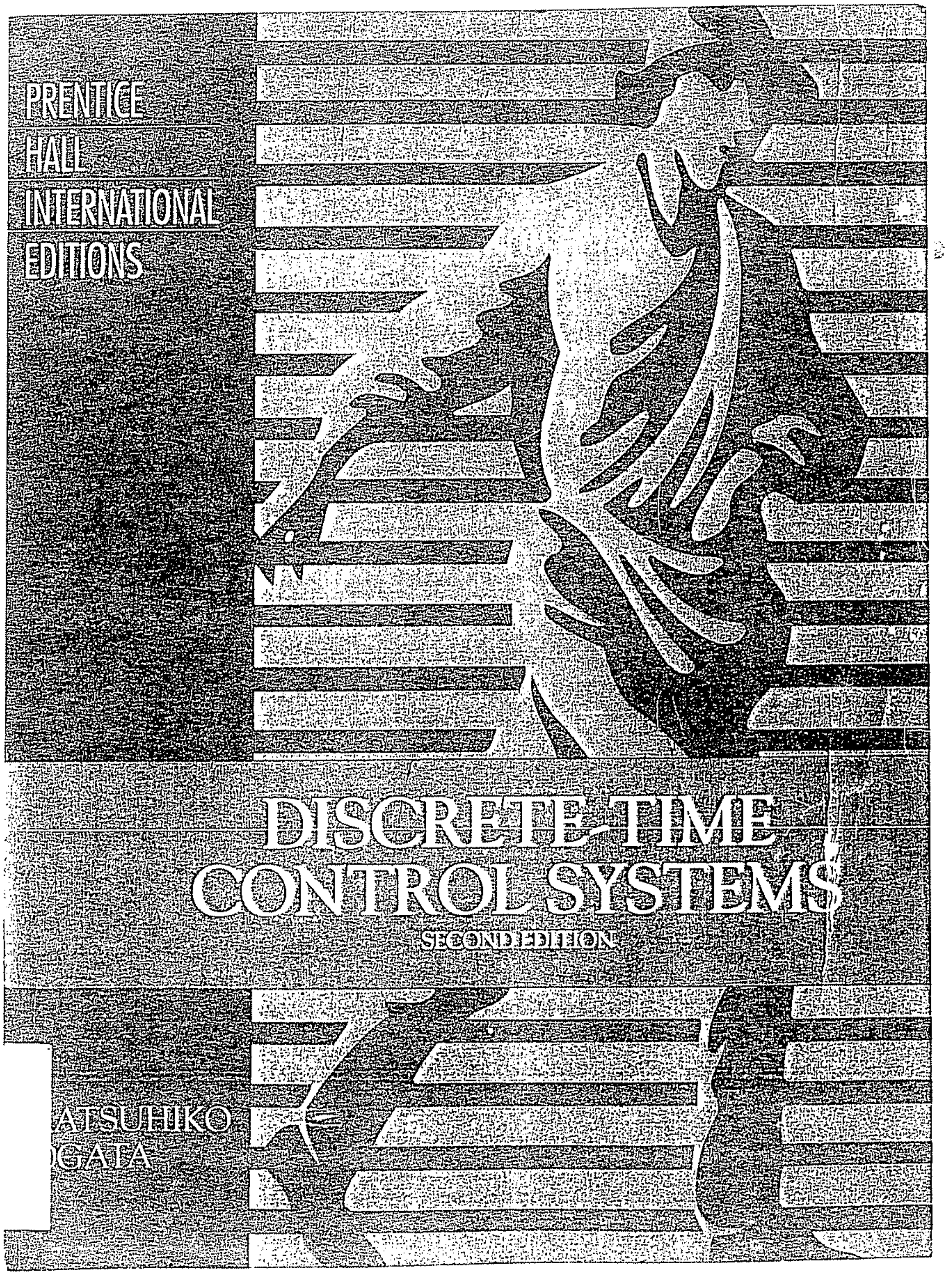
INTERNATIONAL EDITION

Second Edition

Discrete-Time Control Systems

Katsuhiko Ogata

PRENTICE
HALL
INTERNATIONAL
EDITIONS



DISCRETE-TIME CONTROL SYSTEMS

SECOND EDITION

ATSUHIKO
GATA

DISCRETE-TIME CONTROL SYSTEMS

Şerefhan Engin

DISCRETE-TIME CONTROL SYSTEMS

Second Edition

Katsuhiko Ogata
University of Minnesota

Prentice-Hall International, Inc.

A Simon & Schuster Company
Englewood Cliffs, New Jersey 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-328642-8

PRENTICE-HALL INTERNATIONAL (UK) LIMITED *London*
PRENTICE-HALL OF AUSTRALIA PTY LIMITED *Sydney*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL HISPANOAMERICANA S.A., *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED *New Delhi*
PRENTICE-HALL OF JAPAN, INC *Tokyo*
SIMON & SCHUSTER ASIA PTE LTD *Singapore*
EDITORA PRENTICE-HALL DO BRASIL LTDA *Rio de Janeiro*
PRENTICE-HALL, *Englewood Cliffs, New Jersey*

Contents

Chapter 3

z-Plane Analysis of Discrete-Time Control Systems 74

Chapter 4

Design of Discrete-Time Control Systems by Conventional Methods 173

Midterm-1

Chapter 5

State-Space Analysis 293

42p

state-space

Chapter 6

Pole Placement and Observer Design 377

100p

intensive
state-space

10-11-12

Midterm ?

interactive
state-space

Appendix B

z Transform Theory 681

Appendix C

Pole Placement Design with Vector Control 704

Preface

This book presents a comprehensive treatment of the analysis and design of discrete-time control systems. It is written as a textbook for courses on discrete-time control systems or digital control systems for senior and first-year graduate level engineering students.

In this second edition, some of the older material has been deleted and new material has been added throughout the book. The most significant feature of this edition is a greatly expanded treatment of the pole-placement design with minimum-order observer by means of the state-space approach (Chapter 6) and the polynomial-equations approach (Chapter 7).

In this book all materials are presented in such a way that the reader can follow the discussions easily. All materials necessary for understanding the subject matter presented (such as proofs of theorems and steps for deriving important equations for pole placement and observer design) are included to ease understanding of the subject matter presented.

The theoretical background materials for designing control systems are discussed in detail. Once the theoretical aspects are understood, the reader can use MATLAB with advantage to obtain numerical solutions that involve various types of vector-matrix operations. It is assumed that the reader is familiar with the material presented in my book *Solving Control Engineering Problems with MATLAB* (Prentice Hall) or its equivalent.

The prerequisites for the reader are a course on introductory control systems, a course on ordinary differential equations, and familiarity with MATLAB computations. (If the reader is not familiar with MATLAB, it may be studied concurrently.)

Since this book is written from the engineer's point of view, the basic concepts involved are emphasized and highly mathematical arguments are carefully avoided in the presentation. The entire text has been organized toward a gradual development of discrete-time control theory.

The text is organized into eight chapters and three appendixes. The outline of the book is as follows: Chapter 1 gives an introduction to discrete-time control systems. Chapter 2 presents the z transform theory necessary for the study of discrete-time control systems. Chapter 3 discusses the z plane analysis of discrete-time systems, including impulse sampling, data hold, sampling theorem, pulse transfer function, and digital filters. Chapter 4 treats the design of discrete-time control systems by conventional methods. This chapter includes stability analysis of closed-loop systems in the z plane, transient and steady-state response analyses, and design based on the root-locus method, frequency-response method, and analytical method.

Chapter 5 presents state-space analysis, including state-space representations of discrete-time systems, pulse transfer function matrix, discretization method, and Liapunov stability analysis. Chapter 6 discusses pole-placement and observer design. This chapter contains discussions on controllability, observability, pole placement, state observers, and servo systems. Chapter 7 treats the polynomial equations approach to control systems design. This chapter first discusses the Diophantine equation and then presents the polynomial equations approach to control systems design. Finally, model matching control systems are designed using the polynomial equations approach. Chapter 8 presents quadratic optimal control. Both finite-stage and infinite-stage quadratic optimal control problems are discussed. This chapter concludes with a design problem based on quadratic optimal control solved with MATLAB.

Appendix A presents a summary of vector-matrix analysis. Appendix B gives useful theorems of the z transform theory that were not presented in Chapter 2, the inversion integral method, and the modified z transform method. Appendix C discusses the pole-placement design problem when the control signal is a vector quantity.

Examples are presented at strategic points throughout the book so that the reader will have a better understanding of the subject matter discussed. In addition, a number of solved problems (A problems) are provided at the end of each chapter, except Chapter 1. These problems represent an integral part of the text. It is suggested that the reader study all these problems carefully to obtain a deeper understanding of the topics discussed. In addition, many unsolved problems (B problems) are provided for use as homework or quiz problems.

Most of the materials presented in this book have been class-tested in senior and first-year graduate level courses on control systems at the University of Minnesota.

All the materials in this book may be covered in two quarters. In a semester course, the instructor will have some flexibility in choosing the subjects to be covered. In a quarter course, a good part of the first six chapters may be covered. An instructor using this text can obtain a complete solutions manual from the publisher. This book can also serve as a self-study book for practicing engineers who wish to study discrete-time control theory by themselves.

Appreciation is due to my former students who solved all the solved problems (A problems) and unsolved problems (B problems) and made numerous constructive comments about the material in this book.

Katsuhiko Ogata

DISCRETE-TIME
CONTROL
SYSTEMS

1

Introduction to Discrete-Time Control Systems

1-1 INTRODUCTION

In recent years there has been a rapid increase in the use of digital controllers in control systems. Digital controls are used for achieving optimal performance—for example, in the form of maximum productivity, maximum profit, minimum cost, or minimum energy use.

Most recently, the application of computer control has made possible "intelligent" motion in industrial robots, the optimization of fuel economy in automobiles, and refinements in the operation of household appliances and machines such as microwave ovens and sewing machines, among others. Decision-making capability and flexibility in the control program are major advantages of digital control systems.

The current trend toward digital rather than analog control of dynamic systems is mainly due to the availability of low-cost digital computers and the advantages found in working with digital signals rather than continuous-time signals.

Types of Signals. A continuous-time signal is a signal defined over a continuous range of time. The amplitude may assume a continuous range of values or may assume only a finite number of distinct values. The process of representing a variable by a set of distinct values is called *quantization*, and the resulting distinct values are called *quantized* values. The quantized variable changes only by a set of distinct steps.

An analog signal is a signal defined over a continuous range of time whose amplitude can assume a continuous range of values. Figure 1–1(a) shows a continuous-time analog signal, and Figure 1–1(b) shows a continuous-time quantized signal (quantized in amplitude only).

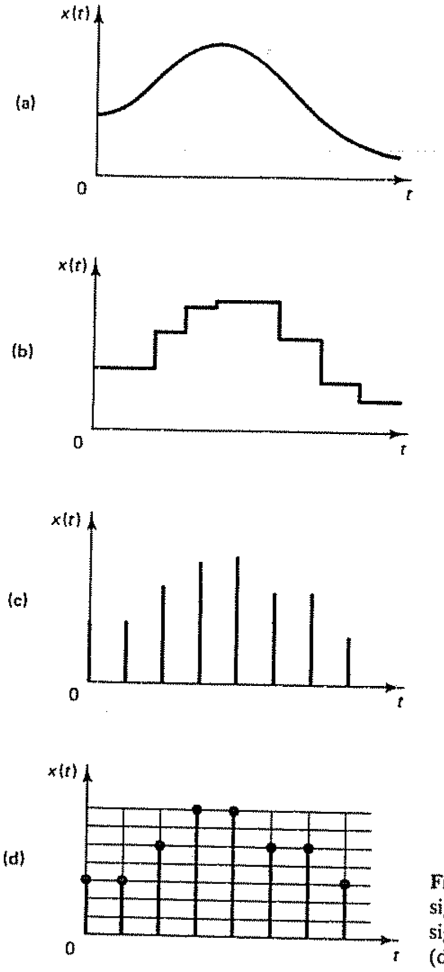


Figure 1–1 (a) Continuous-time analog signal; (b) continuous-time quantized signal; (c) sampled-data signal; (d) digital signal. (numerically coded)

Notice that the analog signal is a special case of the continuous-time signal. In practice, however, we frequently use the terminology "continuous-time" in lieu of "analog." Thus in the literature, including this book, the terms "continuous-time signal" and "analog signal" are frequently interchanged, although strictly speaking they are not quite synonymous.

A discrete-time signal is a signal defined only at discrete instants of time (that is, one in which the independent variable t is quantized). In a discrete-time signal, if the amplitude can assume a continuous range of values, then the signal is called a *sampled-data signal*. A sampled-data signal can be generated by sampling an analog signal at discrete instants of time. It is an amplitude-modulated pulse signal. Figure 1–1(c) shows a sampled-data signal.

A digital signal is a discrete-time signal with quantized amplitude. Such a signal can be represented by a sequence of numbers, for example, in the form of binary numbers. (In practice, many digital signals are obtained by sampling analog signals and then quantizing them; it is the quantization that allows these analog signals to be read as finite binary words.) Figure 1–1(d) depicts a digital signal. Clearly, it is a signal quantized both in amplitude and in time. The use of the digital controller requires quantization of signals both in amplitude and in time.

The term "discrete-time signal" is broader than the term "digital signal" or the term "sampled-data signal." In fact, a discrete-time signal can refer either to a digital signal or to a sampled-data signal. In practical usage, the terms "discrete time" and "digital" are often interchanged. However, the term "discrete time" is frequently used in theoretical study, while the term "digital" is used in connection with hardware or software realizations.

In control engineering, the controlled object is a plant or process. It may be a physical plant or process or a nonphysical process such as an economic process. Most plants and processes involve continuous-time signals; therefore, if digital controllers are involved in the control systems, signal conversions (analog to digital and digital to analog) become necessary. Standard techniques are available for such signal conversions; we shall discuss them in Section 1–4.

Loosely speaking, terminologies such as discrete-time control systems, sampled-data control systems, and digital control systems imply the same type or very similar types of control systems. Precisely speaking, there are, of course, differences in these systems. For example, in a sampled-data control system both continuous-time and discrete-time signals exist in the system; the discrete-time signals are amplitude-modulated pulse signals. Digital control systems may include both continuous-time and discrete-time signals; here, the latter are in a numerically coded form. Both sampled-data control systems and digital control systems are discrete-time control systems.

Many industrial control systems include continuous-time signals, sampled-data signals, and digital signals. Therefore, in this book we use the term "discrete-time control systems" to describe the control systems that include some forms of sampled-data signals (amplitude-modulated pulse signals) and/or digital signals (signals in numerically coded form).

Systems Dealt With in This Book. The discrete-time control systems considered in this book are mostly linear and time invariant, although nonlinear and/or time-varying systems are occasionally included in discussions. A linear system is one in which the principle of superposition applies. Thus, if y_1 is the response of the system to input x_1 and y_2 the response to input x_2, then the system is linear if and only if, for every scalar α and β, the response to input $\alpha x_1 + \beta x_2$ is $\alpha y_1 + \beta y_2$.

A linear system may be described by linear differential or linear difference equations. A time-invariant linear system is one in which the coefficients in the differential equation or difference equation do not vary with time, that is, one in which the properties of the system do not change with time.

Discrete-Time Control Systems and Continuous-Time Control Systems. Discrete-time control systems are control systems in which one or more variables can change only at discrete instants of time. These instants, which we shall denote by kT or t_k $(k = 0, 1, 2, \ldots)$, may specify the times at which some physical measurement is performed or the times at which the memory of a digital computer is read out. The time interval between two discrete instants is taken to be sufficiently short that the data for the time between them can be approximated by simple interpolation.

Discrete-time control systems differ from continuous-time control systems in that signals for a discrete-time control system are in sampled-data form or in digital form. If a digital computer is involved in a control system as a digital controller, any sampled data must be converted into digital data.

Continuous-time systems, whose signals are continuous in time, may be described by differential equations. Discrete-time systems, which involve sampled-data signals or digital signals and possibly continuous-time signals as well, may be described by difference equations after the appropriate discretization of continuous-time signals.

Sampling Processes. The sampling of a continuous-time signal replaces the original continuous-time signal by a sequence of values at discrete time points. A sampling process is used whenever a control system involves a digital controller, since a sampling operation and quantization are necessary to enter data into such a controller. Also, a sampling process occurs whenever measurements necessary for control are obtained in an intermittent fashion. For example, in a radar tracking system, as the radar antenna rotates, information about azimuth and elevation is obtained once for each revolution of the antenna. Thus, the scanning operation of the radar produces sampled data. In another example, a sampling process is needed whenever a large-scale controller or computer is time-shared by several plants in order to save cost. Then a control signal is sent out to each plant only periodically and thus the signal becomes a sampled-data signal.

The sampling process is usually followed by a quantization process. In the quantization process the sampled analog amplitude is replaced by a digital amplitude (represented by a binary number). Then the digital signal is processed by the computer. The output of the computer is sampled and fed to a hold circuit. The output of the hold circuit is a continuous-time signal and is fed to the actuator. We shall present details of such signal-processing methods in the digital controller in Section 1–4.

The term "discretization," rather than "sampling," is frequently used in the analysis of multiple-input–multiple-output systems, although both mean basically the same thing.

It is important to note that occasionally the sampling operation or discretization is entirely fictitious and has been introduced only to simplify the analysis of control systems that actually contain only continuous-time signals. In fact, we often use a suitable discrete-time model for a continuous-time system. An example is a digital-computer simulation of a continuous-time system. Such a digital-computer-simulated system can be analyzed to yield parameters that will optimize a given performance index.

Most of the material presented in this book deals with control systems that can be modeled as linear time-invariant discrete-time systems. It is important to mention that many digital control systems are based on continuous-time design techniques. Since a wealth of experience has been accumulated in the design of continuous-time

High speed, precision Sample-and-Hold Amplifier : AD585
8-bit DAC : AD558
8-bit CMOS DAC : AD7524

controllers, a thorough knowledge of them is highly valuable in designing discrete-time control systems.

1–2 DIGITAL CONTROL SYSTEMS

Figure 1–2 depicts a block diagram of a digital control system showing a configuration of the basic control scheme. The system includes the feedback control and the feedforward control. In designing such a control system, it should be noted that the "goodness" of the control system depends on individual circumstances. We need to choose an appropriate performance index for a given case and design a controller so as to optimize the chosen performance index.

Signal Forms in a Digital Control System. Figure 1–3 shows a block diagram of a digital control system. The basic elements of the system are shown by the blocks. The controller operation is controlled by the clock. In such a digital control system, some points of the system pass signals of varying amplitude in either continuous time or discrete time, while other points pass signals in numerical code, as depicted in the figure.

The output of the plant is a continuous-time signal. The error signal is converted into digital form by the sample-and-hold circuit and the analog-to-digital converter. The conversion is done at the sampling time. The digital computer

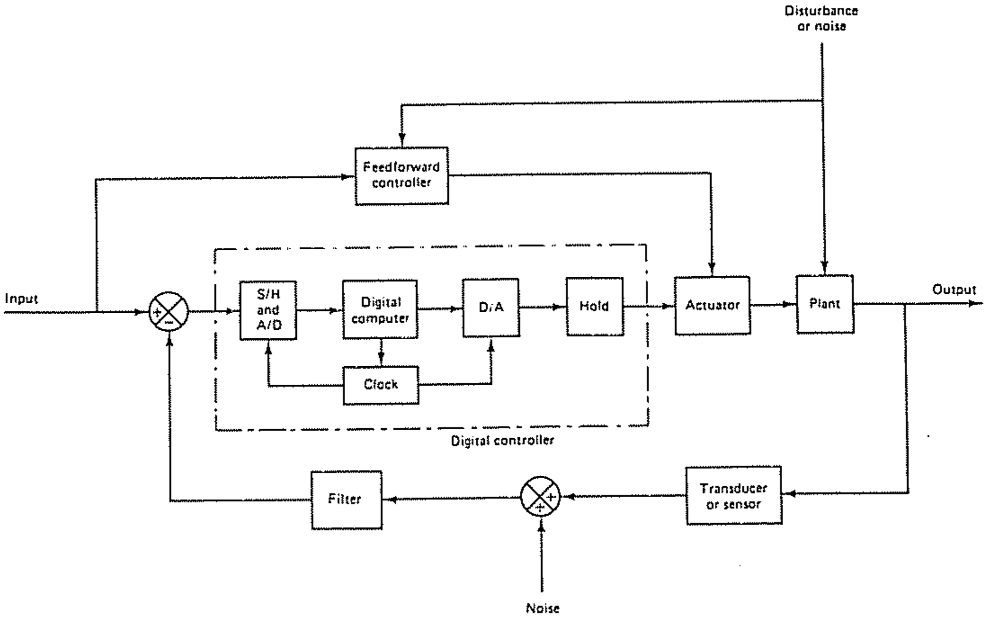
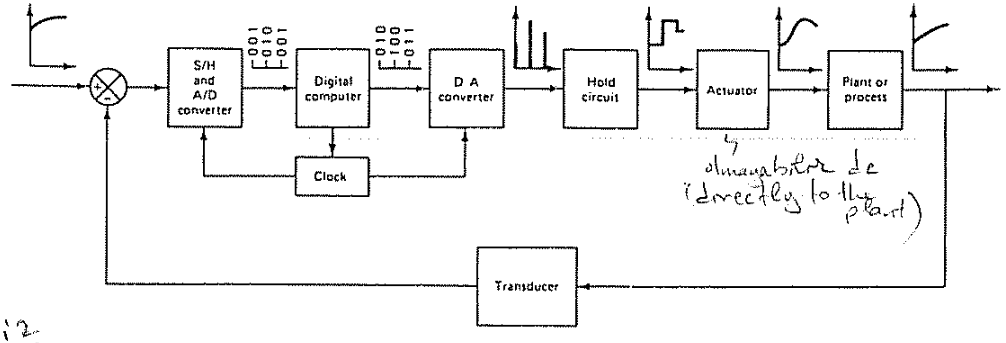


Figure 1–2 Block diagram of a digital control system

piecewise continuous-time signal



dinamiklere de (directly to the plant)

c;i2

Figure 1–3 Block diagram of a digital control system showing signals in binary or graphic form.

processes the sequences of numbers by means of an algorithm and produces new sequences of numbers. At every sampling instant a coded number (usually a binary number consisting of eight or more binary digits) must be converted to a physical control signal, which is usually a continuous-time or analog signal. The digital-to-analog converter and the hold circuit convert the sequence of numbers in numerical code into a piecewise continuous-time signal. The real-time clock in the computer synchronizes the events. The output of the hold circuit, a continuous-time signal, is fed to the plant, either directly or through the actuator, to control its dynamics.

d
disc-time
ling or
tization

The operation that transforms continuous-time signals into discrete-time data is called *sampling* or *discretization*. The reverse operation, the operation that transforms discrete-time data into a continuous-time signal, is called *data-hold*; it amounts to a reconstruction of a continuous-time signal from the sequence of discrete-time data. It is usually done using one of the many extrapolation techniques. In many cases it is done by keeping the signal constant between the successive sampling instants. (We shall discuss such extrapolation techniques in Section 1–4.)

cont-time
-a-hold

The sample-and-hold (S/H) circuit and analog-to-digital (A/D) converter convert the continuous-time signal into a sequence of numerically coded binary words. Such an A/D conversion process is called *coding* or *encoding*. The combination of the S/H circuit and analog-to-digital converter may be visualized as a switch that closes instantaneously at every time interval *T* and generates a sequence of numbers in numerical code. The digital computer operates on such numbers in numerical code and generates a desired sequence of numbers in numerical code. The digital-to-analog (D/A) conversion process is called *decoding*.

Definitions of Terms. Before we discuss digital control systems in detail, we need to define some of the terms that appear in the block diagram of Figure 1–3.

Sample-and-Hold (S/H). "Sample-and-hold" is a general term used for a sample-and-hold amplifier. It describes a circuit that receives an analog input signal and holds this signal at a constant value for a specified period of time. Usually the signal is electrical, but other forms are possible, such as optical and mechanical.

A/D conversion : Coding or encoding (from cont-time signal to numerically coded binary words)
D/A " : decoding

Analog-to-Digital Converter (A/D). An analog-to-digital converter, also called an encoder, is a device that converts an analog signal into a digital signal, usually a numerically coded signal. Such a converter is needed as an interface between an analog component and a digital component. A sample-and-hold circuit is often an integral part of a commercially available A/D converter. The conversion of an analog signal into the corresponding digital signal (binary number) is an approximation, because the analog signal can take on an infinite number of values, whereas the variety of different numbers that can be formed by a finite set of digits is limited. This approximation process is called *quantization*. (More on quantization is presented in Section 1–3.)

Digital-to-Analog Converter (D/A). A digital-to-analog converter, also called a decoder, is a device that converts a digital signal (numerically coded data) into an analog signal. Such a converter is needed as an interface between a digital component and an analog component.

Plant or Process. A plant is any physical object to be controlled. Examples are a furnace, a chemical reactor, and a set of machine parts functioning together to perform a particular operation, such as a servo system or a spacecraft.

A process is generally defined as a progressive operation or development marked by a series of gradual changes that succeed one another in a relatively fixed way and lead toward a particular result or end. In this book we call any operation to be controlled a process. Examples are chemical, economic, and biological processes.

The most difficult part in the design of control systems may lie in the accurate modeling of a physical plant or process. There are many approaches to the plant or process model, but, even so, a difficulty may exist, mainly because of the absence of precise process dynamics and the presence of poorly defined random parameters in many physical plants or processes. Thus, in designing a digital controller, it is necessary to recognize the fact that the mathematical model of a plant or process in many cases is only an approximation of the physical one. Exceptions are found in the modeling of electromechanical systems and hydraulic-mechanical systems, since these may be modeled accurately. For example, the modeling of a robot arm system may be accomplished with great accuracy.

Transducer. A transducer is a device that converts an input signal into an output signal of another form, such as a device that converts a pressure signal into a voltage output. The output signal, in general, depends on the past history of the input.

Transducers may be classified as analog transducers, sampled-data transducers, or digital transducers. An analog transducer is a transducer in which the input and output signals are continuous functions of time. The magnitudes of these signals may be any values within the physical limitations of the system. A sampled-data transducer is one in which the input and output signals occur only at discrete instants of time (usually periodic), but the magnitudes of the signals, as in the case of the analog transducer, are unquantized. A digital transducer is one in which the input and output signals occur only at discrete instants of time and the signal magnitudes are quantized (that is, they can assume only certain discrete levels).

Types of Sampling Operations. As stated earlier, a signal whose independent variable t is discrete is called a discrete-time signal. A sampling operation is basic in transforming a continuous-time signal into a discrete-time signal.

There are several different types of sampling operations of practical importance:

1. *Periodic sampling.* In this case, the sampling instants are equally spaced, or $t_k = kT\ (k = 0, 1, 2, \ldots)$. Periodic sampling is the most conventional type of sampling operation.
2. *Multiple-order sampling.* The pattern of the t_k's is repeated periodically; that is, $t_{k+r} - t_k$ is constant for all k.
3. *Multiple-rate sampling.* In a control system having multiple loops, the largest time constant involved in one loop may be quite different from that in other loops. Hence, it may be advisable to sample slowly in a loop involving a large time constant, while in a loop involving only small time constants the sampling rate must be fast. Thus, a digital control system may have different sampling periods in different feedback paths or may have multiple sampling rates.
4. *Random sampling.* In this case, the sampling instants are random, or t_k is a random variable.

In this book we shall treat only the case where the sampling is periodic.

1–3 QUANTIZING AND QUANTIZATION ERROR

The main functions involved in analog-to-digital conversion are sampling, amplitude quantizing, and coding. When the value of any sample falls between two adjacent "permitted" output states, it must be read as the permitted output state nearest the actual value of the signal. The process of representing a continuous or analog signal by a finite number of discrete states is called *amplitude quantization*. That is, "quantizing" means transforming a continuous or analog signal into a set of discrete states. (Note that quantizing occurs whenever a physical quantity is represented numerically.)

The output state of each quantized sample is then described by a numerical code. The process of representing a sample value by a numerical code (such as a binary code) is called *encoding* or *coding*. Thus, encoding is a process of assigning a digital word or code to each discrete state. The sampling period and quantizing levels affect the performance of digital control systems. So they must be determined carefully.

Quantizing. The standard number system used for processing digital signals is the binary number system. In this system the code group consists of n pulses each indicating either "on" (1) or "off" (0). In the case of quantizing, n "on–off" pulses can represent 2^n amplitude levels or output states.

The quantization level Q is defined as the range between two adjacent decision points and is given by

Samplings
Amplitude quantizing } A to D conversion
Coding
(binary)

8-bit
128
64
32
16
8
4
2
1
255

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | $\equiv 128+32+1 = \underline{161}$
MSB LSB

n: # bit
Q: quantization level
FSR: full-scale range

$$Q = \frac{\text{FSR}}{2^n}$$

where the FSR is the full-scale range. Note that the leftmost bit of the natural binary code has the most weight (one-half of the full scale) and is called the most significant bit (MSB). The rightmost bit has the least weight ($1/2^n$ times the full scale) and is called the least significant bit (LSB). Thus,

analog [illegible]

$$\text{LSB} = \frac{\text{FSR}}{2^n} = Q$$

The least significant bit is the quantization level Q.

Quantization Error. Since the number of bits in the digital word is finite, A/D conversion results in a finite resolution. That is, the digital output can assume only a finite number of levels, and therefore an analog number must be rounded off to the nearest digital level. Hence, any A/D conversion involves quantization error. Such quantization error varies between 0 and $\pm\frac{1}{2}Q$. This error depends on the fineness of the quantization level and can be made as small as desired by making the quantization level smaller (that is, by increasing the number of bits n). In practice, there is a maximum for the number of bits n, and so there is always some error due to quantization. The uncertainty present in the quantization process is called *quantization noise*.

Quant. error $= \pm\frac{1}{2}Q$
$= \frac{1}{2}\frac{FSR}{2^n} = \frac{FSR}{2^{n+1}}$

To determine the desired size of the quantization level (or the number of output states) in a given digital control system, the engineer must have a good understanding of the relationship between the size of the quantization level and the resulting error. The variance of the quantization noise is an important measure of quantization error, since the variance is proportional to the average power associated with the noise.

Figure 1–4(a) shows a block diagram of a quantizer together with its input–output characteristics. For an analog input $x(t)$, the output $y(t)$ takes on only a finite number of levels, which are integral multiples of the quantization level Q.

In numerical analysis the error resulting from neglecting the remaining digits is called the *round-off error*. Since the quantizing process is an approximating process in that the analog quantity is approximated by a finite digital number, the quantization error is a round-off error. Clearly, the finer the quantization level is, the smaller the round-off error.

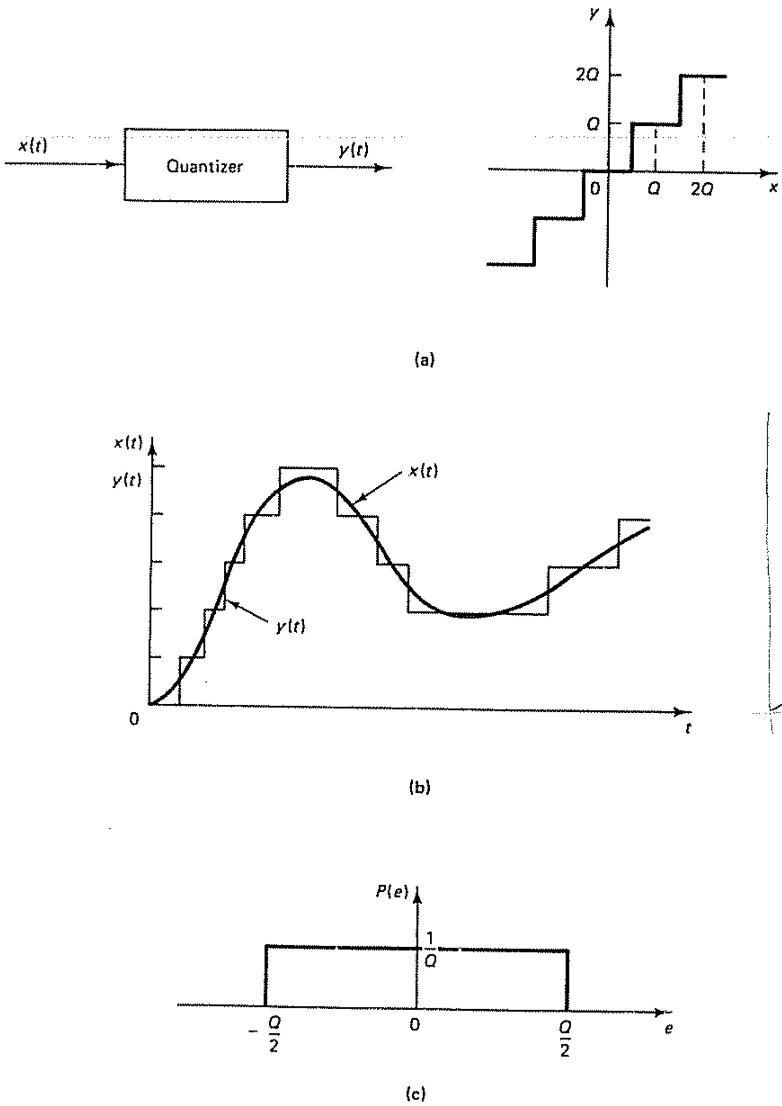
Figure 1–4(b) shows an analog input $x(t)$ and the discrete output $y(t)$, which is in the form of a staircase function. The quantization error $e(t)$ is the difference between the input signal and the quantized output, or

$$e(t) = x(t) - y(t) \quad ✓$$

Note that the magnitude of the quantized error is

$$0 \le |e(t)| \le \tfrac{1}{2}Q \quad ✓$$

For a small quantization level Q, the nature of the quantization error is similar to that of random noise. And, in effect, the quantization process acts as a source of random noise. In what follows we shall obtain the variance of the quantization noise. Such variance can be obtained in terms of the quantization level Q.



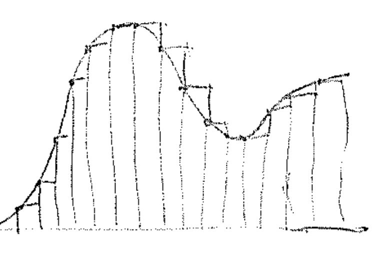


Figure 1–4 (a) Block diagram of a quantizer and its input–output characteristics; (b) analog input $x(t)$ and discrete output $y(t)$; (c) probability distribution $P(e)$ of quantization error $e(t)$

Suppose that the quantization level Q is small and we assume that the quantization error $e(t)$ is distributed uniformly between $-\frac{1}{2}Q$ and $\frac{1}{2}Q$ and that this error acts as a white noise. [This is obviously a rather rough assumption. However, since the quantization error signal $e(t)$ is of a small amplitude, such an assumption may be acceptable as a first-order approximation.] The probability distribution $P(e)$ of

$$\sigma^2 = \frac{1}{Q}\int_{-Q/2}^{+Q/2} \xi^2 \, d\xi = \frac{1}{3Q} \left| \xi^3 \right|_{-Q/2}^{Q/2} = \frac{1}{3Q}\left(\frac{Q^3}{8} + \frac{Q^3}{8}\right) = \frac{Q^2}{12}$$

signal $e(t)$ may be plotted as shown in Figure 1–4(c). The average value of $e(t)$ is zero, or $\overline{e(t)} = 0$. Then the variance σ^2 of the quantization noise is

$$\sigma^2 = E[e(t) - \overline{e(t)}]^2 = \frac{1}{Q}\int_{-Q/2}^{Q/2} \xi^2 \, d\xi = \frac{Q^2}{12} \checkmark$$

std dev $= \frac{Q}{2\sqrt{3}} \cong \frac{Q}{3.4}$

Thus, if the quantization level Q is small compared with the average amplitude of the input signal, then the variance of the quantization noise is seen to be one-twelfth of the square of the quantization level.

1–4 DATA ACQUISITION, CONVERSION, AND DISTRIBUTION SYSTEMS

With the rapid growth in the use of digital computers to perform digital control actions, both the data-acquisition system and the distribution system have become an important part of the entire control system.

The signal conversion that takes place in the digital control system involves the following operations:

1. Multiplexing and demultiplexing
2. Sample and hold
3. Analog-to-digital conversion (sampling, quantizing and encoding)
4. Digital-to-analog conversion (decoding)

Figure 1–5(a) shows a block diagram of a data-acquisition system, and Figure 1–5(b) shows a block diagram of a data-distribution system.

In the data-acquisition system the input to the system is a physical variable such as position, velocity, acceleration, temperature, or pressure. Such a physical variable is first converted into an electrical signal (a voltage or current signal) by a suitable

Data acquisition:

reverse operations

Data distribution:

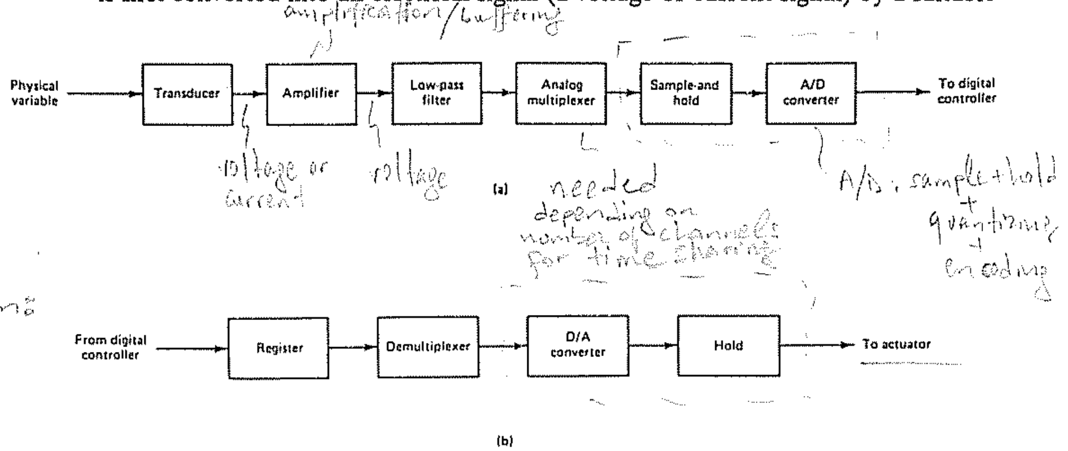


Figure 1–5 (a) Block diagram of a data-acquisition system; (b) block diagram of a data-distribution system.

transducer. Once the physical variable is converted into a voltage or current signal, the rest of the data-acquisition process is done by electronic means.

In Figure 1–5(a) the amplifier (frequently an operational amplifier) that follows the transducer performs one or more of the following functions: It amplifies the voltage output of the transducer; it converts a current signal into a voltage signal; or it buffers the signal. The low-pass filter that follows the amplifier attenuates the high-frequency signal components, such as noise signals. (Note that electronic noises are random in nature and may be reduced by low-pass filters. However, such common electrical noises as power-line interference are generally periodic and may be reduced by means of notch filters.) The output of the low-pass filter is an analog signal. This signal is fed to the analog multiplexer. The output of the multiplexer is fed to the sample-and-hold circuit, whose output is, in turn, fed to the analog-to-digital converter. The output of the converter is the signal in digital form; it is fed to the digital controller.

The reverse of the data-acquisition process is the data-distribution process. As shown in Figure 1–5(b), a data-distribution system consists of registers, a demultiplexer, digital-to-analog converters, and hold circuits. It converts the signal in digital form (binary numbers) into analog form. The output of the D/A converter is fed to the hold circuit. The output of the hold circuit is fed to the analog actuator, which, in turn, directly controls the plant under consideration.

In the following, we shall discuss each individual component involved in the signal-processing system.

Analog Multiplexer. An analog-to-digital converter is the most expensive component in a data-acquisition system. The analog multiplexer is a device that performs the function of time-sharing an A/D converter among many analog channels. The processing of a number of channels with a digital controller is possible because the width of each pulse representing the input signal is very narrow, so the empty space during each sampling period may be used for other signals. If many signals are to be processed by a single digital controller, then these input signals must be fed to the controller through a multiplexer.

Figure 1–6 shows a schematic diagram of an analog multiplexer. The analog

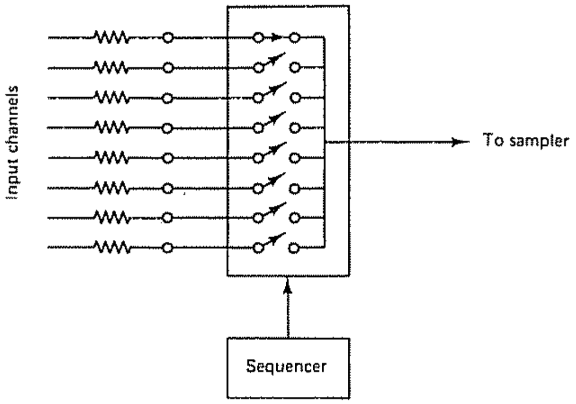
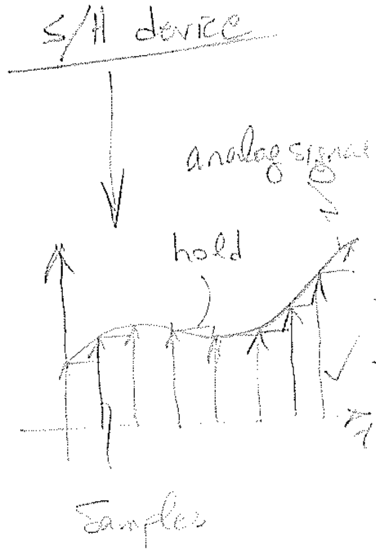


Figure 1–6 Schematic diagram of an analog multiplexer

multiplexer is a multiple switch (usually an electronic switch) that sequentially switches among many analog input channels in some prescribed fashion. The number of channels, in many instances, is 4, 8, or 16. At a given instant of time, only one switch is in the "on" position. When the switch is on in a given input channel, the input signal is connected to the output of the multiplexer for a specified period of time. During the connection time the sample-and-hold circuit samples the signal voltage (analog signal) and holds its value, while the analog-to-digital converter converts the analog value into digital data (binary numbers). Each channel is read in a sequential order, and the corresponding values are converted into digital data in the same sequence.



Demultiplexer. The demultiplexer, which is synchronized with the input sampling signal, separates the composite output digital data from the digital controller into the original channels. Each channel is connected to a D/A converter to produce the output analog signal for that channel.

Sample-and-Hold Circuits. A sampler in a digital system converts an analog signal into a train of amplitude-modulated pulses. The hold circuit holds the value of the sampled pulse signal over a specified period of time. The sample-and-hold is necessary in the A/D converter to produce a number that accurately represents the input signal at the sampling instant. Commercially, sample-and-hold circuits are available in a single unit, known as a sample-and-hold (S/H). Mathematically, however, the sampling operation and the holding operation are modeled separately (see Section 3–2). It is common practice to use a single analog-to-digital converter and multiplex many sampled analog inputs into it.

In practice, sampling duration is very short compared with the sampling period T. When the sampling duration is negligible, the sampler may be considered an "ideal sampler." An ideal sampler enables us to obtain a relatively simple mathematical model for a sample-and-hold. (Such a mathematical model will be discussed in detail in Section 3–2).

Figure 1–7 shows a simplified diagram for the sample-and-hold. The S/H circuit is an analog circuit (simply a voltage memory device) in which an input voltage is acquired and then stored on a high-quality capacitor with low leakage and low dielectric absorption characteristics.

In Figure 1–7 the electronic switch is connected to the hold capacitor. Operational amplifier 1 is an input buffer amplifier with a high input impedance. Operational amplifier 2 is the output amplifier; it buffers the voltage on the hold capacitor.

There are two modes of operation for a sample-and-hold circuit: the tracking mode and the hold mode. When the switch is closed (that is, when the input signal is connected), the operating mode is the tracking mode. The charge on the capacitor in the circuit tracks the input voltage. When the switch is open (the input signal is disconnected), the operating mode is the hold mode and the capacitor voltage holds constant for a specified time period. Figure 1–8 shows the tracking mode and the hold mode.

Note that, practically speaking, switching from the tracking mode to the hold mode is not instantaneous. If the hold command is given while the circuit is in the

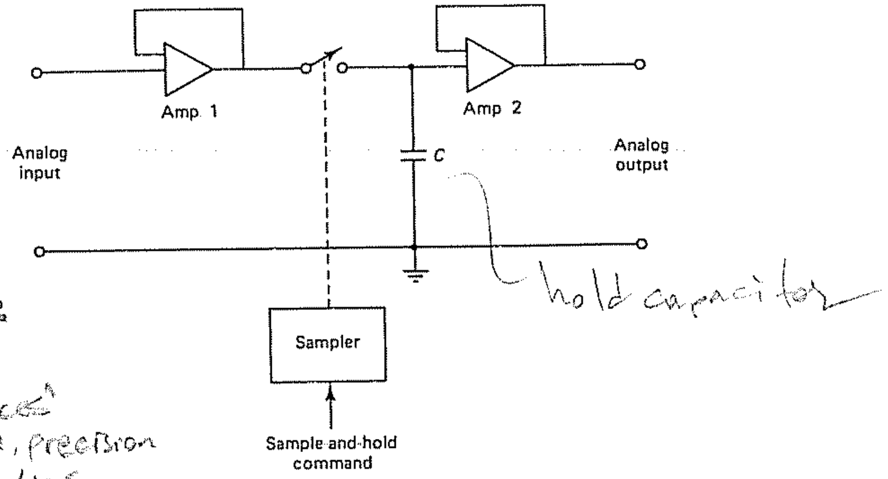


Figure 1–7 Sample-and-hold circuit.

Örnek IC:
AD585
Analog Devices'
high speed, precision
S/H amplifier

tracking mode, then the circuit will stay in the tracking mode for a short while before reacting to the hold command. The time interval during which the switching takes place (that is, the time interval when the measured amplitude is uncertain) is called the *aperture time*.

The output voltage during the hold mode may decrease slightly. The hold mode droop may be reduced by using a high-input-impedance output buffer amplifier. Such an output buffer amplifier must have very low bias current.

The sample-and-hold operation is controlled by a periodic clock.

Types of Analog-to-Digital (A/D) Converters. As stated earlier, the process by which a sampled analog signal is quantized and converted to a binary number is called *analog-to-digital conversion*. Thus, an A/D converter transforms an analog

Sampling + quantizing + Coding
||
A/D conversion

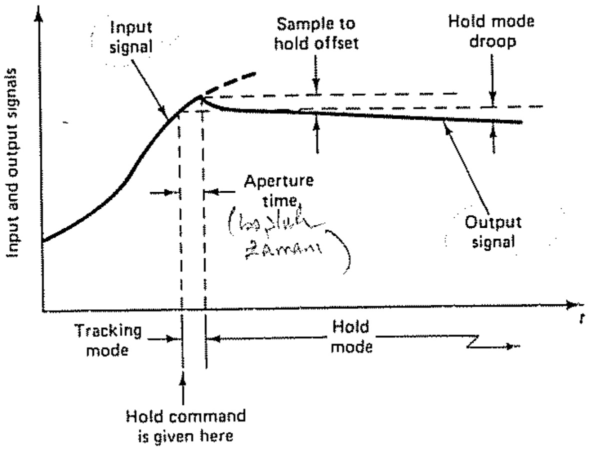


Figure 1–8 Tracking mode and hold mode

Definitions, operations & types of
HW4: ADC & DAC

signal (usually in the form of a voltage or current) into a digital signal or numerically coded word. In practice, the logic is based on binary digits composed of 0s and 1s, and the representation has only a finite number of digits. The A/D converter performs the operations of sample-and-hold, quantizing, and encoding. Note that in the digital system a pulse is supplied every sampling period T by a clock. The A/D converter sends a digital signal (binary number) to the digital controller each time a pulse arrives.

Among the many A/D circuits available, the following types are used most frequently:

1. Successive-approximation type → fast & most frequently used
2. Integrating type
3. Counter type → the simplest A/D
4. Parallel type

Each of these four types has its own advantages and disadvantages. In any particular application, the conversion speed, accuracy, size, and cost are the main factors to be considered in choosing the type of A/D converter. (If greater accuracy is needed, for example, the number of bits in the output signal must be increased.)

As will be seen, analog-to-digital converters use as part of their feedback loops digital-to-analog converters. The simplest type of A/D converter is the counter type. The basic principle on which it works is that clock pulses are applied to the digital counter in such a way that the output voltage of the D/A converter (that is, part of the feedback loop in the A/D converter) is stepped up one least significant bit (LSB) at a time, and the output voltage is compared with the analog input voltage once for each pulse. When the output voltage has reached the magnitude of the input voltage, the clock pulses are stopped. The counter output voltage is then the digital output.

The successive-approximation type of A/D converter is much faster than the counter type and is the one most frequently used. Figure 1–9 shows a schematic diagram of the successive-approximation type of A/D converter.

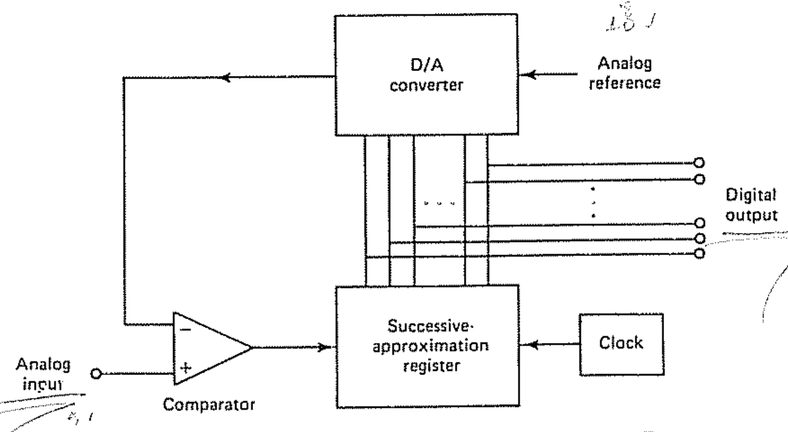


Figure 1–9 Schematic diagram of a successive-approximation-type of A/D converter.

The principle of operation of this type of A/D converter is as follows. The successive-approximation register (SAR) first turns on the most significant bit (half the maximum) and compares it with the analog input. The comparator decides whether to leave the bit on or turn it off. If the analog input voltage is larger, the most significant bit is set on. The next step is to turn on bit 2 and then compare the analog input voltage with three-fourths of the maximum. After n comparisons are completed, the digital output of the successive-approximation register indicates all those bits that remain on and produces the desired digital code. Thus, this type of A/D converter sets 1 bit each clock cycle, and so it requires only n clock cycles to generate n bits, where n is the resolution of the converter in bits. (The number n of bits employed determines the accuracy of conversion.) The time required for the conversion is approximately 2 μsec or less for a 12-bit conversion. ✓

Errors in A/D Converters. Actual analog-to-digital signal converters differ from the ideal signal converter in that the former always have some errors, such as offset error, linearity error, and gain error, the characteristics of which are shown in Figure 1–10. Also, it is important to note that the input–output characteristics change with time and temperature.

Finally, it is noted that commercial converters are specified for three basic temperature ranges: commercial (0°C to 70°C), industrial (−25°C to 85°C), and military (−55°C to 125°C).

Digital-to-Analog (D/A) Converters. At the output of the digital controller the digital signal must be converted to an analog signal by the process called *digital-to-analog conversion*. A D/A converter is a device that transforms a digital input (binary numbers) to an analog output. The output, in most cases, is the voltage signal.

For the full range of the digital input, there are 2^n corresponding different analog values, including 0. For the digital-to-analog conversion there is a one-to-one correspondence between the digital input and the analog output.

So, there is no quantization error

Two methods are commonly used for digital-to-analog conversion: the method using weighted resistors, and the one using the R–$2R$ ladder network. The former is simple in circuit configuration, but its accuracy may not be very good. The latter is a little more complicated in configuration, but is more accurate.

Figure 1–11 shows a schematic diagram of a D/A converter using weighted resistors. The input resistors of the operational amplifier have their resistance values weighted in a binary fashion. When the logic circuit receives binary 1, the switch (actually an electronic gate) connects the resistor to the reference voltage. When the logic circuit receives binary 0, the switch connects the resistor to ground. The digital-to-analog converters used in common practice are of the parallel type: all bits act simultaneously upon application of a digital input (binary numbers).

The D/A converter thus generates the analog output voltage corresponding to the given digital voltage. For the D/A converter shown in Figure 1–11, if the binary number is $b_3 b_2 b_1 b_0$, where each of the b's can be either a 0 or a 1, then the output is

$$V_o = \frac{R_o}{R}\left(b_3 + \frac{b_2}{2} + \frac{b_1}{4} + \frac{b_0}{8}\right)V_{\text{ref}} \qquad ✓$$

/A twomethods :

1 - weighted resistor
simple but
accuracy is low
2 - R-2R ladder network
complicated but
high accuracy

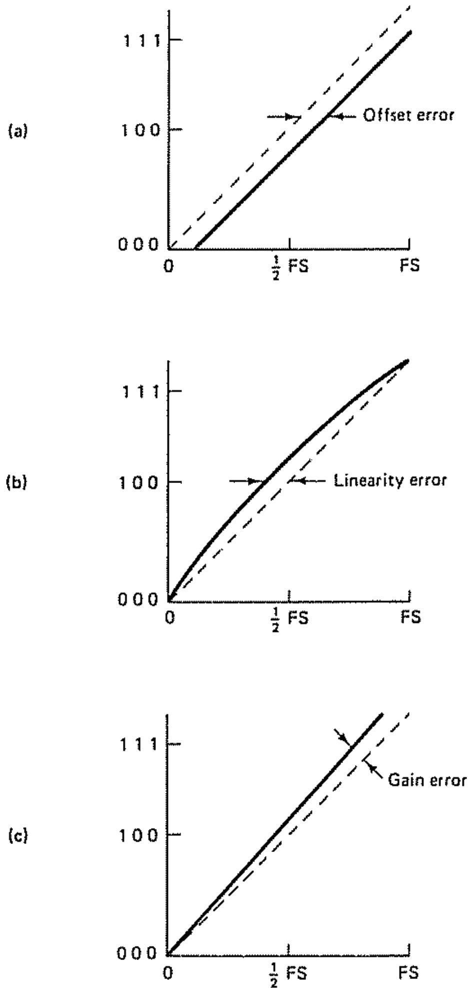


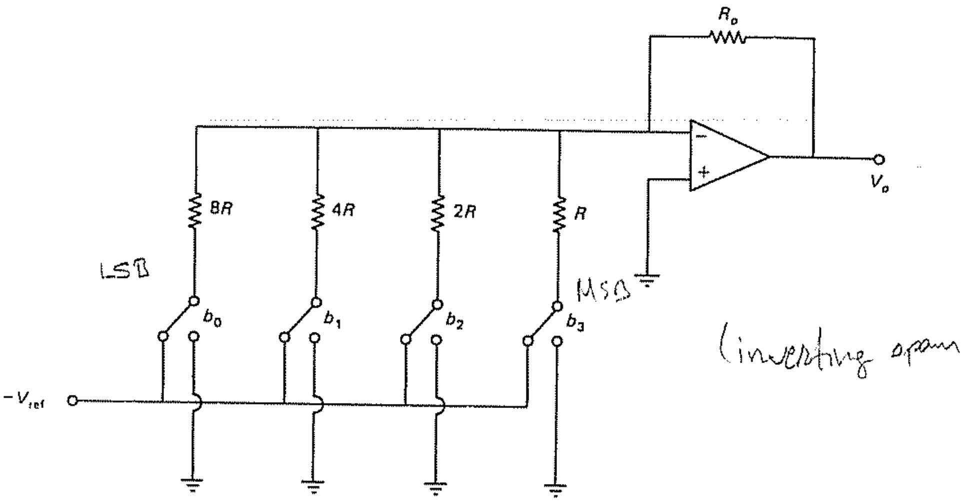
Figure 1–10 Errors in A/D converters: (a) offset error; (b) linearity error; (c) gain error.

Notice that as the number of bits is increased the range of resistor values becomes large and consequently the accuracy becomes poor.

Figure 1–12 shows a schematic diagram of an n-bit D/A converter using an R–$2R$ ladder circuit. Note that with the exception of the feedback resistor (which is $3R$) all resistors involved are either R or $2R$. This means that a high level of accuracy can be achieved. The output voltage in this case can be given by

$$V_o = \frac{1}{2}\left(b_{n-1} + \frac{1}{2}b_{n-2} + \cdots + \frac{1}{2^{n-1}}b_0\right)V_{\text{ref}}$$

Reconstructing the Input Signal by Hold Circuits. The sampling operation produces an amplitude-modulated pulse signal. The function of the hold operation is

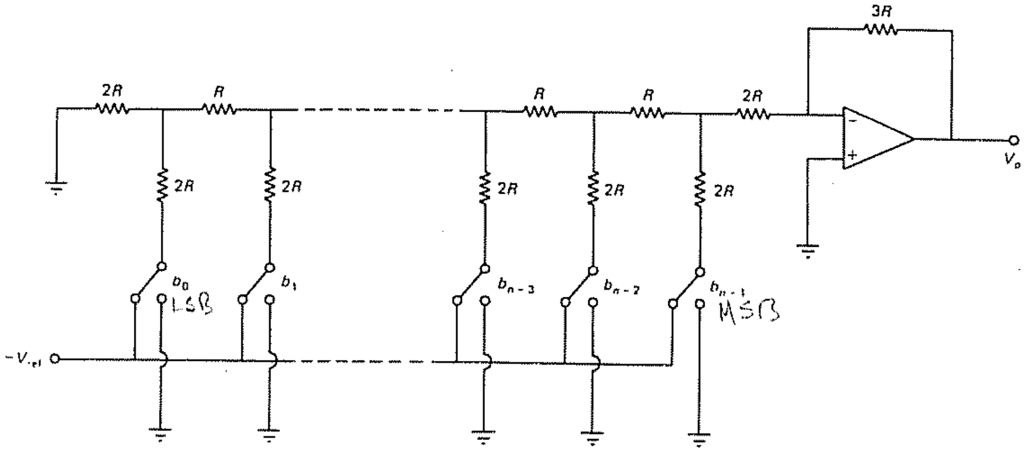


(inverting opamp)

Figure 1–11 Schematic diagram of a D/A converter using weighted resistors. (4 bit)

to reconstruct the analog signal that has been transmitted as a train of pulse samples. That is, the purpose of the hold operation is to fill in the spaces between sampling periods and thus roughly reconstruct the original analog input signal.

The hold circuit is designed to extrapolate the output signal between successive points according to some prescribed manner. The staircase waveform of the output shown in Figure 1–13 is the simplest way to reconstruct the original input signal. The hold circuit that produces such a staircase waveform is called a *zero-order hold*. Because of its simplicity, the zero-order hold is commonly used in digital control systems.



inverting opamp

Figure 1–12 n-Bit D/A converter using an R–$2R$ ladder circuit.

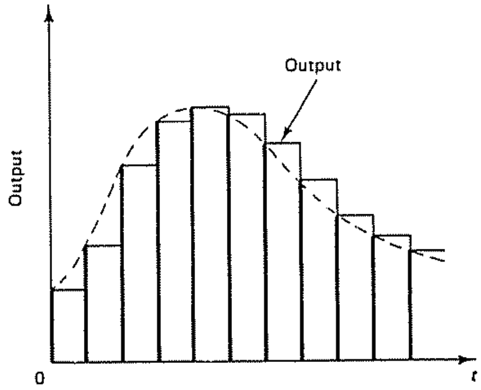


Figure 1–13 Output from a zero-order hold.

More sophisticated hold circuits are available than the zero-order hold. These are called higher-order hold circuits and include the first-order hold and the second-order hold. Higher-order hold circuits will generally reconstruct a signal more accurately than a zero-order hold, but with some disadvantages, as explained next.

The first-order hold retains the value of the previous sample, as well as the present one, and predicts, by extrapolation, the next sample value. This is done by generating an output slope equal to the slope of a line segment connecting previous and present samples and projecting it from the value of the present sample, as shown in Figure 1–14.

As can easily be seen from the figure, if the slope of the original signal does not change much, the prediction is good. If, however, the original signal reverses its slope, then the prediction is wrong and the output goes in the wrong direction, thus causing a large error for the sampling period considered.

An interpolative first-order hold, also called a *polygonal* hold, reconstructs the original signal much more accurately. This hold circuit also generates a straight-line output whose slope is equal to that joining the previous sample value and the present sample value, but this time the projection is made from the current sample point with

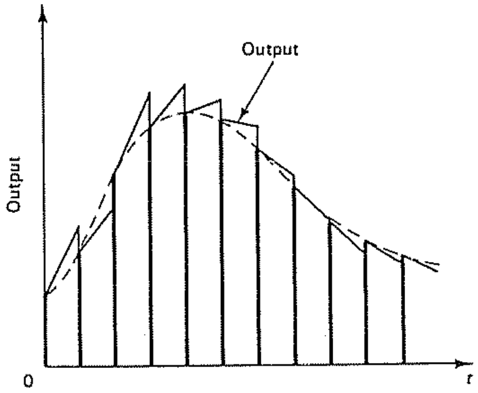


Figure 1–14 Output from a first-order hold.

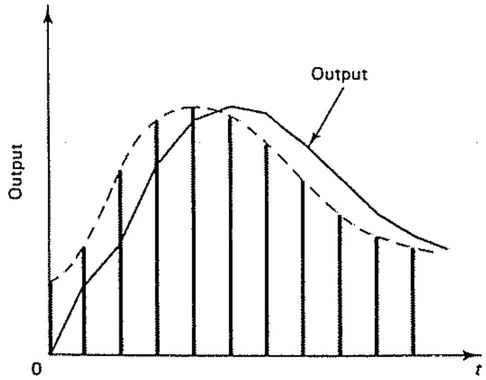


Figure 1–15 Output from an interpolative first-order hold (polygonal hold).

the amplitude of the previous sample. Hence, the accuracy in reconstructing the original signal is better than for other hold circuits, but there is a one-sampling-period delay, as shown in Figure 1–15. In effect, the better accuracy is achieved at the expense of a delay of one sampling period. From the viewpoint of the stability of closed-loop systems, such a delay is not desirable, and so the interpolative first-order hold (polygonal hold) is not used in control system applications.

1–5 CONCLUDING COMMENTS

In concluding this chapter we shall compare digital controllers and analog controllers used in industrial control systems and review digital control of processes. Then we shall present an outline of the book.

Digital Controllers and Analog Controllers. Digital controllers operate only on numbers. Decision making is one of their important functions. They are often used to solve problems involved in the optimal overall operation of industrial plants.

Digital controllers are extremely versatile. They can handle nonlinear control equations involving complicated computations or logic operations. A very much wider class of control laws can be used in digital controllers than in analog controllers. Also, in the digital controller, by merely issuing a new program the operations being performed can be changed completely. This feature is particularly important if the control system is to receive operating information or instructions from some computing center where economic analysis and optimization studies are made.

Digital controllers are capable of performing complex computations with constant accuracy at high speed and can have almost any desired degree of computational accuracy at relatively little increase in cost.

Originally, digital controllers were used as components only in large-scale control systems. At present, however, thanks to the availability of inexpensive microcomputers, digital controllers are being used in many large- and small-scale control systems. In fact, digital controllers are replacing the analog controllers that have been used in many small-scale control systems. Digital controllers are often superior in performance and lower in price than their analog counterparts.

Analog controllers represent the variables in an equation by continuous physical quantities. They can easily be designed to serve satisfactorily as non-decision-making controllers. But the cost of analog computers or analog controllers increases rapidly as the complexity of the computations increases, if constant accuracy is to be maintained.

There are additional advantages of digital controllers over analog controllers. Digital components, such as sample-and-hold circuits, A/D and D/A converters, and digital transducers, are rugged in construction, highly reliable, and often compact and lightweight. Moreover, digital components have high sensitivity, are often cheaper than their analog counterparts, and are less sensitive to noise signals. And, as mentioned earlier, digital controllers are flexible in allowing programming changes.

Digital Control of Processes. In industrial process control systems, it is generally not practical to operate for a very long time at steady state, because certain changes may occur in production requirements, raw materials, economic factors, and processing equipments and techniques. Thus, the transient behavior of industrial processes must always be taken into consideration. Since there are interactions among process variables, using only one process variable for each control agent is not suitable for really complete control. By the use of a digital controller, it is possible to take into account all process variables, together with economic factors, production requirements, equipment performance, and all other needs, and thereby to accomplish optimal control of industrial processes.

Note that a system capable of controlling a process as completely as possible will have to solve complex equations. The more complete the control, the more important it is that the correct relations between operating variables be known and used. The system must be capable of accepting instructions from such varied sources as computers and human operators and must also be capable of changing its control subsystem completely in a short time. Digital controllers are most suitable in such situations. In fact, an advantage of the digital controller is flexibility, that is, ease of changing control schemes by reprogramming.

In the digital control of a complex process, the designer must have a good knowledge of the process to be controlled and must be able to obtain its mathematical model. (The mathematical model may be obtained in terms of differential equations or difference equations, or in some other form.) The designer must be familiar with the measurement technology associated with the output of the process and other variables involved in the process. He or she must have a good working knowledge of digital computers as well as modern control theory. If the process is complicated, the designer must investigate several different approaches to the design of the control system. In this respect, a good knowledge of simulation techniques is helpful.

Outline of the Book. The objective of this book is to present a detailed account of the control theory that is relevant to the analysis and design of discrete-time control systems. Our emphasis is on understanding the basic concepts involved.

In this book, digital controllers are often designed in the form of pulse transfer functions or equivalent difference equations, which can be easily implemented in the form of computer programs.

The outline of the book is as follows. Chapter 1 has presented introductory material. Chapter 2 presents the *z* transform theory. This chapter includes *z* transforms of elementary functions, important properties and theorems of the *z* transform, the inverse *z* transform, and the solution of difference equations by the *z* transform method. Chapter 3 treats background materials for the *z* plane analysis of control systems. This chapter includes discussions of impulse sampling and reconstruction of original signals from sampled signals, pulse transfer functions, and realization of digital controllers and digital filters.

Chapter 4 first presents mapping between the *s* plane and the *z* plane and then discusses stability analysis of closed-loop systems in the *z* plane, followed by transient and steady-state response analyses, design by the root-locus and frequency-response methods, and an analytical design method. Chapter 5 gives state-space representation of discrete-time systems, the solution of discrete-time state-space equations, and the pulse transfer function matrix. Then, discretization of continuous-time state-space equations and Liapunov stability analysis are treated.

Chapter 6 presents control systems design in the state space. We begin the chapter with a detailed presentation of controllability and observability. We then present design techniques based on pole placement, followed by discussion of full-order state observers and minimum-order state observers. We conclude this chapter with the design of servo systems. Chapter 7 treats the polynomial-equations approach to the design of control systems. We begin the chapter with discussions of Diophantine equations. Then we present the design of regulator systems and control systems using the solution of Diophantine equations. The approach here is an alternative to the pole-placement approach combined with minimum-order observers. The design of model-matching control systems is included in this chapter. Finally, Chapter 8 treats quadratic optimal control problems in detail.

The state-space analysis and design of discrete-time control systems, presented in Chapters 5, 6, and 8, make extensive use of vectors and matrices. In studying these chapters the reader may, as need arises, refer to Appendix A, which summarizes the basic materials of vector-matrix analysis. Appendix B presents materials in *z* transform theory not included in Chapter 2. Appendix C treats pole-placement design problems when the control is a vector quantity.

In each chapter, except Chapter 1, the main text is followed by solved problems and unsolved problems. The reader should study all solved problems carefully. Solved problems are an integral part of the text. Appendixes A, B, and C are followed by solved problems. The reader who studies these solved problems will have an increased understanding of the material presented.