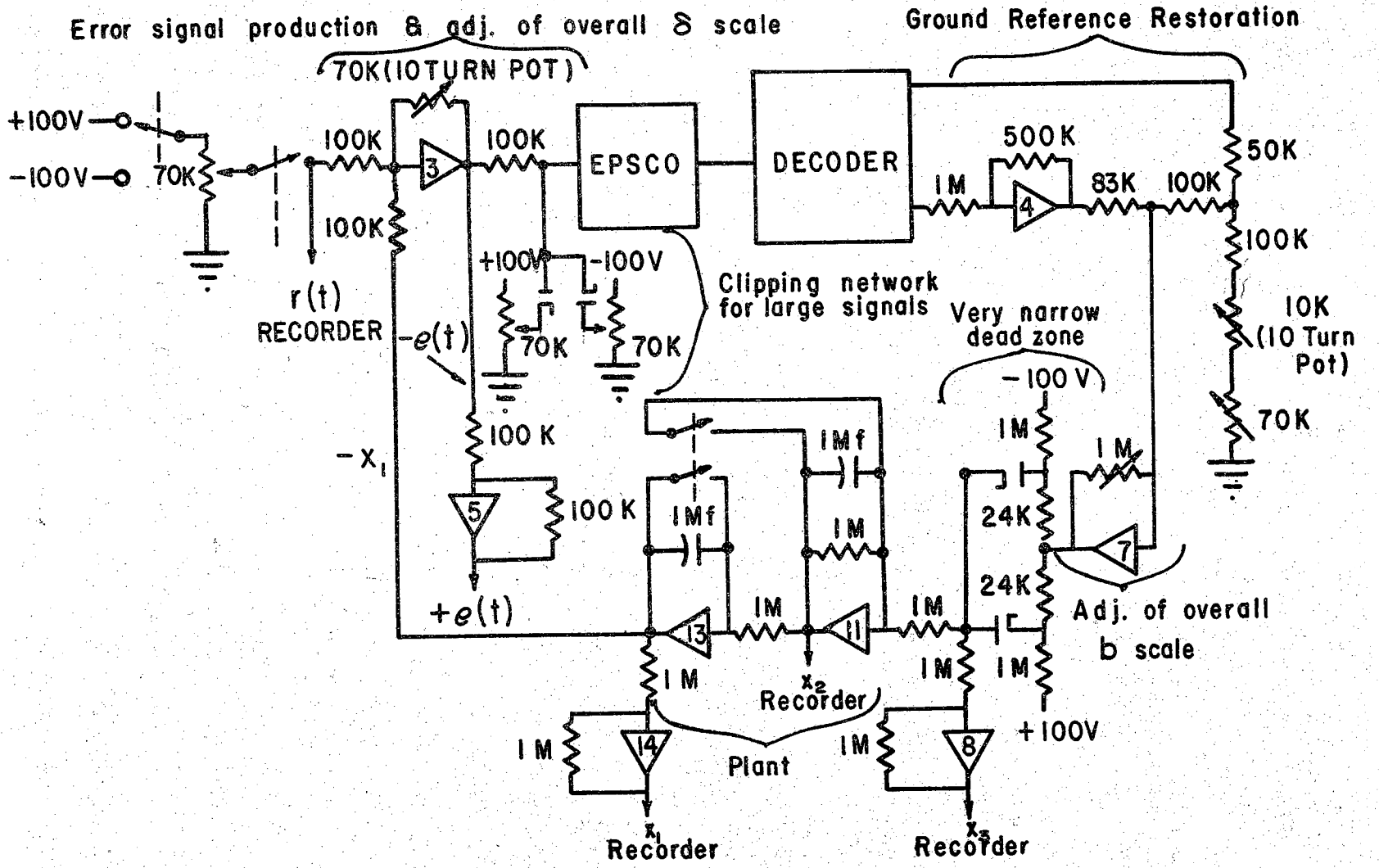


Fig. B-2. Analog Simulation Flow Diagram

(Since the output to the e(t) recorder channel follows Amplifier 3, the gain K must be taken into account in the calibration of this channel so that the true value of e(t) will be recorded). The circuitry associated with Amplifier 4 and the input to Amplifier 7 provides a d.c. level change (ground reference restoration) so that the output from the decoder, normally biased at 145 volts above ground, is referenced to ground potential, which avoids grounding problems and a shock hazard in other parts of the system. The 70K potentiometer provides a course and the 10K, 10 turn potentiometer provides a fine adjustment on the ground reference. The feedback resistor around Amplifier 7 is adjustable to establish the overall scale on the b values of the quantizer much as is done with Amplifier 3 to establish the scale on the δ's.

It was found to be necessary to provide a clipping network between the output of Amplifier 3 and the Epsco in order to prevent voltages considerably greater than 10 volts, the Epsco's full scale input, from reaching it in those cases where the initial error signal is very large. Without the clipping circuit the Epsco was found to give erroneous results for the first few sampling periods follow the receipt of a signal of magnitude much greater than 10 volts. A circuit producing a dead zone whose magnitude is much less than the dead zone of the quantizer is inserted between Amplifiers 7 and 11. This circuit prevents any small deviations from zero output of Amplifier 7 from being fed to the plant where it would be integrated producing a drift in the output. Account is taken of the additional dead zone in establishing the b values of the quantizer.