1

Computer Control

1.1 Introduction

Practically all control systems that are implemented today are based on computer control. It is therefore important to understand computer-controlled systems well. Such systems can be viewed as approximations of analog-control systems, but this is a poor approach because the full potential of computer control is not used. At best the results are only as good as those obtained with analog control. It is much better to master computer-controlled systems, so that the full potential of computer control can be used. There are also phenomena that occur in computer-controlled systems that have no correspondence in analog systems. It is important for an engineer to understand this. The main goal of this book is to provide a solid background for understanding, analyzing, and designing computer-controlled systems.

A computer-controlled system can be described schematically as in Fig. 1.1. The output from the process $y(t)$ is a continuous-time signal. The output is converted into digital form by the analog-to-digital (A-D) converter. The A-D converter can be included in the computer or regarded as a separate unit, according to one's preference. The conversion is done at the sampling times, t_k. The computer interprets the converted signal, $\{y(t_k)\}$, as a sequence of numbers, processes the measurements using an algorithm, and gives a new sequence of numbers, $\{u(t_k)\}$. This sequence is converted to an analog signal by a digital-to-analog (D-A) converter. The events are synchronized by the real-time clock in the computer. The digital computer operates sequentially in time and each operation takes some time. The D-A converter must, however, produce a continuous-time signal. This is normally done by keeping the control signal constant between the conversions. In this case the system runs open loop in the time interval between the sampling instants because the control signal is constant irrespective of the value of the output.

The computer-controlled system contains both continuous-time signals and *sampled*, or *discrete-time*, signals. Such systems have traditionally been called

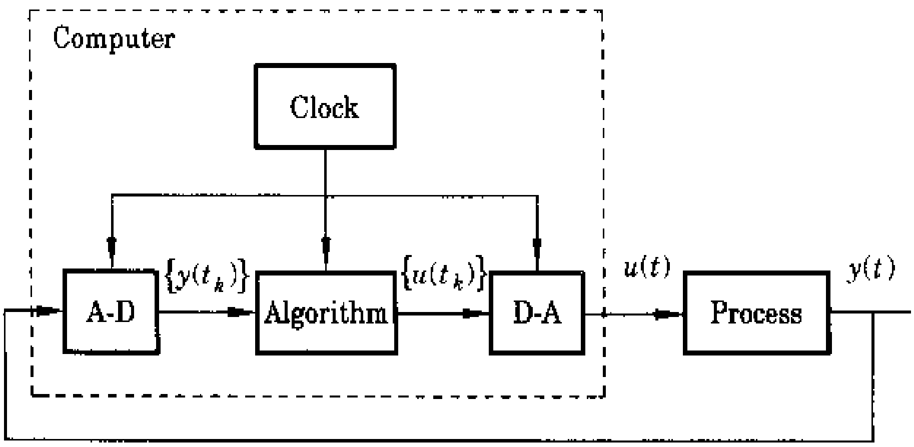


Figure 1.1 Schematic diagram of a computer-controlled system.

sampled-data systems, and this term will be used here as a synonym for *computer-controlled systems*.

The mixture of different types of signals sometimes causes difficulties. In most cases it is, however, sufficient to describe the behavior of the system at the sampling instants. The signals are then of interest only at discrete times. Such systems will be called *discrete-time systems*. Discrete-time systems deal with sequences of numbers, so a natural way to represent these systems is to use difference equations.

The purpose of the book is to present the control theory that is relevant to the analysis and design of computer-controlled systems. This chapter provides some background. A brief overview of the development of computer-control technology is given in Sec. 1.2. The need for a suitable theory is discussed in Sec. 1.3. Examples are used to demonstrate that computer-controlled systems cannot be fully understood by the theory of linear time-invariant continuous-time systems. An example shows not only that computer-controlled systems can be designed using continuous-time theory and approximations, but also that substantial improvements can be obtained by other techniques that use the full potential of computer control. Section 1.4 gives some examples of inherently sampled systems. The development of the theory of sampled-data systems is outlined in Sec. 1.5.

1.2 Computer Technology

The idea of using digital computers as components in control systems emerged around 1950. Applications in missile and aircraft control were investigated first. Studies showed that there was no potential for using the general-purpose digital computers that were available at that time. The computers were too big, they consumed too much power, and they were not sufficiently reliable. For this reason special-purpose computers—digital differential analyzers (DDAs)—were developed for the early aerospace applications.

The idea of using digital computers for process control emerged in the mid-1950s. Serious work started in March 1956 when the aerospace company Thomson Ramo Woodridge (TRW) contacted Texaco to set up a feasibility study. After preliminary discussions it was decided to investigate a polymerization unit at the Port Arthur, Texas, refinery. A group of engineers from TRW and Texaco made a thorough feasibility study, which required about 30 people-years. A computer-controlled system for the polymerization unit was designed based on the RW-300 computer. The control system went on-line March 12, 1959. The system controlled 26 flows, 72 temperatures, 3 pressures, and 3 compositions. The essential functions were to minimize the reactor pressure, to determine an optimal distribution among the feeds of 5 reactors, to control the hot-water inflow based on measurement of catalyst activity, and to determine the optimal recirculation.

The pioneering work done by TRW was noticed by many computer manufacturers, who saw a large potential market for their products. Many different feasibility studies were initiated and vigorous development was started. To discuss the dramatic developments, it is useful to introduce six periods:

Pioneering period ≈ 1955

Direct-digital-control period ≈ 1962

Minicomputer period ≈ 1967

Microcomputer period ≈ 1972

General use of digital control ≈ 1980

Distributed control ≈ 1990

It is difficult to give precise dates, because the development was highly diversified. There was a wide difference between different application areas and different industries; there was also considerable overlap. The dates given refer to the emergence of new approaches.

Pioneering Period

The work done by TRW and Texaco evoked substantial interest in process industries, among computer manufacturers, and in research organizations. The industries saw a potential tool for increased automation, the computer industries saw new markets, and universities saw a new research field. Many feasibility studies were initiated by the computer manufacturers because they were eager to learn the new technology and were very interested in knowing what a proper process-control computer should look like. Feasibility studies continued throughout the sixties.

The computer systems that were used were slow, expensive, and unreliable. The earlier systems used vacuum tubes. Typical data for a computer around 1958 were an addition time of 1 ms, a multiplication time of 20 ms, and a mean time between failures (MTBF) for a central processing unit of 50–100 h. To make full use of the expensive computers, it was necessary to have them perform many tasks. Because the computers were so unreliable, they controlled the process by printing instructions to the process operator or by changing the set points of analog regulators. These supervisory modes of operation were referred to as an *operator guide* and a *set-point control.*

The major tasks of the computer were to find the optimal operating conditions, to perform scheduling and production planning, and to give reports about production and raw-material consumption. The problem of finding the best operating conditions was viewed as a static optimization problem. Mathematical models of the processes were necessary in order to perform the optimization. The models used—which were quite complicated—were derived from physical models and from regression analysis of process data. Attempts were also made to carry out on-line optimization.

Progress was often hampered by lack of process knowledge. It also became clear that it was not sufficient to view the problems simply as static optimization problems; dynamic models were needed. A significant proportion of the effort in many of the feasibility studies was devoted to modeling, which was quite time-consuming because there was a lack of good modeling methodology. This stimulated research into system-identification methods.

A lot of experience was gained during the feasibility studies. It became clear that process control puts special demands on computers. The need to respond quickly to demands from the process led to development of the *interrupt feature*, which is a special hardware device that allows an external event to interrupt the computer in its current work so that it can respond to more urgent process tasks. Many sensors that were needed were not available. There were also several difficulties in trying to introduce a new technology into old industries.

The progress made was closely monitored at conferences and meetings and in journals. A series of articles describing the use of computers in process control was published in the journal *Control Engineering*. By March 1961, 37 systems had been installed. A year later the number of systems had grown to 159. The applications involved control of steel mills and chemical industries and generation of electric power. The development progressed at different rates in different industries. Feasibility studies continued through the 1960s and the 1970s.

Direct-Digital-Control Period

The early installations of control computers operated in a supervisory mode, either as an operator guide or as a set-point control. The ordinary analog-control equipment was needed in both cases. A drastic departure from this approach was made by Imperial Chemical Industries (ICI) in England in 1962. A complete analog instrumentation for process control was replaced by one computer, a Ferranti Argus. The computer measured 224 variables and controlled 129 valves directly. This was the beginning of a new era in process control: Analog technology was simply replaced by digital technology; the function of the system was the same. The name *direct digital control* (DDC) was coined to emphasize that the computer-controlled the process directly. In 1962 a typical process-control computer could add two numbers in 100 μs and multiply them in 1 ms. The MTBF was around 1000 h.

Cost was the major argument for changing the technology. The cost of an analog system increased linearly with the number of control loops; the initial cost of a digital system was large, but the cost of adding an additional loop was small. The digital system was thus cheaper for large installations. Another advantage was that operator communication could be changed drastically; an operator communication panel could replace a large wall of analog instruments. The panel used in the ICI system was very simple—a digital display and a few buttons.

Flexibility was another advantage of the DDC systems. Analog systems were changed by rewiring; computer-controlled systems were changed by reprogramming. Digital technology also offered other advantages. It was easy to have interaction among several control loops. The parameters of a control loop could be made functions of operating conditions. The programming was simplified by introducing special DDC languages. A user of such a language did not need to know anything about programming, but simply introduced inputs, outputs, regulator types, scale factors, and regulator parameters into tables. To the user the systems thus looked like a connection of ordinary regulators. A drawback of the systems was that it was difficult to do unconventional control strategies. This certainly hampered development of control for many years.

DDC was a major change of direction in the development of computer-controlled systems. Interest was focused on the basic control functions instead of the supervisory functions of the earlier systems. Considerable progress was made in the years 1963–1965. Specifications for DDC systems were worked out jointly between users and vendors. Problems related to choice of sampling period and control algorithms, as well as the key problem of reliability, were discussed extensively. The DDC concept was quickly accepted although DDC systems often turned out to be more expensive than corresponding analog systems.

Minicomputer Period

There was substantial development of digitel computer technology in the 1960s. The requirements on a process-control computer were neatly matched with progress in integrated-circuit technology. The computers became smaller, faster, more reliable, and cheaper. The term *minicomputer* was coined for the new computers that emerged. It was possible to design efficient process-control systems by using minicomputers.

The development of minicomputer technology combined with the increasing knowledge gained about process control with computers during the pioneering and DDC periods caused a rapid increase in applications of computer control. Special process-control computers were announced by several manufacturers. A typical process computer of the period had a word length of 16 bits. The primary memory was 8–124 k words. A disk drive was commonly used as a secondary memory. The CDC 1700 was a typical computer of this period, with an addition time of 2 μs and a multiplication time of 7 μs. The MTBF for a central processing unit was about 20,000 h.

An important factor in the rapid increase of computer control in this period was that digital computer control now came in a smaller "unit." It was thus possible to use computer control for smaller projects and for smaller problems. Because of minicomputers, the number of process computers grew from about 5000 in 1970 to about 50,000 in 1975.

Microcomputer Period and General Use of Computer Control

The early use of computer control was restricted to large industrial systems because digital computing was only available in expensive, large, slow, and unreliable machines. The minicomputer was still a fairly large system. Even as performance continued to increase and prices to decrease, the price of a minicomputer mainframe in 1975 was still about $10,000. This meant that a small system rarely cost less than $100,000. Computer control was still out of reach for a large number of control problems. But with the development of the microcomputer in 1972, the price of a card computer with the performance of a 1975 minicomputer dropped to $500 in 1980. Another consequence was that digital computing power in 1980 came in quanta as small as $50. The development of microelectronics has continued with advances in very large-scale integration (VLSI) technology; in the 1990s microprocessors became available for a few dollars. This has had a profound impact on the use of computer control. As a result practically all controllers are now computer-based. Mass markets such as automotive electronics has also led to the development of special-purpose computers, called microcontrollers, in which a standard computer chip has been augmented with A-D and D-A converters, registers, and other features that make it easy to interface with physical equipment.

Practically all control systems developed today are based on computer control. Applications span all areas of control, generation, and distribution of electricity; process control; manufacturing; transportation; and entertainment. Mass-market applications such as automotive electronics, CD players, and videos are particularly interesting because they have motivated computer manufacturers to make chips that can be used in a wide variety of applications.

As an illustration Fig. 1.2 shows an example of a single-loop controller for process control. Such systems were traditionally implemented using pneumatic or electronic techniques, but they are now always computer-based. The controller has the traditional proportional, integral, and derivative actions (PID), which are implemented in a microprocessor. With digital control it is also possible to obtain added functionality. In this particular case, the regulator is provided with automatic tuning, gain scheduling, and continuous adaptetion of feedforward and feedback gains. These functions are difficult to implement with analog techniques. The system is a typical case that shows how the functionality of a traditional product can be improved substantially by use of computer control.

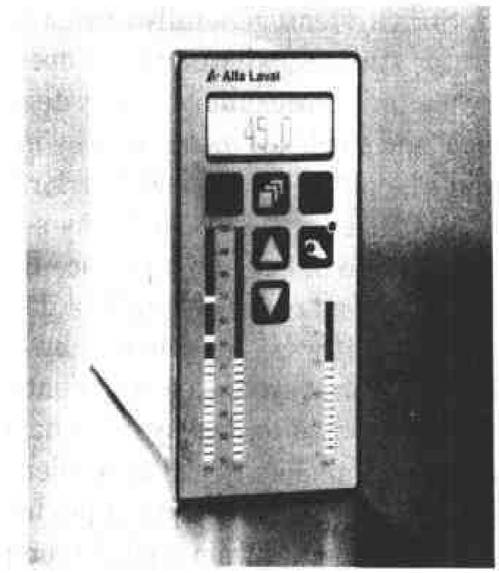


Figure 1.2 A standard single-loop controller for process control. (By courtesy of Alfa Laval Automation, Stockholm, Sweden.)

Logic, Sequencing, and Control

Industrial automation systems have traditionally had two components, controllers and relay logic. Relays were used to sequence operations such as startup and shutdown and they were also used to ensure safety of the operations by providing interlocks. Relays and controllers were handled by different categories of personnel at the plant. Instrument engineers were responsible for the controllers and electricians were responsible for the relay systems. We have already discussed how the controllers were influenced by microcomputers. The relay systems went through a similar change with the advent of microelectronics. The so-called *programmable logic controller (PLC)* emerged in the beginning of the 1970s as replacements for relays. They could be programmed by electricians and in familiar notations, that is, as rungs of relay contact logic or as logic (AND/OR) statements. Americans were the first to bring this novelty to the market, relying primarily on relay contact logic, but the Europeans were hard on their heels, preferring logic statements. The technology became a big success, primarily in the discrete parts manufacturing industry (for obvious reasons). However, in time, it evolved to include regulatory control and data-handling capabilities as well, a development that has broadened the range of applications for it. The attraction was, and is, the ease with which controls, including intraloop dependencies, can be implemented and changed, without any impact on hardware.

Distributed Control

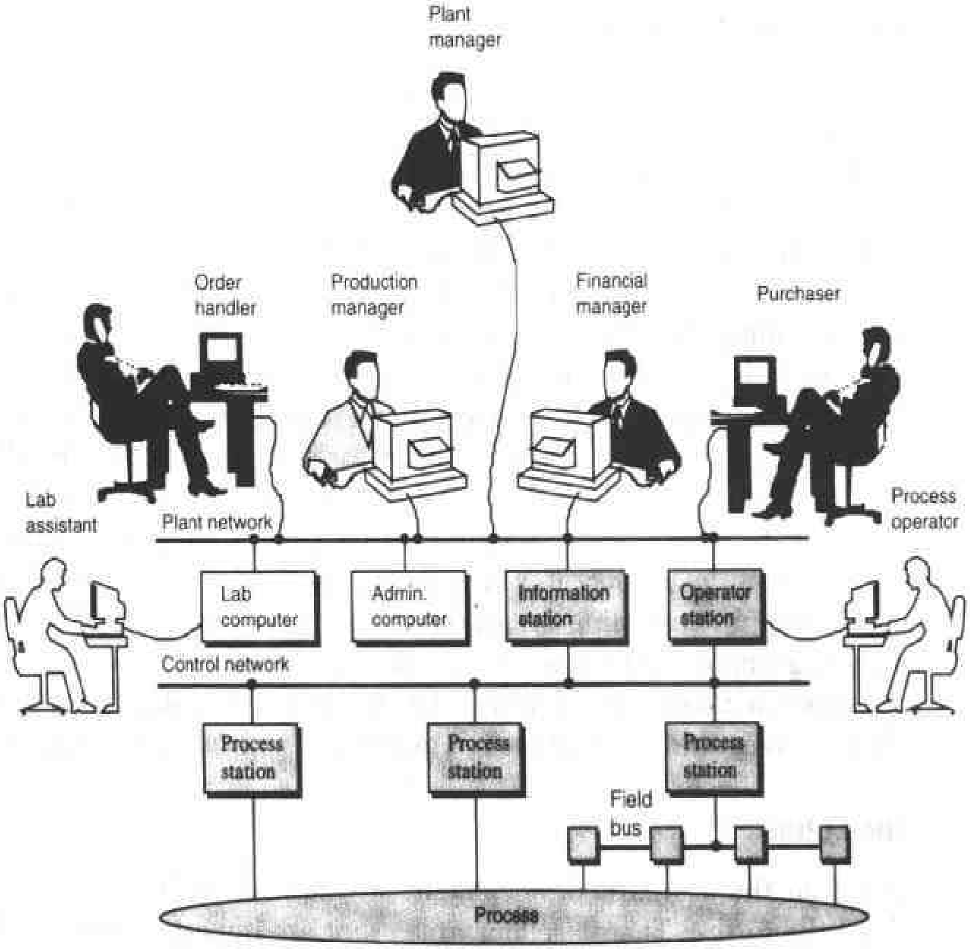
The microprocessor has also had a profound impact on the way computers were applied to control entire production plants. It became economically feasible to develop systems consisting of several interacting microcomputers sharing the overall workload. Such systems generally consist of process stations, controlling the process; operator stations, where process operators monitor activities; and various auxiliary stations, for example, for system configuration and programming, data storage, and so on, all interacting by means of some kind of communications network. The allure was to boost performance by facilitating parallel multitasking, to improve overall availability by not putting "all the eggs in one basket," to further expandability and to reduce the amount of control cabling. The first system of this kind to see the light of day was Honeywell's TDC 2000 (the year was 1975), but it was soon followed by others. The term "distributed control" was coined. The first systems were oriented toward regulatory control, but over the years distributed control systems have adopted more and more of the capabilities of programmable (logic) controllers, making today's distributed control systems able to control all aspects of production and enabling operators to monitor and control activities from a single computer console.

Plantwide Supervision and Control

The next development phase in industrial process-control systems was facilitated by the emergence of common standards in computing, making it possible to integrate virtually all computers and computer systems in industrial plants into a monolithic whole to achieve real-time exchange of data across what used to be closed system borders. Such interaction enables

- top managers to investigate all aspects of operations
- production managers to plan and schedule production on the basis of current information
- order handlers and liaison officers to provide instant and current information to inquiring customers
- process operators to look up the cost accounts and the quality records of the previous production run to do better next time

all from the computer screens in front of them, all in real time. An example of such a system is shown in Fig. 1.3. ABB's Advant OCS (open control system) seems to be a good exponent of this phase. It consists of process controllers with local and/or remóte I/O, operator stations, information management stations, and engineering stations that are interconnected by high-speed communications buses at the field, process-sectional, and plantwide levels. By supporting industry standards in computing such as Unix, Windows, and SQL, it makes interfacing with the surrounding world of computers easy. The system features a real-time process database that is distributed among the process controllers of the system to avoid redundancy in data storage, data inconsistency, and to


Plant manager
Order handler
Production manager
Financial manager
Purchaser
Lab assistant
Plant network
Process operator
Lab computer
Admin. computer
Information station
Operator station
Control network
Process station
Process station
Process station
Field bus
Process

Information-Handling Capabilities

Advant OCS offers basic ready-to-use information management functions such as historical data storage and playback, a versatile report generator, and a supplementary calculation package. It also offers open interfaces to third-party applications and to other computers in the plant. The historical data-storage and -retrieval service enables users to collect data from any system station at specified intervals, on command or on occurrence of specified events, performs a wide range of calculations on this data, and stores the results in so-called logs. Such logs can be accessed for presentation on any operator station or be used by applications on information stations or on external stations for a wide range of purposes. A report generator makes it possible to collect data for reports from the process database, from other reports, or the historical database. Output can be generated at specified times, on occurrence of specified events, or on request by an operator or software application. Unix- or Windows-based application programming interfaces offer a wide range of system services that give programmers a head start and safeguard engineering quality. Applications developed on this basis can be installed on the information management stations of the system, that is, close enough to the process to offer real-time performance.

The Future

Based on the dramatic developments in the past, it is tempting to speculate about the future. There are four areas that are important for the development of computer process control.

- Process knowledge
- Measurement technology
- Computer technology
- Control theory

Knowledge about process control and process dynamics is increasing slowly but steadily. The possibilities of learning about process characteristics are increasing substantially with the installation of process-control systems because it is then easy to collect data, perform experiments, and analyze the results. Progress in system identification and data analysis has also provided valuable information.

Progress in measurement technology is hard to predict. Many things can be done using existing techniques. The possibility of combining outputs of several different sensors with mathematical models is interesting. It is also possible to obtain automatic calibration with a computer. The advent of new sensors will, however, always offer new possibilities.

Spectacular developments are expected in computer technology with the introduction of VLSI. The ratio of price to performance will continue to drop substantially. The future microcomputers are expected to have computing power greater than the large mainframes of today. Substantial improvements are also expected in display techniques and in communications.

Programming has so far been one of the bottlenecks. There were only marginal improvements in productivity in programming from 1950 to 1970. At the end of the 1970s, many computer-controlled systems were still programmed in assembler code. In the computer-control field, it has been customary to overcome some of the programming problems by providing table-driven software. A user of a DDC, system is thus provided with a so-called DDC package that allows the user to generate a DDC system simply by filling in a table, so very little effort is needed to generate a system. The widespread use of packages hampers development, however, because it is very easy to use DDC, but it is a major effort to do something else. So only the well-proven methods are tried.

Control theory has made substantial progress since 1955. Only some of this theory, however, has made its way into existing computer-controlled systems, even though feasibility studies have indicated that significant improvements can be made. Model predictive control and adaptive control are some of the theoretical areas that are being applied in the industry today. To use these theories, it is necessary to fully understand the basic concepts of computer control. One reason for not using more complex digital controllers is the cost of programming. As already mentioned, it requires little effort to use a package provided by a vendor. It is, however, a major effort to try to do something else. Several signs show that this situation can be expected to change. Personal computers with interactive high-level languages are starting to be used for process control. With an interactive language, it is very easy to try new things. It is, however, unfortunately very difficult to write *safe* real-time control systems. This will change as better interactive systems become available.

Thus, there are many signs that point to interesting developments in the field of computer-controlled systems. A good way to be prepared is to learn the theory presented in this book.

1.3 Computer-Control Theory

Using computers to implement controllers has substantial advantages. Many of the difficulties with analog implementation can be avoided. For example, there are no problems with accuracy or drift of the components. It is very easy to have sophisticated calculations in the control law, and it is easy to include logic and nonlinear functions. Tables can be used to store data in order to accumulate knowledge about the properties of the system. It is also possible to have effective user interfaces.

A schematic diagram of a computer-controlled system is shown in Fig. 1.1. The system contains essentially five parts: the process, the A-D and D-A converters, the control algorithm, and the clock. Its operation is controlled by the clock. The times when the measured signals are converted to digital form are called the *sampling instants*; the time between successive samplings is called the *sampling period* and is denoted by h. Periodic sampling is normally used, but there are, of course, many other possibilities. For example, it is possible to sample when the output signals have changed by a certain amount. It is also possible to use different sampling periods for different loops in a system. This is called *multirate sampling.*

In this section we will give examples that illustrate the differences and the similarities of analog and computer-controlled systems. It will be shown that essential new phenomena that require theoretical attention do indeed occur.

Time Dependence

The presence of the the clock in Fig. 1.1 makes computer-controlled systems time-varying. Such systems can exhibit behavior that does not occur in linear time-invariant systems.

Example 1.1 Time dependence in digital filtering

A digital filter is a simple example of a computer-controlled system. Suppose that we want to implement a compensator that is simply a first-order lag. Such a compensator can be implemented using A-D conversion, a digital computer, and D-A

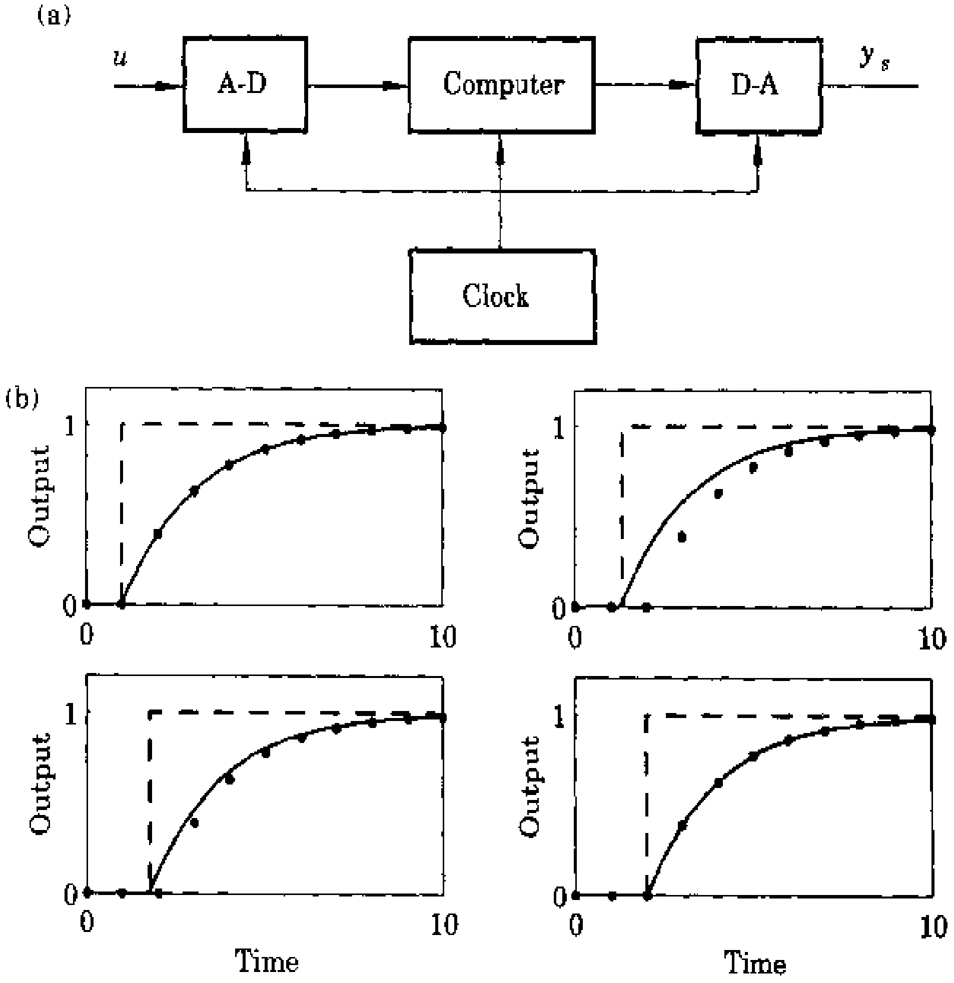


Figure 1.4 (a) Block diagram of a digital filter. (b) Step responses (dots) of a digital computer implementation of a first-order lag for different delays in the input step (dashed) compared with the first sampling instant. For comparison the response of the corresponding continuous-time system (solid) is also shown.

conversion. The first-order differential equation is approximated by a first-order difference equation. The step response of such a system is shown in Fig. 1.4. The figure clearly shows that the sampled system is not time-invariant because the response depends on the time when the step occurs. If the input is delayed, then the output is delayed by the same amount only if the delay is a multiple of the sampling period. ∎

The phenomenon illustrated in Fig. 1.4 depends on the fact that the system is controlled by a clock (compare with Fig. 1.1). The response of the system to an external stimulus will then depend on how the external event is synchronized with the internal clock of the computer system.

Because sampling is often periodic, computer-controlled systems will often result in closed-loop systems that are linear *periodic systems*. The phenomenon shown in Fig. 1.4 is typical for such systems. Later we will illustrate other consequences of periodic sampling.

A Naive Approach to Computer-Controlled Systems

We may expect that a computer-controlled system behaves as a continuous-time system if the sampling period is sufficiently small. This is true under very reasonable assumptions. We will illustrate this with an example.

Example 1.2 Controlling the arm of a disk drive

A schematic diagram of a disk-drive assembly is shown in Fig. 1.5. Let J be the moment of inertia of the arm assembly. The dynamics relating the position y of the arm to the voltage u of the drive amplifier is approximately described by the transfer function

$$G(s) = \frac{k}{Js^2} \tag{1.1}$$

where k is a constant. The purpose of the control system is to control the position of the arm so that the head follows a given track and that it can be rapidly moved to a different track. It is easy to find the benefits of improved control. Better trackkeeping allows narrower tracks and higher packing density. A faster control system reduces the search time. In this example we will focus on the search problem, which is a typical servo problem. Let u_c be the command signal and denote Laplace transforms with capital letters. A simple servo controller can be described by

$$U(s) = \frac{bK}{a} U_c(s) - K\frac{s+b}{s+a} Y(s) \tag{1.2}$$

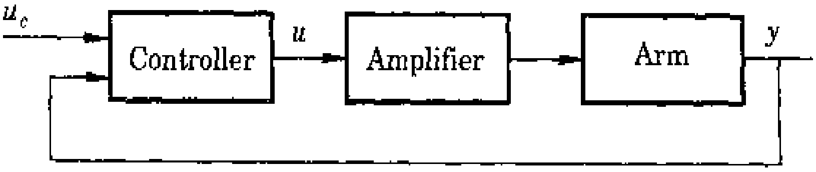


Figure 1.5 A system for controlling the position of the arm of a disk drive.

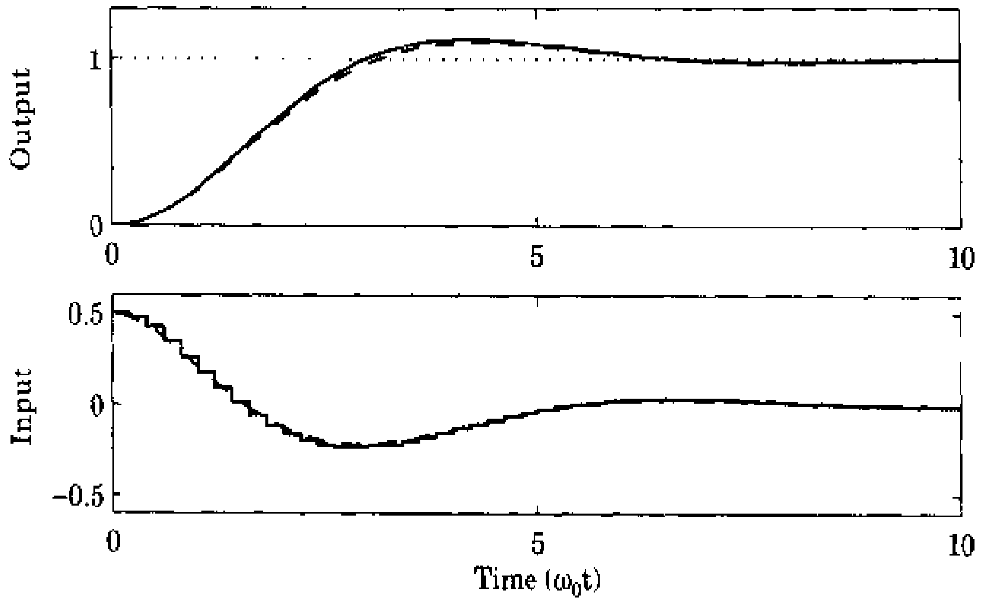


Figure 1.6 Simulation of the disk arm servo with analog (dashed) and computer control (solid). The sampling period is $h = 0.2/\omega_0$.

This controller is a two-degree-of-freedom controller where the feedback from the measured signal is simply a lead-lag filter. If the controller parameters are chosen as

$$\begin{aligned} a &= 2\omega_0 \\ b &= \omega_0/2 \\ K &= 2\frac{J\omega_0^2}{k} \end{aligned}$$

a closed system with the characteristic polynomial

$$P(s) = s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3$$

is obtained. This system has a reasonable behavior with a settling time to 5% of $5.52/\omega_0$. See Fig. 1.6. To obtain an algorithm for a computer-controlled system, the control law given by (1.2) is first written as

$$U(s) = \frac{bK}{a}U_c(s) - KY(s) + K\frac{a-b}{s+a}Y(s) = K\left(\frac{b}{a}U_c(s) - Y(s) + X(s)\right)$$

This control law can be written as

$$\begin{aligned} u(t) &= K\left(\frac{b}{a}u_c(t) - y(t) + x(t)\right) \\ \frac{dx}{dt} &= -ax + (a-b)y \end{aligned} \tag{1.3}$$

To obtain an algorithm for a control computer, the derivative dx/dt is approximated with a difference. This gives

$$\frac{x(t+h) - x(t)}{h} = -ax(t) + (a-b)y(t)$$

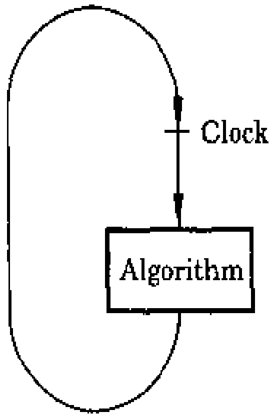


Figure 1.7 Scheduling a computer program.

The following approximation of the continuous algorithm (1.3) is then obtained:

$$\begin{aligned} u(t_k) &= K\left(\frac{b}{a}u_c(t_k) - y(t_k) + x(t_k)\right) \\ x(t_k + h) &= x(t_k) + h\Big((a-b)y(t_k) - ax(t_k)\Big) \end{aligned} \tag{1.4}$$

This control law should be executed at each sampling instant. This can be accomplished with the following computer program.

```
y: = adin(in2)      {read process value}
u:=K*(a/b*uc-y+x).
dout(u)             {output control signal}
newx:=x+h*((a-b)*y-a*x)
```

Arm position y is read from an analog input. Its desired value u_c is assumed to he given digitally. The algorithm has one state, variable x, which is updated at each sampling instant. The control law is computed and the value is converted to an analog signal. The program is executed periodically with period h by a scheduling program, as illustrated in Fig. 1.7. Because the approximation of the derivative by a difference is good if the interval h is small, we can expect the behavior of the computer-controlled system to be close to the continuous-time system. This is illustrated in Fig. 1.6, which shows the arm positions and the control signals for the systems with $h = 0.2/\omega_0$. Notice that the control signal for the computer-controlled system is constant between the sampling instants. Also notice that the difference between the outputs of the systems is very small. The computer-controlled system has slightly higher overshoot and the settling time to 5% is a little longer, $5.7/\omega_0$ instead of $5.5/\omega_0$. The difference between the systems decreases when the sampling period decreases. When the sampling period increases the computer-controlled system will, however, deteriorate. This is illustrated in Fig. 1.8, which shows the behavior of the system for the sampling periods $h = 0.5/\omega_0$ and $h = 1.08/\omega_0$. The response is quite reasonable for short sampling periods, but the system becomes unstable for long sampling periods. ∎

We have thus shown that it is straightforward to obtain an algorithm for computer control simply by writing the continuous-time control law as a differential equation and approximating the derivatives by differences. The example indi-

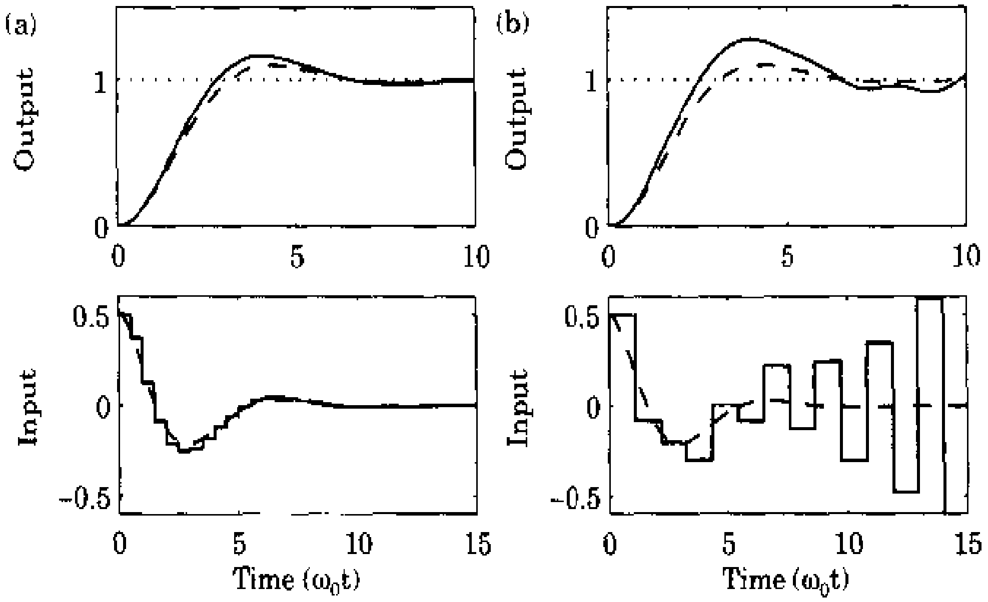


Figure 1.8 Simulation of the disk arm servo with computer control having sampling rates (a) $h = 0.5/\omega_0$ and (b) $h = 1.08/\omega_0$. For comparison, the signals for analog control are shown with dashed lines.

cated that the procedure seemed to work well if the sampling period was sufficiently small. The overshoot and the settling time are, however, a little larger for the computer-controlled system. This approach to design of computer-controlled systems will be discussed fully in the following chapters.

Deadbeat Control

Example 1.2 seems to indicate that a computer-controlled system will be inferior to a continuous-time example. We will now show that this is not necessarily the case. The periodic nature of the control actions can be actually used to obtain control strategies with superior performance.

Example 1.3 Disk drive with deadbeat control

Consider the disk drive in the previous example. Figure 1.9 shows the behavior of a computer-controlled system with a very long sampling interval $h = 1.4/\omega_0$. For comparison we have also shown the arm position, its velocity, and the control signal for the continuous controller used in Example 1.2. Notice the excellent behavior of the computer-controlled system. It settles much quicker than the continuous-time system even if control signals of the same magnitude are used. The 5% settling time is $2.34/\omega_0$, which is much shorter than the settling time $5.5/\omega_0$ of the continuous system. The output also reaches the desired position without overshoot and it remains constant when it has achieved its desired value, which happens in finite time. This behavior cannot be obtained with continuous-time systems because the solutions to such systems are sums of functions that are products of polynomials and exponential functions. The behavior obtained can be also described in the following way: The arm acceleratss with constant acceleration until is is halfway to the desired position and it then decelerates with constant retardation. The control

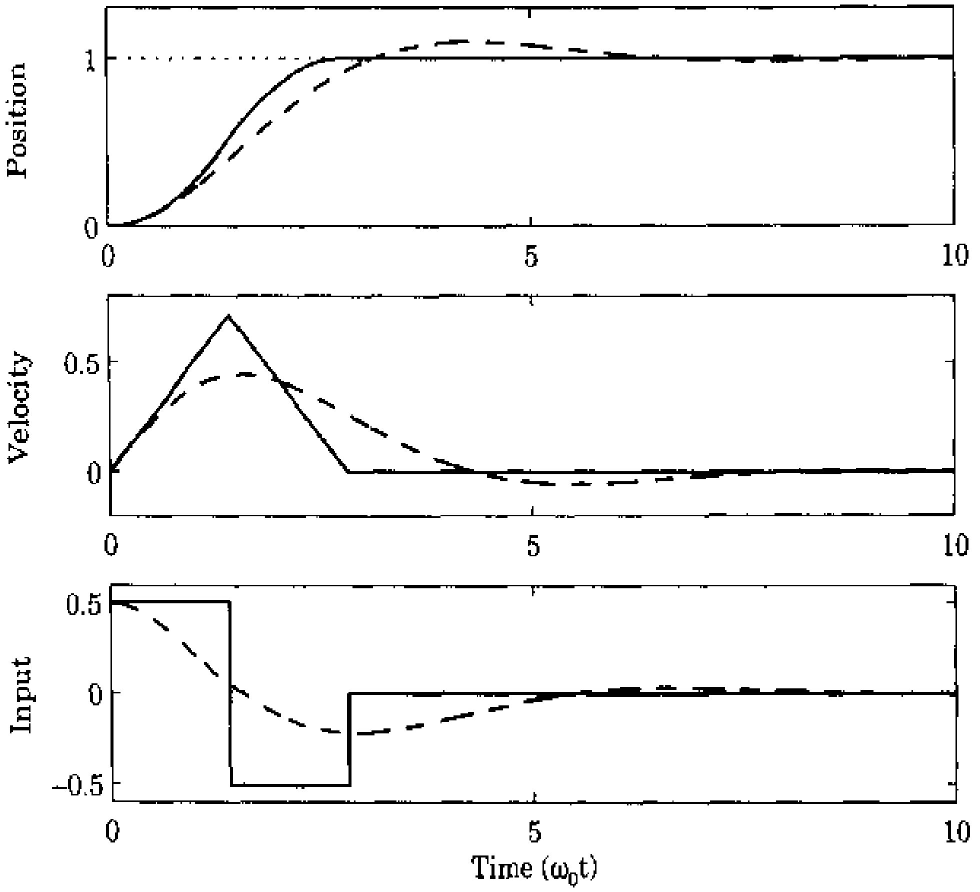


Figure 1.9 Simulation of the disk arm servo with deadbeat control (solid). The sampling period is $h = 1.4/\omega_0$. The analog controller from Example 1.2 is also shown (dashed).

strategy used has the same form as the control strategy in Example 1.2, that is,

$$u(t_k) = t_0 u_c(t_k) + t_1 u_c(t_{k-1}) - s_0 y(t_k) - s_1 y(t_{k-1}) - r_1 u(t_{k-1}) \tag{1.5}$$

The parameter values are different. When controlling the disk drive, the system can be implemented in such a way that sampling is initiated when the command signal is changed. In this way it is possible to avoid the extra time delay that occurs due to the lack of synchronization of sampling and command signal changes illustrated in Fig. 1.4. ■

The example shows that control strategies with different behavior can be obtained with computer control. In the particular example the response time can be reduced by a factor of 2. The control strategy in Example 1.3 is called *deadbeat control* because the system is at rest when the desired position is reached. Such a control scheme cannot be obtained with a continuous-time controller.

Aliasing

One property of the time-varying nature of computer-controlled systems was illustrated in Fig. 1.4. We will now illustrate another property that has far-reaching consequences. Stable linear time-invariant systems have the property

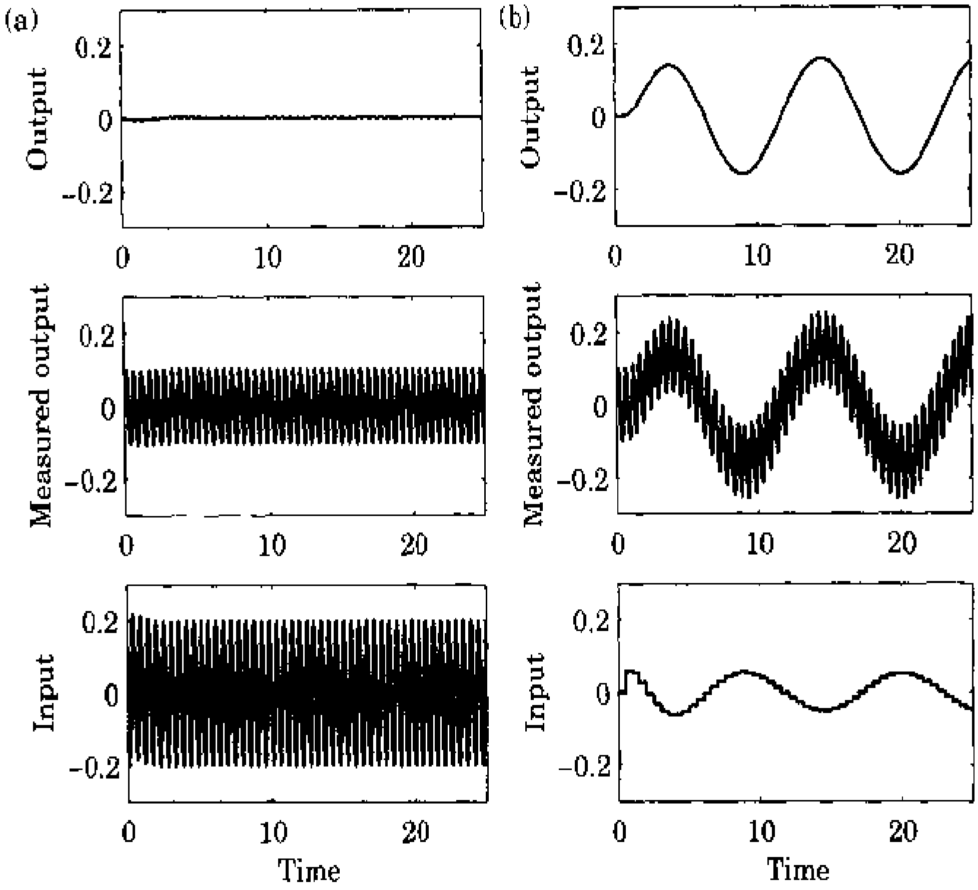


Figure 1.10 Simulation of the disk arm servo with analog and computer control. The frequency ω_0 is 1, the sampling period is $h = 0.5$, and there is a measurement noise $n = 0.1 \sin 12t$. (a) Continuous-time system; (b) sampled-data system.

that the steady-state response to sinusoidal excitations is sinusoidal with the frequency of the excitation signal. It will be shown that computer-controlled systems behave in a much more complicated way because sampling will create signals with new frequencies. This can drastically deteriorate performance if proper precautions are not taken.

Example 1.4 Sampling creates new frequencies

Consider the systems for control of the disk drive arm discussed in Example 1.2. Assume that the frequency ω_0 is 1 rad/s, let the sampling period be $h = 0.5/\omega_0$, and assume that there is a sinusoidal measurement noise with amplitude 0.1 and frequency 12 rad/s. Figure 1.10 shows interesting variables for the continuous-time system and the computer-controlled system. There is clearly a drastic difference between the systems. For the continuous-time system, the measurement noise has very little influence on the arm position. It does, however, create substantial control action with the frequency of the measurement noise. The high-frequency measurement noise is not noticeable in the control signal for the computer-controlled system, but there is also a substantial low-frequency component.

To understand what happens, we can consider Fig. 1.11, which shows the control signal and the measured signal on an expanded scale. The figure shows

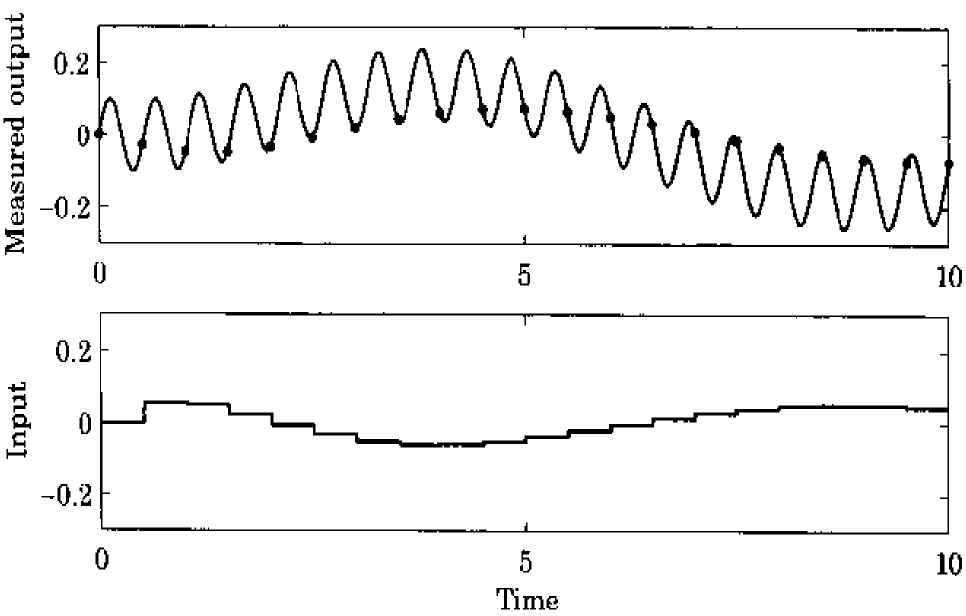


Figure 1.11 Simulation of the disk arm servo with computer control. The frequency ω_0 is 1, the sampling period is $h = 0.5$, and there is a measurement noise $n = 0.1 \sin 12t$.

that there is a considerable variation of the measured signal over the sampling period and the low-frequency variation is obtained by sampling the high-frequency signal at a slow rate. ■

We have thus made the striking observation that sampling creates signals with new frequencies. This is clearly a phenomenon that we must understand in order to deal with computer-controlled systems. At this stage we do not wish to go into the details of the theory; let it suffice to mention that sampling of a signal with frequency ω creates signal components with frequencies

$$\omega_{\text{sampled}} = n\omega_s \pm \omega \tag{1.6}$$

where $\omega_s = 2\pi/h$ is the sampling frequency, and n is an arbitrary integer. Sampling thus creates new frequencies. This is further discussed in Sec. 7.4.

In the particular example we have $\omega_s = 4\pi = 12.57$, and the measurement signal has the frequency 12 rad/s. In this case we find that sampling creates a signal component with the frequency 0.57 rad/s. The period of this signal is thus 11 s. This is the low-frequency component that is clearly visible in Fig. 1.11.

Example 1.4 illustrated that lower frequencies can be created by sampling. It follows from (1.6) that sampling also can give frequencies that are higher than the excitation frequency. This is illustrated in the following example.

Example 1.5 Creation of higher frequencies by sampling

Figure 1.12 shows what can happen when a sinusoidal signal of frequency 4.9 Hz is applied to the system in Example 1.1, which has a sampling period of 10 Hz. It follows from Eq. (1.6) that a signal component with frequency 5.1 Hz is created by sampling. This signal interacts with the original signal with frequency 4.9 Hz to give the beating of 0.1 Hz shown in the figure. ■

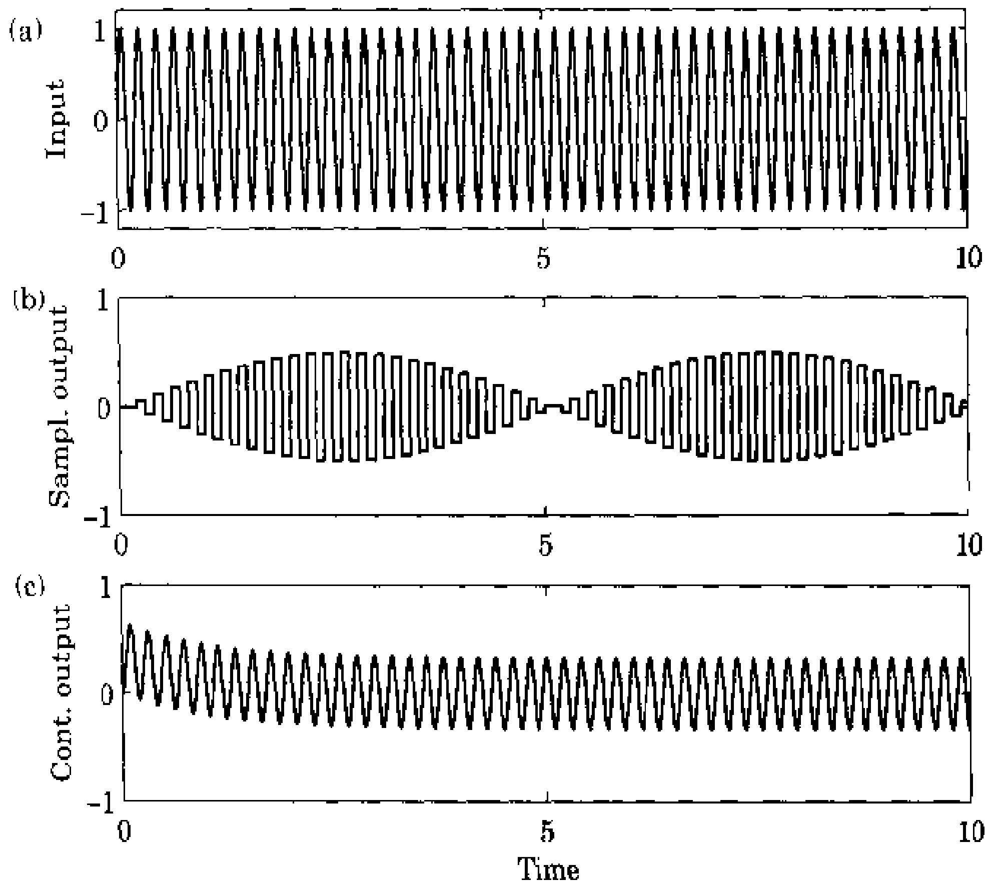


Figure 1.12 Sinusoidal excitation of the sampled system in Example 1.5. (a) Input sinusoidal with frequency 4.9 Hz. (b) Sampled-system output. The sampling period is 0.1 s. (c) Output of the corresponding continuous-time system.

There are many aspects of sampled systems that indeed can be understood by linear time-invariant theory. The examples given indicate, however, that the sampled systems cannot be fully understood within that framework. It is thus useful to have other tools for analysis.

The phenomenon that the sampling process creates new frequency components is called *aliasing*. A consequence of Eq. (1.6) is that there will be low-frequency components created whenever the sampled signal contains frequencies that are larger than half the sampling frequency. The frequency $\omega_N = \omega_s/2$ is called the *Nyquist frequency* and is an important parameter of a sampled system.

Presampling Filters or Antialiasing Filters

To avoid the difficulties illustrated in Fig. 1.10, it is essential that all signal components with frequencies higher than the Nyquist frequency are removed before a signal is sampled. By doing this the signals sampled will not change much over a sampling interval and the difficulties illustrated in the previous examples are avoided. The filters that reduce the high-frequency components of the signals are called *antialiasing filters*. These filters are an important component of computer-controlled systems. The proper selection of sampling periods and antialiasing filters are important aspects of the design of computer-controlled systems.

Difference Equations

Although a computer-controlled system may have a quite complex behavior, it is very easy to describe the behavior of the system at the sampling instants. We will illustrate this by analyzing the disk drive with a deadbeat controller.

Example 1.6 Difference equations

The input-output properties of the process Eq. (1.1) can be described by

$$y(t_k) - 2y(t_{k-1}) + y(t_{k-2}) = \frac{kh^2}{2J}\Big(u(t_{k-1}) + u(t_{k-2})\Big) \tag{1.7}$$

This equation is exact if the control signal is constant over the sampling intervals. The deadbeat control strategy is given by Eq. (1.5) and the closed-loop system thus can be described by the equations.

$$\begin{aligned} y(t_k) - 2y(t_{k-1}) + y(t_{k-2}) &= \alpha\Big(u(t_{k-1}) + u(t_{k-2})\Big) \\ u(t_{k-1}) + r_1 u(t_{k-2}) &= t_0 u_c(t_{k-1}) - s_0 y(t_{k-1}) - s_1 y(t_{k-2}) \end{aligned} \tag{1.8}$$

where $\alpha = kh^2/2J$. Elimination of the control signal u between these equations gives

$$\begin{aligned} &y(t_k) + (r_1 - 2 + \alpha s_0)y(t_{k-1}) + \Big(1 - 2r_1 + \alpha(s_0 + s_1)\Big)y(t_{k-2}) + (r_1 + \alpha s_1)y(t_{k-3}) \\ &\quad = \frac{\alpha t_0}{2}\Big(u_c(t_{k-1}) + u_c(t_{k-2})\Big) \end{aligned}$$

The parameters of the deadbeat controller are given by

$$\begin{aligned} r_1 &= 0.75 \\ s_0 &= \frac{1.25}{\alpha} = \frac{2.5J}{kh^2} \\ s_1 &= -\frac{0.75}{\alpha} = -\frac{1.5J}{kh^2} \\ t_0 &= \frac{1}{4\alpha} = \frac{1}{2} \end{aligned}$$

With these parameters the closed-loop system becomes

$$y(t_k) = \frac{1}{2}\Big(u_c(t_{k-1}) + u_c(t_{k-2})\Big)$$

It follows from this equation that the output is the average value of the past two values of the command signal. Compare with Fig. 1.9. ∎

The example illustrates that the behavior of the computer-controlled system at the sampling instants is described by a linear difference equation. This observation is true for general linear systems. Difference equations, therefore, will be a key element of the theory of computer-controlled systems, they play the same role as differential equations for continuous systems, and they will give the values of the important system variables at the sampling instants. If we are satisfied by this knowledge, it is possible to develop a simple theory for analysis and design of sampled systems. To have a more complete knowledge of the behavior of the systems, we must also analyse the behavior between the sampling instants and make sure that the system variables do not change too much over a sampling period.

Is There a Need for a Theory for Computer-Controlled Systems?

The examples in this section have demonstrated that computer-controlled systems can be designed simply by using continuous-time theory and approximating the differential equations describing the controllers by difference equations. The examples also have shown that computer-controlled systems have the potential of giving control schemes, such as the deadbeat strategy, with behavior that cannot be obtained by continuous-time systems. It also has been demonstrated that sampling can create phenomena that are not found in linear time-invariant systems. It also has been demonstrated that the selection of the sampling period is important and that it is necessary to use antialiasing filters. These issues clearly indicate the need for a theory for computer-controlled systems.

1.4 Inherently Sampled Systems

Sampled models are natural descriptions for many phenomena. The theory of sampled-data systems, therefore, has many applications outside the field of computer control.

Sampling due to the Measurement System

In many cases, sampling will occur naturally in connection with the measurement procedure. A few examples follow.

Example 1.7 Radar

When a radar antenna rotates, information about range and direction is naturally obtained once per revolution of the antenna. A sampled model is thus the natural way to describe a radar system. Attempts to describe radar systems were, in fact, one of the starting points of the theory of sampled systems. ■

Example 1.8 Analytical instruments

In process-control systems, there are many variables that cannot be measured on-line, so a sample of the product is analyzed off-line in an analytical instrument such as a mass spectrograph or a chromatograph. ■

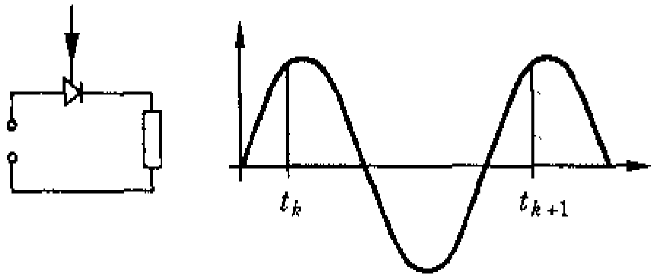


Figure 1.13 Thyristor control circuit.

Example 1.9 Economic systems

Accounting procedures in economic systems are often tied to the calendar. Although transactions may occur at any time, information about important variables is accumulated only at certain times—for example, daily, weekly, monthly, quarterly, or yearly. ■

Example 1.10 Magnetic flow meters

A magnetic flow meter is based on the principle that current that moves in a magnetic field generates a voltage. In a typical meter a magnetic field is generated across the pipe and the voltage is measured in a direction orthogonal to the field. To compensate for electrolytic voltages that often are present, it is common to use a pulsed operation in which the field is switched on and off periodically. This switching causes an inherent sampling. ■

Sampling due to Pulsed Operation

Many systems are inherently sampled because information is transmitted using pulsed information. Electronic circuits are a prototype example. They were also one source of inspiration for the development of sampled-data theory. Other examples follow.

Example 1.11 Thyristor control

Power electronics using thyristors are sampled systems. Consider the circuit in Fig. 1.13. The current can be switched on only when the voltage is positive. When the current is switched on, it remains on until the current has a zero crossing. The current is thus synchronized to the periodicity of the power supply. The variation of the ingition time will cause the sampling period to vary, which must be taken care of when making models for thyristor circuits. ■

Example 1.12 Biological systems

Biological systems are fundamentally sampled because the signal transmission in the nervous system is in the form of pulses. ■

Example 1.13 Internal-combustion engines

An internal-combustion engine is a sampled system. The ignition can be viewed as a clock that synchronizes the operation of the engine. A torque pulse is generated at each ignition. ■

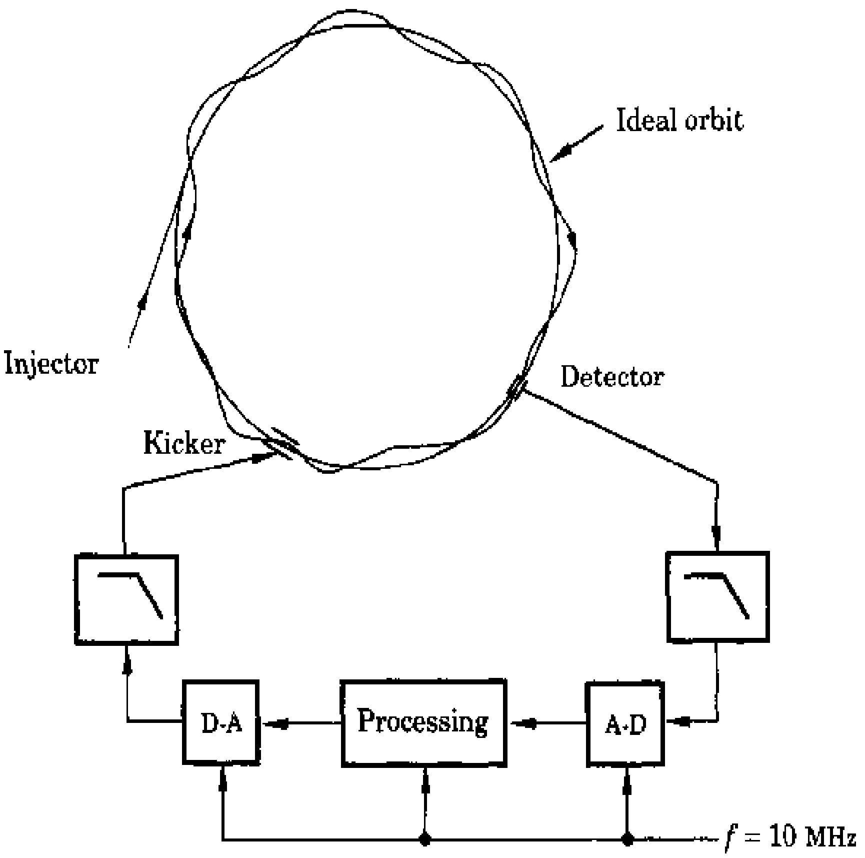


Figure 1.14 Particle accelerator with stochastic cooling.

Example 1.14 Particle accelerators

Particle accelerators are the key experimental tool in particle physics. The Dutch engineer Simnon van der Meer made a major improvement in accelerators by introducing feedback to control particle paths, which made it possible to increase the beam intensity and to improve the beam quality substantially. The method, which is called stochastic cooling, was a key factor in the successful experiments at CERN. As a result van der Meer shared the 1984 Nobel Prize in Physics with Carlo Rubbia.

A schematic diagram of the system is shown in Fig. 1.14. The particles enter into a circular orbit via the injector. The particles are picked up by a detector at a fixed position and the energy of the particles is increased by the kicker, which is located at a fixed position. The system is inherently sampled because the particles are only observed when they pass the detector and control only acts when they pass the kicker.

From the point of view of sampled systems, it is interesting to observe that there is inherent sampling both in sensing and actuation. ■

The systems in these examples are periodic because of their pulsed operation. Periodic systems are quite difficult to handle, but they can be considerably simplified by studying the systems at instants synchronized with the pulses—that is, by using sampled-data models. The processes then can be described as time-invariant discrete-time systems at the sampling instants. Examples 1.11 and 1.13 are of this type.

1.5 How Theory Developed

Although the major applications of the theory of sampled systems are currently in computer control, many of the problems were encountered earlier. In this section some of the main ideas in the development of the theory are discussed. Many of the ideas are extensions of the ideas for continuous-time systems.

The Sampling Theorem

Because all computer-controlled systems operate on values of the process variables at discrete times only, it is very important to know the conditions under which a signal can be recovered from its values in discrete points only. The key issue was explored by Nyquist, who showed that to recover a sinusoidal signal from its samples, it is necessary to sample at least twice per period. A complete solution was given in an important work by Shannon in 1949. This is very fundamental for the understanding of some of the phenomena occuring in discrete-time systems.

Difference Equations

The first germs of a theory for sampled systems appeared in connection with analyses of specific control systems. The behavior of the chopper-bar galvanometer, investigated in Oldenburg and Sartorius (1948), was one of the earliest contributions to the theory. It was shown that many properties could be understood by analyzing a linear time-invariant difference equation. The difference equation replaced the differential equations in continuous-time theory. For example, stability could be investigated by the Schur-Cohn method, which is equivalent to the Routh-Hurwitz criterion.

Numerical Analysis

The theory of sampled-data analysis is closely related to numerical analysis. Integrals are evaluated numerically by approximating them with sums. Many optimization problems can be described in terms of difference equations. Ordinary differential equations are integrated by approximating them by difference equations. For instance, step-length adjustment in integration routines can be regarded as a sampled-data control problem. A large body of theory is available that is related to computer-controlled systems. Difference equations are an important element of this theory, too.

Transform Methods

During and after World War II, a lot of activity was devoted to analysis of radar systems. These systems are naturally sampled because a position measurement is obtained once per antenna revolution. One problem was to find ways to describe these new systems. Because transform theory had been so useful for continuous-time systems, it was natural to try to develop a similar theory for sampled systems. The first steps in this direction were taken by Hurewicz (1947). He introduced the transform of a sequence $f(kh)$, defined by

$$Z\{f(kh)\} = \sum_{k=0}^{\infty} z^{-k} f(kh)$$

This transform is similar to the *generating function*, which had been used so successfully in many branches of applied mathematics. The transform was later defined as the *z-transform* by Ragazzini and Zadeh (1952). Transform theory was developed independently in the Soviet Union, in the United States, and in Great Britain. Tsypkin (1949) and Tsypkin (1950) called the transform the *discrete Laplace transform* and developed a systematic theory for pulse-controlled systems based on the transform. The transform method was also independently developed by Barker (1952) in England.

In the United States the transform was further developed in a Ph.D. dissertation by Jury at Columbia University. Jury developed tools both for analysis and design. He also showed that sampled systems could be better than their continuous-time equivalents. (See Example 1.3 in Sec. 1.3.) Jury also emphasized that it was possible to obtain a closed-loop system that exactly achieved steady state in finite time. In later works he also showed that sampling can cause cancellation of poles and zeros. A closer investigation of this property later gave rise to the notions of observability and reachability.

The z-transform theory leads to comparatively simple results. A limitation of the theory, however, is that it tells what happens to the system only at the sampling instants. The behavior between the sampling instants is not just an academic question, because it was found that systems could exhibit *hidden oscillations*. These oscillations are zero at the sampling instants, but very noticeable in between.

Another approach to the theory of sampled system was taken by Linvill (1951). Following ideas due to MacColl (1945), he viewed the sampling as an amplitude modulation. Using a describing-function approach, Linvill effectively described intersample behavior. Yet another approach to the analysis of the problem was the *delayed z-transform*, which was developed by Tsypkin in 1950, Barker in 1951, and Jury in 1956. It is also known as the *modified z-transform*.

Much of the development of the theory was done by a group at Columbia University led by John Ragazzini. Jury, Kalman, Bertram, Zadeh, Franklin, Friedland, Kranc, Freeman, Sarachik, and Sklansky all did their Ph.D. work for Ragazzini.

Toward the end of the 1950s, the z-transform approach to sampled systems had matured, and several textbooks appeared almost simultaneously: Jury (1958), Ragazzini and Franklin (1958), Tsypkin (1958), and Tou (1959). This theory, which was patterned after the theory of linear time-invariant continuous-time systems, gave good tools for analysis and synthesis of sampled systems. A few modifications had to be made because of the time-varying nature of sampled systems. For example, all operations in a block-diagram representation do not commute!

State-Space Theory

A very important event in the late 1950s was the development of state-space theory. The major inspiration came from mathematics and the theory of ordinary differential equations and from mathematicians such as Lefschetz, Pontryagin, and Bellman. Kalman deserves major credit for the state-space approach to control theory. He formulated many of the basic concepts and solved many of the important problems.

Several of the fundamental concepts grew out of an analysis of the problem of whether it would be possible to get systems in which the variables achieved steady state in finite time. The analysis of this problem led to the notions of reachability and observability. Kalman's work also led to a much simpler formulation of the analysis of sampled systems: The basic equations could be derived simply by starting with the differential equations and integrating them under the assumption that the control signal is constant over the sampling period. The discrete-time representation is then obtained by only considering the system at the sampling points. This leads to a very simple state-space representation of sampled-data systems.

Optimal and Stochastic Control

There were also several other important developments in the late 1950s. Bellman (1957) and Pontryagin et al. (1962) showed that many design problems could be formulated as optimization problems. For nonlinear systems this led to nonclassical calculus of variations. An explicit solution was given for linear systems with quadratic loss functions by Bellman, Glicksberg, and Gross (1958). Kalman (1960a) showed in a celebrated paper that the linear quadratic problem could be reduced to a solution of a Riccati equation. Kalman also showed that the classical Wiener filtering problem could be reformulated in the state-space framework. This permitted a "solution" in terms of recursive equations, which were very well suited to computer calculation.

In the beginning of the 1960s, a stochastic variational problem was formulated by assuming that disturbances were random processes. The optimal control problem for linear systems could be formulated and solved for the case of quadratic loss functions. This led to the development of *stochastic control theory*. The work resulted in the so-called *Linear Quadratic Gaussian (LQG)* theory. This is now a major design tool for multivariable linear systems.

Algebraic System Theory

The fundamental problems of linear system theory were reconsidered at the end of the 1960s and the beginning of the 1970s. The algebraic character of the problems was reestablished, which resulted in a better understanding of the foundations of linear system theory. Techniques to solve specific problems using polynomial methods were another result [see Kalman, Falb, and Arbib (1969), Rosenbrock (1970), Wonham (1974), Kučera (1979, 1991), and Blomberg and Ylinen (1983)].

System Identification

All techniques for analysis and design of control systems are based on the availability of appropriate models for process dynamics. The success of classical control theory that almost exclusively builds on Laplace transforms was largely due to the fact that the transfer function of a process can be determined experimentally using frequency response. The development of digital control was accompanied by a similar development of system identification methods. These allow experimental determination of the pulse-transfer function or the difference equations that are the starting point of analysis and design of digital control systems. Good sources of information on these techniques are Åström and Eykhoff (1971), Norton (1986), Ljung (1987), Söderström and Stoica (1989), and Johansson (1993).

Adaptive Control

When digital computers are used to implement a controller, it is possible to implement more complicated control algorithms. A natural step is to include both parameter estimation methods and control design algorithms. In this way it is possible to obtain adaptive control algorithms that determine the mathematical models and perform control system design on-line. Research on adaptive control began in the mid-1950s. Significant progress was made in the 1970s when feasibility was demonstrated in industrial applications. The advent of the microprocessor made the algorithms cost-effective, and commercial adaptive regulators appeared in the early 1980s. This has stimulated vigorous research on theoretical issues and significant product development. See, for instance, Åström and Wittenmark (1973, 1980, 1995), Åström (1983b, 1987), and Goodwin and Sin (1984).

Automatic Tuning

Controller parameters are often tuned manually. Experience has shown that it is difficult to adjust more than two parameters manually. From the user point of view it is therefore helpful to have tuning tools built into the controllers. Such systems are similar to adaptive controllers. They are, however, easier to design and use. With computer-based controllers it is easy to incorporate tuning tools. Such systems also started to appear industrially in the mid-1980s. See Åström and Hägglund (1995).

1.6 Notes and References

To acquire mature knowledge about a field it is useful to know its history and to read some of the original papers. Jury and Tsypkin (1971), and Jury (1980), written by two of the originators of sampled-data theory, give a useful perspective. Early work on sampled systems is found in MacColl (1945), Hurewicz

(1947), and Oldenburg and Sartorius (1948). The sampling theorem was given in Kotelnikov (1933) and Shannon (1949).

Major contributions to the early theory of sampled-data systems were obtained in England by Lawden (1951) and Barker (1952); in the United States by Linvill (1951), Ragazzini and Zadeh (1952), and Jury (1956); and in the Soviet Union by Tsypkin (1949) and Tsypkin (1950).The first textbooks on sampled-data theory appeared toward the end of the 1950s. They were Jury (1958), Ragazzini and Franklin (1958), Tsypkin (1958), and Tou (1959). A large number of textbooks have appeared since then. Among the more common ones we can mention Ackermann (1972, 1996), Kuo (1980), Franklin and Powell (1989), and Isermann (1989, 1991).

The idea of formulating control problems in the state space also resulted in a reformulation of sampled-data theory. Kalman (1961) is seminal.

Some fundamental references on optimal and stochastic control are Bellman (1957), Bellman, Glicksberg, and Gross (1958), Kalman (1960a), Pontryagin et al. (1962), and Åström (1970). The algebraic system approach is discussed in Kalman, Falb, and Arbib (1969), Rosenbrock (1970), Wonham (1974), Kučera (1979, 1991, 1993), and Blomberg and Ylinen (1983).

System identification is surveyed in Åström and Eykhoff (1971), Ljung and Söderström (1983), Norton (1986), Ljung (1987), Söderström and Stoica (1989), and Johansson (1993). Adaptive control is discussed in Bellman (1961), Åström and Wittenmark (1973, 1980, 1995), Åström (1983b, 1987), Goodwin and Sin (1984), Gupta (1986), and Åström and Hägglund (1995).

A survey of distributed computer systems is found in Lucas (1986). In Gustafsson, Lundh, and Söderlind (1988), it is shown how step-length control in numerical integration can be regarded as a control problem. This is also discussed in Hairer and Wanner (1991).

Many additional references are given in the following sections. We also recommend the proceedings of the IFAC Symposia on Digital Computer Applications to Process Control and on Identification and System Parameter Estimation, which are published by Pergamon Press.