Chapter 5

Industrial Control Systems

CHAPTER CONTENTS

The control system is one of the three basic components of an automated system (Section 4.1). In this chapter, we examine industrial control systems, in particular how digital computers are used to implement the control function in production. *Industrial control* is defined here as the automatic regulation of unit operations and their associated equipment as well as the integration and coordination of the unit operations into the larger production system. In the context of our book, the term *unit operations* usually refers to manufacturing operations; however, the term also applies to the operation of material handling and other industrial equipment. Let us begin our chapter by comparing the application of industrial control in the processing industries with its application in the discrete manufacturing industries.

PROCESS INDUSTRIES VERSUS DISCRETE MANUFACTURING INDUSTRIES

In our previous discussion of industry types in Chapter 2, we divided industries and their production operations into two basic categories: (1) process industries and (2) discrete manufacturing industries. Process industries perform their production operations on *amounts* of materials, because the materials tend to be liquids, gases, powders, and similar materials, whereas discrete manufacturing industries perform their operations on *quantities* of materials, because the materials tend to be discrete parts and products. The kinds of unit operations performed on the materials are different in the two industry categories. Some of the typical unit operations in each category are listed in Table 5.1.

5.1.1 Levels of Automation in the Two Industries

The levels of automation (Section 4.3) in the two industries are compared in Table 5.2. The significant differences are seen in the low and intermediate levels. At the device level, there are differences in the types of actuators and sensors used in the two industry categories, simply because the processes and equipment are different. In the process industries, the devices are used mostly for the control loops in chemical, thermal, or similar processing operations, whereas in discrete manufacturing, the devices control the mechanical actions of machines. At the next level up, the difference is that unit operations are controlled in the process industries, and machines are controlled in the discrete manufacturing operations. At the third level, the difference is between control of interconnected unit processing operations and interconnected machines. At the upper levels (plant and enterprise), the control issues are similar, allowing for the fact that the products and processes are different.

5.1.2 Variables and Parameters in the Two Industries

The distinction between process industries and discrete manufacturing industries extends to the variables and parameters that characterize the respective production operations. The reader will recall from the previous chapter (Section 4.1.2) that we defined variables as outputs of the process and parameters as inputs to the process. In the process industries, the variables and parameters of interest tend to be continuous, whereas in discrete manufacturing, they tend to be discrete. Let us explain the differences with reference to Figure 5.1.

TABLE 5.1 Typical Unit Operations in the Process Industries and Discrete Manufacturing Industries

Typical Unit Operations in the Process Industries	*Typical Unit Operations in the Discrete Manufacturing Industries*
Chemical reactions	Casting
Comminution	Forging
Deposition (e.g., chemical vapor deposition)	Extrusion
Distillation	Machining
Mixing and blending of ingredients	Mechanical assembly
Separation of ingredients	Plastic molding
	Sheet metal stamping

TABLE 5.2 Levels of Automation in the Process Industries and Discrete Manufacturing Industries

Level	Level of Automation in the Process Industries	Level of Automation in the Discrete Manufacturing Industries
5	*Corporate level*—management information system, strategic planning, high-level management of enterprise	*Corporate level*—management information system, strategic planning, high-level management of enterprise
4	*Plant level*—scheduling, tracking materials, equipment monitoring	*Plant or factory level*—scheduling, tracking work-in-process, routing parts through machines, machine utilization
3	*Supervisory control level*—control and coordination of several interconnected unit operations that make up the total process	*Manufacturing cell or system level*—control and coordination of groups of machines and supporting equipment working in coordination, including material handling equipment
2	*Regulatory control level*—control of unit operations	*Machine level*—production machines and workstations for discrete part and product manufacture
1	*Device level*—sensors and actuators comprising the basic control loops for unit operations	*Device level*—sensors and actuators to accomplish control of machine actions

A *continuous variable* (or parameter) is one that is uninterrupted as time proceeds, at least during the manufacturing operation. A continuous variable is generally considered to be *analog*, which means it can take on any value within a certain range. The variable is not restricted to a discrete set of values. Production operations in both the process industries and discrete parts manufacturing are characterized by continuous variables. Examples include force, temperature, flow rate, pressure, and velocity. All of these variables

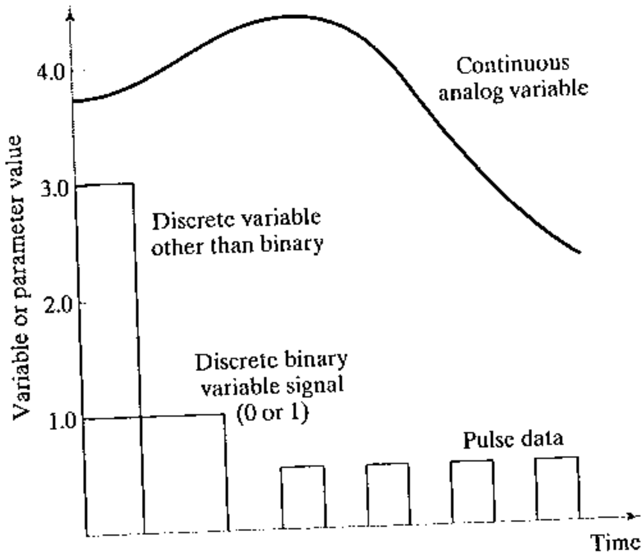


Figure 5.1 Continuous and discrete variables and parameters in manufacturing operations.

(whichever ones apply to a given production process) are continuous over time during the process, and they can take on any of an infinite number of possible values within a certain practical range.

A *discrete variable* (or parameter) is one that can take on only certain values within a given range. The most common type of discrete variable is *binary*, meaning it can take on either of two possible values, ON or OFF, open or closed, and so on. Examples of discrete binary variables and parameters in manufacturing include limit switch open or closed, motor on or off, and workpart present or not present in a fixture. Not all discrete variables (and parameters) are binary. Other possibilities are variables that can take on more than two possible values but less than an infinite number, that is, *discrete other than binary*. Examples include daily piece counts in a production operation and the display of a digital tachometer. A special form of discrete variable (and parameter) is *pulse data*, which consist of a train of pulses as shown in Figure 5.1. As a discrete variable, a pulse train might be used to indicate piece counts, for example, parts passing on a conveyor activate a photocell to produce a pulse for each part detected. As a process parameter, a pulse train might be used to drive a stepper motor.

5.2 CONTINUOUS VERSUS DISCRETE CONTROL

Industrial control systems used in the process industries have tended to emphasize the control of continuous variables and parameters. By contrast, the manufacturing industries produce discrete parts and products, and their controllers have tended to emphasize discrete variables and parameters. Just as we have two basic types of variables and parameters that characterize production operations, we also have two basic types of control: (1) *continuous control*, in which the variables and parameters are continuous and analog; and (2) *discrete control*, in which the variables and parameters are discrete, mostly binary discrete. Some of the differences between continuous control and discrete control are summarized in Table 5.3.

TABLE 5.3 Comparison Between Continuous Control and Discrete Control

Comparison Factor	*Continuous Control in Process Industries*	*Discrete Control in Discrete Manufacturing Industries*
Typical measures of product output	Weight measures, liquid volume measures, solid volume measures	Number of parts, number of products
Typical quality measures	Consistency, concentration of solution, absence of contaminants, conformance to specification	Dimensions, surface finish, appearance, absence of defects, product reliability
Typical variables and parameters	Temperature, volume flow rate, pressure	Position, velocity, acceleration, force
Typical sensors	Flow meters, thermocouples, pressure sensors	Limit switches, photoelectric sensors, strain gages, piezoelectric sensors
Typical actuators	Valves, heaters, pumps	Switches, motors, pistons
Typical process time constants	Seconds, minutes, hours	Less than a second

In reality, most operations in the process and discrete manufacturing industries include both continuous and discrete variables and parameters. Consequently, many industrial controllers are designed with the capability to receive, operate on, and transmit both types of signals and data. In Chapter 6, we discuss the various types of signals and data in industrial control systems and how the data are converted for use by digital computer controllers.

To complicate matters, since digital computers began replacing analog controllers in continuous process control applications around 1960, continuous process variables are no longer measured continuously. Instead, they are sampled periodically, in effect creating a discrete sampled-data system that approximates the actual continuous system. Similarly, the control signals sent to the process are typically stepwise functions that approximate the previous continuous control signals transmitted by analog controllers. Hence, in digital computer process control, even continuous variables and parameters possess characteristics of discrete data, and these characteristics must be considered in the design of the computer-process interface and the control algorithms used by the controller.

5.2.1 Continuous Control Systems

In continuous control, the usual objective is to maintain the value of an output variable at a desired level, similar to the operation of a feedback control system as defined in the previous chapter (Section 4.1.3). However, most continuous processes in the practical world consist of many separate feedback loops, all of which have to be controlled and coordinated to maintain the output variable at the desired value. Examples of continuous processes are the following:

- Control of the output of a chemical reaction that depends on temperature, pressure, and input flow rates of several reactants. All of these variables and/or parameters are continuous.
- Control of the position of a workpart relative to a cutting tool in a contour milling operation in which complex curved surfaces are generated. The position of the part is defined by x-, y-, and z-coordinate values. As the part moves, the x, y, and z values can be considered as continuous variables and/or parameters that change over time to machine the part.

There are several ways to achieve the control objective in a continuous process control system. In the following paragraphs, we survey the most prominent categories.

Regulatory Control. In regulatory control, the objective is to maintain process performance at a certain level or within a given tolerance band of that level. This is appropriate, for example, when the performance attribute is some measure of product quality, and it is important to keep the quality at the specified level or within a specified range. In many applications, the performance measure of the process, sometimes called the *index of performance*, must be calculated based on several output variables of the process. Except for this feature, regulatory control is to the overall process what feedback control is to an individual control loop in the process, as suggested by Figure 5.2.

The trouble with regulatory control (and also with a simple feedback control loop) is that compensating action is taken only after a disturbance has affected the process output. An error must be present for any control action to be taken. The presence of an error

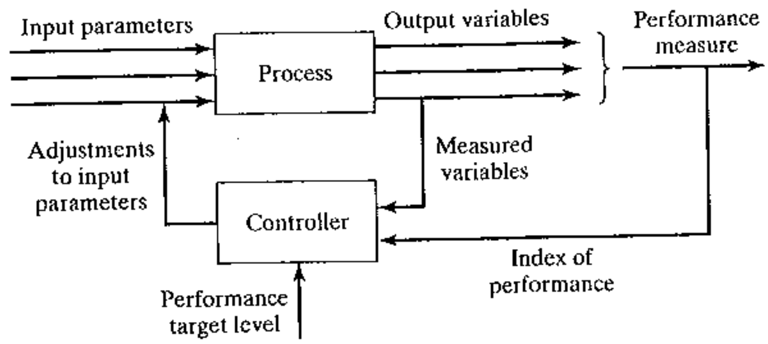


Figure 5.2 Regulatory control.

means that the output of the process is different from the desired value. The following control mode, feedforward control, addresses this issue.

Feedforward Control. The strategy in feedforward control is to anticipate the effect of disturbances that will upset the process by sensing them and compensating for them before they can affect the process. As shown in Figure 5.3, the feedforward control elements sense the presence of a disturbance and take corrective action by adjusting a process parameter that compensates for any effect the disturbance will have on the process. In the ideal case, the compensation is completely effective. However, complete compensation is unlikely because of imperfections in the feedback measurements, actuator operations, and control algorithms, so feedforward control is usually combined with feedback control, as shown in our figure. Regulatory and feedforward control are more closely associated with the process industries than with discrete product manufacturing.

Steady-State Optimization. This term refers to a class of optimization techniques in which the process exhibits the following characteristics: (1) there is a well-defined index of performance, such as product cost, production rate, or process yield; (2) the

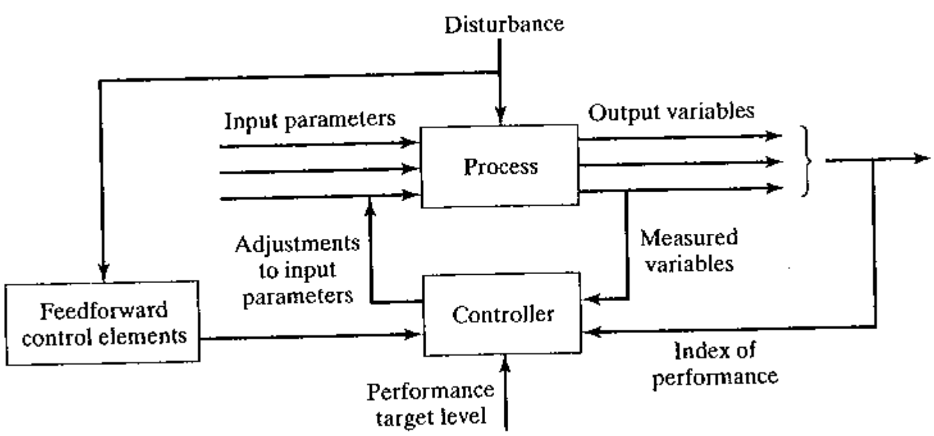


Figure 5.3 Feedforward control, combined with feedback control.

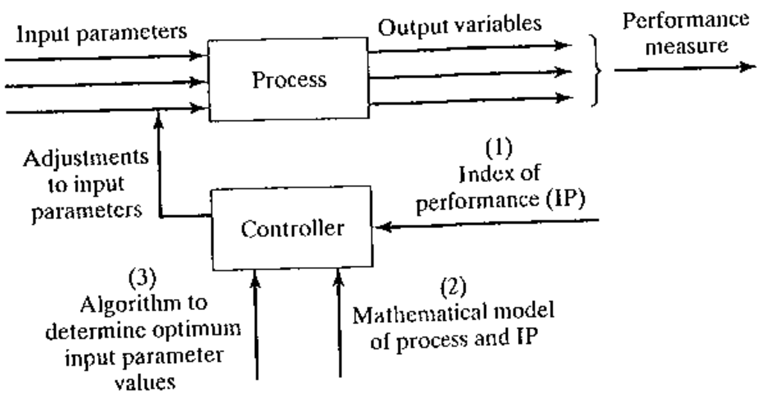


Figure 5.4 Steady-state (open loop) optimal control.

relationship between the process variables and the index of performance is known; and (3) the values of the system parameters that optimize the index of performance can be determined mathematically. When these characteristics apply, the control algorithm is designed to make adjustments in the process parameters to drive the process toward the optimal state. The control system is open loop, as seen in Figure 5.4. Several mathematical techniques are available for solving steady-state optimal control problems, including differential calculus, calculus of variations, and various mathematical programming methods.

Adaptive Control. Steady-state optimal control operates as an open loop system. It works successfully when there are no disturbances that invalidate the known relationship between process parameters and process performance. When such disturbances are present in the application, a self-correcting form of optimal control can be used, called adaptive control. Adaptive control combines feedback control and optimal control by measuring the relevant process variables during operation (as in feedback control) and using a control algorithm that attempts to optimize some index of performance (as in optimal control).

Adaptive control is distinguished from feedback control and steady-state optimal control by its unique capability to cope with a time-varying environment. It is not unusual for a system to operate in an environment that changes over time and for the changes to have a potential effect on system performance. If the internal parameters or mechanisms of the system are fixed, as in feedback control or optimal control, the system may perform quite differently in one type of environment than in another. An adaptive control system is designed to compensate for its changing environment by monitoring its own performance and altering some aspect of its control mechanism to achieve optimal or near-optimal performance. In a production process, the "time-varying environment" consists of the variations in processing variables, raw materials, tooling, atmospheric conditions, and the like, any of which may affect performance.

The general configuration of an adaptive control system is illustrated in Figure 5.5. To evaluate its performance and respond accordingly, an adaptive control system performs three functions, as shown in the figure:

1. *Identification function.* In this function, the current value of the index of performance of the system is determined, based on measurements collected from the process. Since the environment changes over time, system performance also

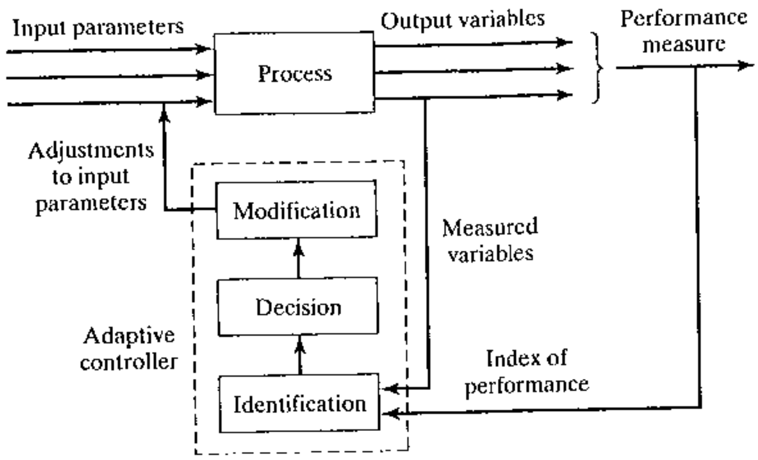


Figure 5.5 Configuration of an adaptive control system.

changes. Accordingly, the identification function must be accomplished more or less continuously over time during system operation.

2. *Decision function.* Once system performance has been determined, the next function is to decide what changes should be made to improve performance. The decision function is implemented by means of the adaptive system's programmed algorithm. Depending on this algorithm, the decision may be to change one or more input parameters to the process, to alter some of the internal parameters of the controller, or to make other changes.
3. *Modification function.* The third function of adaptive control is to implement the decision. Whereas decision is a logic function, modification is concerned with physical changes in the system. It involves hardware rather than software. In modification, the system parameters or process inputs are altered using available actuators to drive the system toward a more optimal state.

Adaptive control is most applicable at levels 2 and 3 in our automation hierarchy (Table 5.2). Adaptive control has been the subject of research and development for several decades; it was originally motivated by problems of high-speed flight control in the age of jet aircraft. The principles have been applied in other areas as well, including manufacturing. One notable example is *adaptive control machining*, in which changes in process variables such as cutting force, power, and vibration are used to effect control over process parameters such as cutting speed and feed rate.

On-Line Search Strategies. On-line search strategies can be used to address a special class of adaptive control problem in which the decision function cannot be sufficiently defined; that is, the relationship between the input parameters and the index of performance is not known, or not known well enough to use adaptive control as previously described. Therefore, it is not possible to decide on the changes in the internal parameters of the system to produce the desired performance improvement. Instead, experiments must be performed on the process. Small systematic changes are made in the input parameters of the process to observe the effect of these changes on the output variables. Based on the results of these experiments, larger changes are made in the input parameters to drive the process toward improved performance.

On-line search strategies include a variety of schemes to explore the effects of changes in process parameters, ranging from trial-and-error techniques to gradient methods. All of the schemes attempt to determine which input parameters cause the greatest positive effect on the index of performance and then move the process in that direction. There is little evidence that on-line search techniques are used much in discrete parts manufacturing. Their applications are more common in the continuous process industries.

Other Specialized Techniques. Other specialized techniques include strategies that are currently evolving in control theory and computer science. Examples include learning systems, expert systems, neural networks, and other artificial intelligence methods for process control.

5.2.2 Discrete Control Systems

In discrete control, the parameters and variables of the system are changed at discrete moments in time. The changes involve variables and parameters that are also discrete, typically binary (ON/OFF). The changes are defined in advance by means of a program of instructions, for example, a work cycle program (Section 4.1.2). The changes are executed either because the state of the system has changed or because a certain amount of time has elapsed. These two cases can be distinguished as (1) event-driven changes or (2) time-driven changes [2].

An *event-driven change* is executed by the controller in response to some event that has caused the state of the system to be altered. The change can be to initiate an operation or terminate an operation, start a motor or stop it, open a valve or close it, and so forth. Examples of event-driven changes are

- A robot loads a workpart into the fixture, and the part is sensed by a limit switch. Sensing the part's presence is the event that alters the system state. The event-driven change is that the automatic machining cycle can now commence.
- The diminishing level of plastic molding compound in the hopper of an injection molding machine triggers a low-level switch, which in turn opens a valve to start the flow of new plastic into the hopper. When the level of plastic reaches the high-level switch, this triggers the valve to close, thus stopping the flow of pellets into the hopper.
- Counting parts moving along a conveyor past an optical sensor is an event-driven system. Each part moving past the sensor is an event that drives the counter.

A *time-driven change* is executed by the control system either at a specific point in time or after a certain time lapse has occurred. As before, the change usually consists of starting something or stopping something, and the time when the change occurs is important. Examples of time-driven changes are

- In factories with specific starting times and ending times for the shift and uniform break periods for all workers, the "shop clock" is set to sound a bell at specific moments during the day to indicate these start and stop times.
- Heat treating operations must be carried out for a certain length of time. An automated heat treating cycle consists of automatic loading of parts into the furnace (perhaps by a robot) and then unloading after the parts have been heated for the specified length of time.

- In the operation of a washing machine, once the laundry tub has been filled to the preset level, the agitation cycle continues for a length of time set on the controls. When this time is up, the timer stops the agitation and initiates draining of the tub. (By comparison with the agitation cycle, filling the laundry tub with water is event-driven. Filling continues until the proper level is sensed, which causes the inlet valve to close.)

The two types of change correspond to two different types of discrete control, called combinational logic control and sequential control. *Combinational logic control* is used to control the execution of event-driven changes, and *sequential control* is used to manage time-driven changes. These types of control are discussed in our expanded coverage of discrete control in Chapter 9.

Discrete control is widely used in discrete manufacturing as well as the process industries. In discrete manufacturing, it is used to control the operation of conveyors and other material transport systems (Chapter 10), automated storage systems (Chapter 11), standalone production machines (Chapter 14), automated transfer lines (Chapter 16), automated assembly systems (Chapter 17), and flexible manufacturing systems (Chapter 19). All of these systems operate by following a well-defined sequence of start-and-stop actions, such as powered feed motions, parts transfers between workstations, and on-line automated inspections.

In the process industries, discrete control is associated more with batch processing than with continuous processes. In a typical batch processing operation, each batch of starting ingredients is subjected to a cycle of processing steps that involves changes in process parameters (e.g., temperature and pressure changes), possible flow from one container to another during the cycle, and finally packaging. The packaging step differs depending on the product. For foods, packaging may involve canning or boxing. For chemicals, it means filling containers with the liquid product. And for pharmaceuticals, it may involve filling bottles with medicine tablets. In batch process control, the objective is to manage the sequence and timing of processing steps as well as to regulate the process parameters in each step. Accordingly, batch process control typically includes both continuous control as well as discrete control.

5.3 COMPUTER PROCESS CONTROL

The use of digital computers to control industrial processes had its origins in the continuous process industries in the late 1950s (Historical Note 5.1). Prior to that point, analog controllers were used to implement continuous control, and relay systems were used to implement discrete control. At that time, computer technology was in its infancy, and the only computers available for process control were large, expensive mainframes. Compared with today's technology, the digital computers of the 1950s were slow, unreliable, and not well suited to process control applications. The computers that were installed sometimes cost more than the processes they controlled. Around 1960, digital computers started replacing analog controllers in continuous process control applications, and around 1970, programmable logic controllers started replacing relay banks in discrete control applications. Advances in computer technology since the 1960s and 1970s have resulted in the development of the microprocessor. Today, virtually all industrial processes, certainly new installations, are controlled by digital computers based on microprocessor technology. Microprocessor-based controllers are discussed in Section 5.3.3.

Historical Note 5.1 Computer process control [1, 7].

Control of industrial processes by digital computers can be traced to the process industries in the late 1950s and early 1960s. These industries, such as oil refineries and chemical plants, use high-volume continuous production processes characterized by many variables and associated control loops. The processes had traditionally been controlled by analog devices, each loop having its own set point value and in most instances operating independently of other loops. Any coordination of the process was accomplished in a central control room, where workers adjusted the individual settings, attempting to achieve stability and economy in the process. The cost of the analog devices for all of the control loops was considerable, and the human coordination of the process was less than optimal. The commercial development of the digital computer in the 1950s offered the opportunity to replace some of the analog control devices with the computer.

The first known attempt to use a digital computer for process control was at a Texaco refinery in Port Arthur, Texas in the late 1950s. Texaco had been contacted in 1956 by computer manufacturer Thomson Ramo Woodridge (TRW), and a feasibility study was conducted on a polymerization unit at the refinery. The computer control system went on line in March 1959. The control application involved 26 flows, 72 temperatures, three pressures, and three compositions. This pioneering work did not escape the notice of other companies in the process industries as well as other computer companies. The process industries saw computer process control as a means of automation, and the computer companies saw a potential market for their products.

The available computers in the late 1950s were not reliable, and most of the subsequent process control installations operated by either printing out instructions for the operator or by making adjustments in the set points of analog controllers, thereby reducing the risk of process downtime due to computer problems. The latter mode of operation was called *set point control*. By March 1961, a total of 37 computer process control systems had been installed. Much experience was gained from these early installations. The *interrupt feature* (Section 5.3.2), by which the computer suspends current program execution to quickly respond to a process need, was developed during this period.

The first *direct digital control* (DDC) system (Section 5.3.3), in which certain analog devices are replaced by the computer, was installed by Imperial Chemical Industries in England in 1962. In this implementation, 224 process variables were measured, and 129 actuators (valves) were controlled. Improvements in DDC technology were made, and additional systems were installed during the 1960s. Advantages of DDC noted during this time included (1) cost savings by eliminating analog instrumentation, (2) simplified operator display panels, and (3) flexibility due to reprogramming capability.

Computer technology was advancing, leading to the development of the *minicomputer* in the late 1960s. Process control applications were easier to justify using these smaller, less expensive computers. Development of the *microcomputer* in the early 1970s continued this trend. Lower cost process control hardware and interface equipment (such as analog-to-digital converters) were becoming available due to the larger markets made possible by low-cost computer controllers.

Most of the developments in computer process control up to this time were biased toward the process industries rather than discrete part and product manufacturing. Just as analog devices had been used to automate process industry operations, relay banks were widely used to satisfy the discrete process control (ON/OFF) requirements in manufacturing automation. The *programmable logic controller* (PLC), a control computer designed for discrete process control, was developed in the early 1970s (Historical Note 9.1). Also, *numerical control* (NC) machine tools (Historical Note 7.1) and industrial *robots* (Historical Note 8.1), technologies that preceded computer control, started to be designed with digital computers as their controllers.

The availability of low-cost microcomputers and programmable logic controllers resulted in a growing number of installations in which a process was controlled by multiple computers networked together. The term *distributed control* was used for this kind of system, the first of which was a product offered by Honeywell in 1975. In the early 1990s, *personal computers* (PCs) began to be utilized on the factory floor, sometimes to provide scheduling and engineering data to shop floor personnel, in other cases as the operator interface to processes controlled by PLCs. Today, a growing number of PCs are being used to directly control manufacturing operations.

In this section on computer process control, we identify the requirements placed on the computer in industrial control applications. We then examine the capabilities that have been incorporated into the control computer to address these requirements, and finally we survey the various forms of computer control used in industry.

5.3.1 Control Requirements

Whether the application involves continuous control, discrete control, or both, there are certain basic requirements that tend to be common to nearly all process control applications. By and large, they are concerned with the need to communicate and interact with the process on a real-time basis. A *real-time controller* is a controller that is able to respond to the process within a short enough time period that process performance is not degraded. Real-time control usually requires the controller to be capable of *multitasking*, which means coping with multiple tasks concurrently without the tasks interfering with one another.

There are two basic requirements that must be managed by the controller to achieve real-time control:

1. *Process-initiated interrupts*. The controller must be able to respond to incoming signals from the process. Depending on the relative importance of the signals, the computer may need to interrupt execution of a current program to service a higher priority need of the process. A process-initiated interrupt is often triggered by abnormal operating conditions, indicating that some corrective action must be taken promptly.
2. *Timer-initiated actions*. The controller must be capable of executing certain actions at specified points in time. Timer-initiated actions can be generated at regular time intervals, ranging from very low values (e.g., 100 μs) to several minutes, or they can be generated at distinct points in time. Typical timer-initiated actions in process control include (1) scanning sensor values from the process at regular sampling intervals, (2) turning on and off switches, motors, and other binary devices associated with the process at discrete points in time during the work cycle, (3) displaying performance data on the operator's console at regular times during a production run, and (4) recomputing optimal process parameter values at specified times.

These two requirements correspond to the two types of changes mentioned previously in the context of discrete control systems: (1) event-driven changes and (2) time-driven changes.

In addition to these basic requirements, the control computer must also deal with other types of interruptions and events. These include the following:

3. *Computer commands to process.* In addition to receiving incoming signals from the process, the control computer must send control signals to the process to accomplish a corrective action. These output signals may actuate a certain hardware device or readjust a set point in a control loop.
4. *System- and program-initiated events.* These are events related to the computer system itself. They are similar to the kinds of computer operations associated with business and engineering applications of computers. A *system-initiated event* involves communications among computers and peripheral devices linked together in a network. In these multiple computer networks, feedback signals, control commands, and other data must be transferred back and forth among the computers in the overall control of the process. A *program-initiated event* occurs when the program calls for some non-process-related action, such as the printing or display of reports on a printer or monitor. In process control, system- and program-initiated events generally occupy a low level of priority compared with process interrupts, commands to the process, and timer-initiated events.
5. *Operator-initiated events.* Finally, the control computer must be able to accept input from operating personnel. Operator-initiated events include (1) entering new programs; (2) editing existing programs; (3) entering customer data, order number, or startup instructions for the next production run; (4) requesting process data; and (5) calling for emergency stops.

5.3.2 Capabilities of Computer Control

The above requirements can be satisfied by providing the controller with certain capabilities that allow it to interact on a real-time basis with the process and the operator. The capabilities are (1) polling, (2) interlocks, (3) interrupt system, and (4) exception handling.

Polling (Data Sampling). In computer process control, polling refers to the periodic sampling of data that indicates the status of the process. When the data consist of a continuous analog signal, sampling means that the continuous signal is substituted with a series of numerical values that represent the continuous signal at discrete moments in time. The same kind of substitution holds for discrete data, except that the number of possible numerical values the data can take on is more limited—certainly the case with binary data. We discuss the techniques by which continuous and discrete data are entered into and transmitted from the computer in Chapter 6. Other names for polling include *sampling* and *scanning*.

In some systems, the polling procedure simply requests whether any changes have occurred in the data since the last polling cycle and then collects only the new data from the process. This tends to shorten the cycle time required for polling. Issues related to polling include

1. *Polling frequency.* This is the reciprocal of the time interval between data collections.
2. *Polling order.* The polling order is the sequence in which the different data collection points of the process are sampled.
3. *Polling format.* This refers to the manner in which the sampling procedure is designed. The alternatives include (a) entering all new data from all sensors and other devices every polling cycle; (b) updating the control system only with data that have changed since the last polling cycle; or (c) using *high-level and low-level scanning*, or

conditional scanning, in which only certain key data are collected each polling cycle (high-level scanning), but if the data indicates some irregularity in the process, a low-level scan is undertaken to collect more complete data to ascertain the source of the irregularity.

These issues become increasingly critical with very dynamic processes in which changes in process status occur rapidly.

Interlocks. An interlock is a safeguard mechanism for coordinating the activities of two or more devices and preventing one device from interfering with the other(s). In process control, interlocks provide a means by which the controller is able to sequence the activities in a work cell, ensuring that the actions of one piece of equipment are completed before the next piece of equipment begins its activity. Interlocks work by regulating the flow of control signals back and forth between the controller and the external devices.

There are two types of interlocks, input interlocks and output interlocks, where input and output are defined relative to the controller. An *input interlock* is a signal that originates from an external device (e.g., a limit switch, sensor, or production machine) and that is sent to the controller. Input interlocks can be used for either of the following functions:

1. To proceed with the execution of the work cycle program. For example, the production machine communicates a signal to the controller that it has completed its processing of the part. This signal constitutes an input interlock indicating that the controller can now proceed to the next step in the work cycle, which is to unload the part.
2. To interrupt the execution of the work cycle program. For example, while unloading the part from the machine, the robot accidentally drops the part. The sensor in its gripper transmits an interlock signal to the controller indicating that the regular work cycle sequence should be interrupted until corrective action is taken.

An *output interlock* is a signal sent from the controller to some external device. It is used to control the activities of each external device and to coordinate their operation with that of the other equipment in the cell. For example, an output interlock can be used to send a control signal to a production machine to begin its automatic cycle after the workpart has been loaded into it.

Interrupt System. Closely related to interlocks is the interrupt system. As suggested by our discussion of input interlocks, there are occasions when it becomes necessary for the process or operator to interrupt the regular controller operation to deal with more pressing matters. All computer systems are capable of being interrupted, if nothing else, by turning off the power. A more sophisticated interrupt system is required for process control applications. An *interrupt system* is a computer control feature that permits the execution of the current program to be suspended to execute another program or subroutine in response to an incoming signal indicating a higher priority event. Upon receipt of an interrupt signal, the computer system transfers to a predetermined subroutine designed to deal with the specific interrupt. The status of the current program is remembered so that its execution can be resumed when servicing of the interrupt has been completed.

TABLE 5.4 Possible Priority Levels in an Interrupt System

Priority Level	*Computer Function*
1 (lowest priority)	Most operator inputs
2	System and program interrupts
3	Timer interrupts
4	Commands to process
5	Process interrupts
6 (highest priority)	Emergency stop (operator input)

Interrupt conditions can be classified as internal or external. *Internal interrupts* are generated by the computer system itself. These include timer-initiated events, such as polling of data from sensors connected to the process, or sending commands to the process at specific points in clock time. System- and program-initiated interrupts are also classified as internal because they are generated within the system. *External interrupts* are external to the computer system; they include process-initiated interrupts and operator inputs.

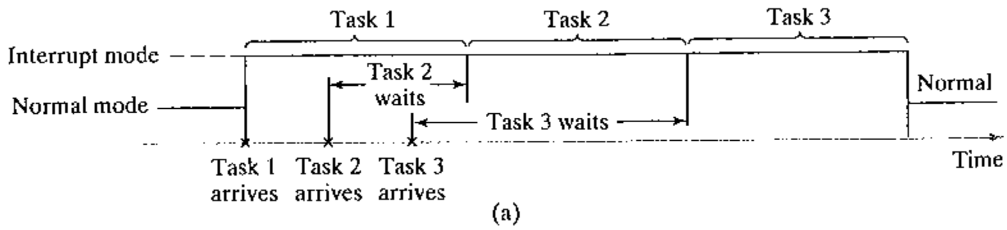
An interrupt system is required in process control because it is essential that more important programs (ones with higher priority) be executed before less important programs (ones with lower priorities). The system designer must decide what level of priority should be attached to each control function. A higher priority function can interrupt a lower priority function. A function at a given priority level cannot interrupt a function at the same priority level. The number of priority levels and the relative importance of the functions depend on the requirements of the individual process control situation. For example, emergency shutdown of a process because of safety hazards would occupy a very high priority level, even if it is an operator-initiated interrupt. Most operator inputs would have low priorities.

One possible organization of priority rankings for process control functions is shown in Table 5.4. Of course, the priority system may have more or fewer than the number of levels shown here, depending on the control situation. For example, some process interrupts may be more important than others, and some system interrupts may take precedence over certain process interrupts, thus requiring more than the six levels indicated in our table.

To respond to the various levels of priority defined for a given control application, an interrupt system can have one or more interrupt levels. A *single-level interrupt system* has only two modes of operation: normal mode and interrupt mode. The normal mode can be interrupted, but the interrupt mode cannot. This means that overlapping interrupts are serviced on a first-come, first-served basis, which could have potentially hazardous consequences if an important process interrupt was forced to wait its turn while a series of less important operator and system interrupts were serviced. A *multilevel interrupt system* has a normal operating mode plus more than one interrupt level. The normal mode can be interrupted by any interrupt level, but the interrupt levels have relative priorities that determine which functions can interrupt others. Example 5.1 illustrates the difference between the single-level and multilevel interrupt systems.

EXAMPLE 5.1 Single-Level Versus Multilevel Interrupt Systems

Three interrupts representing tasks of three different priority levels arrive for service in the reverse order of their respective priorities. Task 1 with the lowest priority, arrives first. Soon after, higher priority Task 2 arrives. And soon after



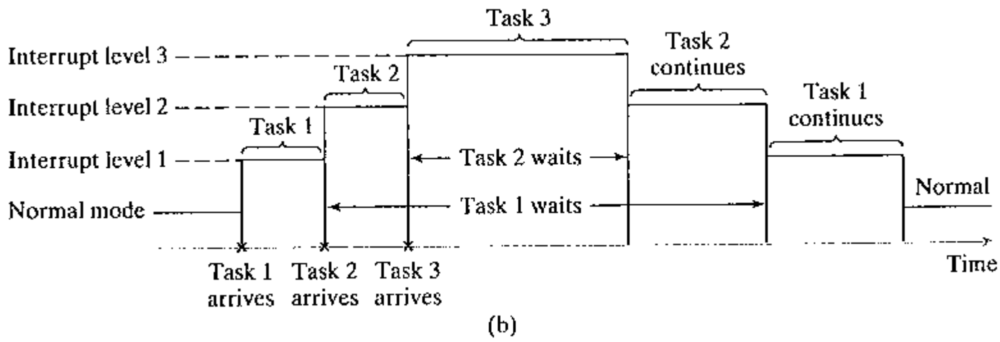


Figure 5.6 Response of the computer control system in Example 5.1 to three priority interrupts for (a) a single-level interrupt system and (b) a multilevel interrupt system. Task 3 is the highest level priority. Task 1 is the lowest level. Tasks arrive for servicing in the order 1, then 2, then 3. In (a), Task 3 must wait until Tasks 1 and 2 have been completed. In (b), Task 3 interrupts execution of Task 2, whose priority level allowed it to interrupt Task 1.

that, highest priority Task 3 arrives. How would the computer control system respond under (a) a single-level interrupt system and (b) a multilevel interrupt system?

Solution: The response of the system for the two interrupt systems is shown in Figure 5.6.

Exception Handling. In process control, an *exception* is an event that is outside the normal or desired operation of the process or control system. Dealing with the exception is an essential function in industrial process control and generally occupies a major portion of the control algorithm. The need for exception handling may be indicated through the normal polling procedure or by the interrupt system. Examples of events that may invoke exception handling routines include

- product quality problem,
- process variables operating outside their normal ranges,
- shortage of raw materials or supplies necessary to sustain the process,
- hazardous conditions such as a fire,
- controller malfunction.

In effect, exception handling is a form of error detection and recovery, discussed in the context of advanced automation capabilities (Section 4.2.3).

5.3.3 Forms of Computer Process Control

There are various ways in which computers can be used to control a process. First, we can distinguish between process monitoring and process control as illustrated in Figure 5.7. In process monitoring, the computer is used to simply collect data from the process, while in process control, the computer regulates the process. In some process control implementations, certain actions are implemented by the control computer that do not require feedback data to be collected from the process. This is open loop control. However, in most cases, some form of feedback or interlocking is required to ensure that the control instructions have been properly carried out. This more common situation is closed loop control.

In this section, we survey the various forms of computer process monitoring and control, all but one of which are commonly used in industry today. Direct digital control (DDC) represents a transitory phase in the evolution of computer process control technology. In its pure form, it is no longer used today. However, we briefly describe DDC to reveal the opportunities it contributed. Distributed control systems, often implemented using personal computers, are the most recent means of implementing computer process control.

Computer Process Monitoring. Computer process monitoring is one of the ways in which the computer can be interfaced with a process. It involves the use of the computer to observe the process and associated equipment and to collect and record data from the operation. The computer is not used to directly control the process. Control remains in the hands of humans who use the data to guide them in managing and operating the process. The data collected by the computer in computer process monitoring can generally be classified into three categories:

1. *Process data.* These are measured values of input parameters and output variables that indicate process performance. When the values are found to indicate a problem, the human operator takes corrective action.

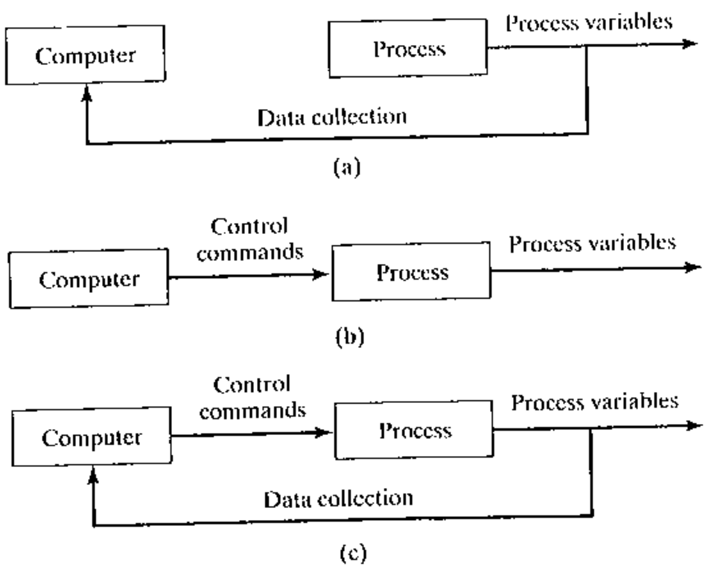


Figure 5.7 (a) process monitoring, (b) open loop process control, and (c) closed loop process control.

2. *Equipment data.* These data indicate the status of the equipment in the work cell. The data are used to monitor machine utilization, schedule tool changes, avoid machine breakdowns, diagnose equipment malfunctions, and plan preventive maintenance.
3. *Product data.* Government regulations require certain manufacturing industries to collect and preserve production data on their products. The pharmaceutical and medical supply industries are prime examples. Computer monitoring is the most convenient means of satisfying these regulations. A firm may also want to collect product data for its own use.

Collecting data from factory operations can be accomplished by any of several means. Shop data can be entered by workers through manual terminals located throughout the plant or can be collected automatically by means of limit switches, sensor systems, bar code readers, or other devices. Sensors are described in Chapter 6. Automatic identification and data collection technologies are discussed in Chapter 12. The collection and use of production data in factory operations for scheduling and tracking purposes is called *shop floor control*, covered in Chapter 25.

Direct Digital Control. DDC was certainly one of the important steps in the development of computer process control. Let us briefly examine this computer control mode and its limitations, which motivated improvements leading to modern computer control technology. DDC is a computer process control system in which certain components in a conventional analog control system are replaced by the digital computer. The regulation of the process is accomplished by the digital computer on a time-shared, sampled-data basis rather than by the many individual analog components working in a dedicated continuous manner. With DDC, the computer calculates the desired values of the input parameters and set points, and these values are applied through a direct link to the process, hence the name "direct digital" control.

The difference between direct digital control and analog control can be seen by comparing Figures 5.8 and 5.9. The first figure shows the instrumentation for a typical analog control loop. The entire process would have many individual control loops, but only one is shown here. Typical hardware components of the analog control loop include the sensor and transducer, an instrument for displaying the output variable (such an instrument is not always included in the loop), some means for establishing the set point of the loop (shown

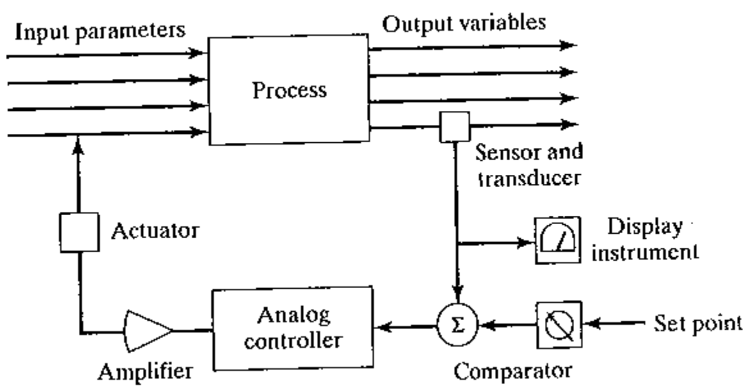


Figure 5.8 A typical analog control loop.

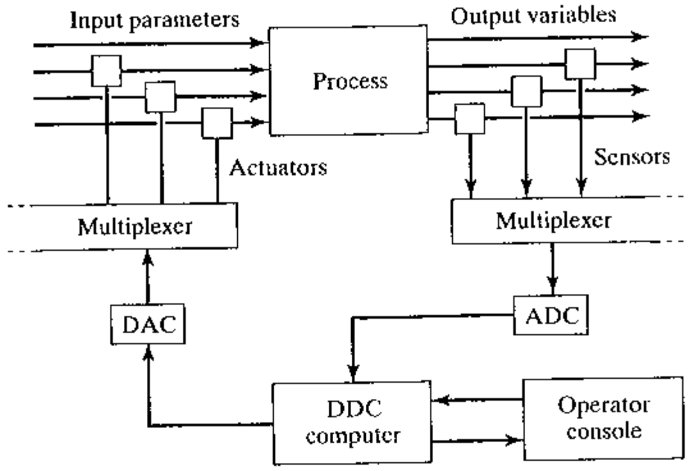


Figure 5.9 Components of a DDC system.

as a dial in the figure, suggesting that the setting is determined by a human operator), a comparator (to compare set point with measured output variable), the analog controller, an amplifier, and the actuator that determines the input parameter to the process.

In the DDC system (Figure 5.9), some of the control loop components remain unchanged, including (probably) the sensor and transducer as well as the amplifier and actuator. Components likely to be replaced in DDC include the analog controller, recording and display instruments, set point dials, and comparator. New components in the loop include the digital computer, analog-to-digital and digital-to-analog converters (ADCs and DACs), and multiplexers to share data from different control loops with the same computer.

DDC was originally conceived as a more efficient means of performing the same kinds of control actions as the analog components it replaced. However, the practice of simply using the digital computer to imitate the operation of analog controllers seems to have been a transitional phase in computer process control. Additional opportunities for the control computer were soon recognized, including:

- *More control options than traditional analog.* With digital computer control, it is possible to perform more complex control algorithms than with the conventional proportional-integral-derivative control modes used by analog controllers; for example, on/off control or nonlinearities in the control functions can be implemented.
- *Integration and optimization of multiple loops.* This is the ability to integrate feedback measurements from multiple loops and to implement optimizing strategies to improve overall process performance.
- *Ability to edit the control programs.* Using a digital computer makes it relatively easy to change the control algorithm when necessary by simply reprogramming the computer. Reprogramming the analog control loop is likely to require hardware changes that are more costly and less convenient.

These enhancements have rendered the original concept of direct digital control more or less obsolete. In addition, computer technology itself has progressed dramatically so that much smaller and less expensive yet more powerful computers are available for process control than the large mainframes available in the early 1960s. This has allowed computer process control to be economically justified for much smaller scale processes and equipment. It has also motivated the use of *distributed control systems,* in which a network of microcomputers is utilized to control a complex process consisting of multiple unit operations and/or machines.

Numerical Control and Robotics. Numerical control (NC) is another form of industrial computer control. It involves the use of the computer (again, a microcomputer) to direct a machine tool through a sequence of processing steps defined by a program of instructions specifying the details of each step and their sequence. The distinctive feature of NC is control of the relative position of a tool with respect to the object (workpart) being processed. Computations must be made to determine the trajectory that will be followed by the cutting tool to shape the part geometry. Hence, NC requires the controller to execute not only sequence control but geometric calculations as well. Because of its importance in manufacturing automation and industrial control, NC is covered in detail in Chapter 7.

Closely related to NC is industrial robotics, in which the joints of the manipulator (robot arm) are controlled to move the end of the arm through a sequence of positions during the work cycle. As in NC, the controller must perform calculations during the work cycle to implement motion interpolation, feedback control, and other functions. In addition, a robotic work cell usually includes other equipment besides the robot, and the activities of the other equipment in the work cell must be coordinated with those of the robot. This coordination is achieved using interlocks. We discuss industrial robotics in Chapter 8.

Programmable Logic Controllers. Programmable logic controllers (PLCs) were introduced around 1970 as an improvement on the electromechanical relay controllers used at the time to implement discrete control in the discrete manufacturing industries. The evolution of PLCs has been facilitated by advances in computer technology, and present-day PLCs are capable of much more than the 1970s controllers. We can define a modern *programmable logic controller* as a microprocessor-based controller that uses stored instructions in programmable memory to implement logic, sequencing, timing, counting, and arithmetic control functions for controlling machines and processes. Today's PLCs are used for both continuous control and discrete control applications in both the process industries and discrete manufacturing. We cover PLCs and the kinds of control they are used to implement in Chapter 9.

Supervisory Control. The term *supervisory control* is usually associated with the process industries, but the concept applies equally well to discrete manufacturing automation, where it corresponds to cell or system level control. Supervisory control represents a higher level of control than DDC, NC, and PLCs. In general, these other types of control systems are interfaced directly to the process. By contrast, supervisory control is often superimposed on these process-level control systems and directs their operations. The relationship between supervisory control and the process-level control techniques is illustrated in Figure 5.10.

In the context of the process industries, *supervisory control* denotes a control system that manages the activities of a number of integrated unit operations to achieve certain economic objectives for the process. In some applications, supervisory control is not much more than regulatory control or feedforward control. In other applications, the supervisory control system is designed to implement optimal or adaptive control. It seeks to optimize some well-defined objective function, which is usually based on economic criteria such as yield, production rate, cost, quality, or other objectives that pertain to process performance.

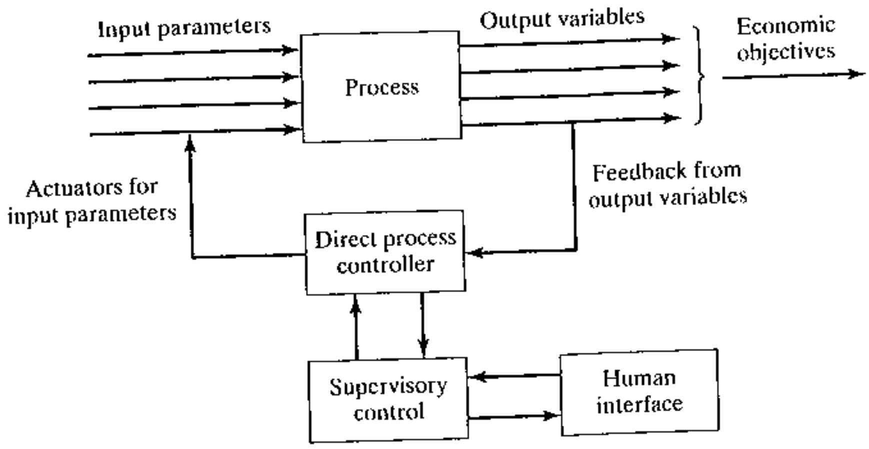


Figure 5.10 Supervisory control superimposed on other process-level control systems.

In the context of discrete manufacturing, *supervisory control* can be defined as the control system that directs and coordinates the activities of several interacting pieces of equipment in a manufacturing cell or system, such as a group of machines interconnected by a material handling system. Again, the objectives of supervisory control are motivated by economic considerations. The control objectives might include minimizing part or product costs by determining optimum operating conditions, maximizing machine utilization through efficient scheduling, or minimizing tooling costs by tracking tool lives and scheduling tool changes.

It is tempting to conceptualize a supervisory control system as being completely automated, so that the system operates with no human interference or assistance. But in virtually all cases, supervisory control systems are designed to allow interaction with human operators, and the responsibility for control is shared between the controller and the human. The relative proportions of responsibility differ, depending on the application.

Distributed Control Systems. With the development of the microprocessor, it became feasible to connect multiple microcomputers together to share and distribute the process control workload. The term *distributed control system* (DCS) is used to describe such a configuration, which consists of the following components and features [8]:

- Multiple process control stations located throughout the plant to control the individual loops and devices of the process.
- A central control room equipped with operator stations, where supervisory control of the plant occurs.
- Local operator stations distributed throughout the plant. This provides the DCS with redundancy. If a control failure occurs in the central control room, the local operator stations take over the central control functions. If a local operator station fails, the other local operator stations assume the functions of the failed station.
- All process and operator stations interact with each other by means of a communications network, or data highway, as it is often called.

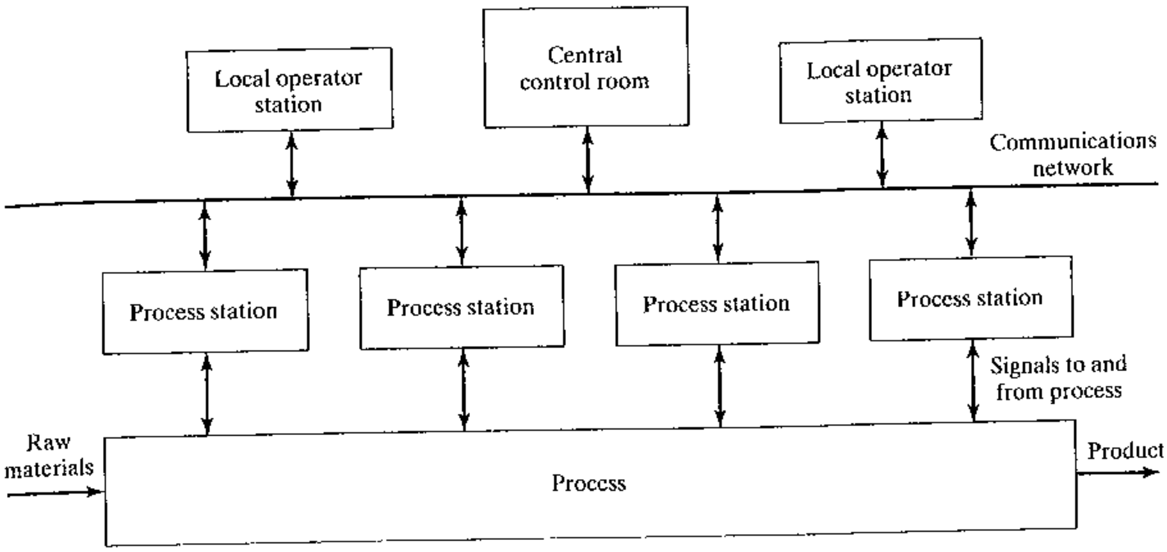


Figure 5.11 Distributed control system.

These components are illustrated in a typical configuration of a distributed process control system presented in Figure 5.11. There are a number of benefits and advantages of the DCS: (1) A DCS can be installed for a given application in a very basic configuration, then enhanced and expanded as needed in the future; (2) since the system consists of multiple computers, this facilitates parallel multitasking; (3) because of its multiple computers, a DCS has built-in redundancy; (4) control cabling is reduced compared with a central computer control configuration; and (5) networking provides process information throughout the enterprise for more efficient plant and process management.

Development of DCSs started around 1970. One of the first commercial systems was Honeywell's TDC 2000, introduced in 1975 [1]. The first DCS applications were in the process industries. In the discrete manufacturing industries, programmable logic controllers were introduced about the same time. The concept of distributed control applies equally well to PLCs; that is, multiple PLCs located throughout a factory to control individual pieces of equipment but integrated by means of a common communications network. Introduction of the PC shortly after the DCS and PLC, and its subsequent increase in computing power and reduction in cost over the years, have stimulated a significant growth in the adoption of PC-based DCSs for process control applications.

PCs in Process Control. Today, PCs dominate the computer world. They have become the standard tool by which business is conducted, whether in manufacturing or in the service sector. Thus, it is no surprise that PCs are being used in growing numbers in process control applications. Two basic categories of PC implementations in process control can be distinguished: (1) operator interface and (2) direct control. Whether used as the operator interface or for direct control, PCs are likely to be networked with other computers to create distributed control systems.

When used as the operator interface, the PC is interfaced to one or more PLCs or other devices (possibly other microcomputers) that directly control the process. Personal computers have been used to perform the operator interface function since the early 1980s. In this function, the computer performs certain monitoring and supervisory control functions, but it does not directly control the process. Some advantages of using a PC only as the operator interface are that (1) the PC provides a user-friendly interface for the operator; (2) the PC can be used for all of the conventional computing and data processing functions that PCs traditionally perform; (3) the PLC or other device that is directly controlling the process is isolated from the PC, so a PC failure will not disrupt control of the process; and (4) the computer can be easily upgraded as PC technology advances and capabilities improve, while the PLC control software and connections with the process can remain in place.

The second way of implementing PCs in process control is *direct control*, which means that the PC is interfaced directly to the process and controls its operations in real time. The traditional thinking has been that it is too risky to permit the PC to directly control the production operation. If the computer were to fail, the uncontrolled operation might stop working, produce a defective product, or become unsafe. Another factor is that conventional PCs, equipped with the usual business-oriented operating system and applications software, are designed for computing and data processing functions, not for process control. They are not intended to be interfaced with an external process in the manner necessary for real-time process control. Finally, most PCs are designed to be used in an office environment, not in the harsh factory atmosphere.

Recent advances in both PC technology and available software have challenged this traditional thinking. Starting in the early 1990s, PCs have been installed at an accelerating pace for direct control of industrial processes. Several factors have enabled this trend:

- Widespread familiarity with PCs. User-friendly software for the home and business has certainly contributed to the popularity of PCs. There is a growing expectation among workers that they will be provided with a computer in their workplace, even if that workplace is in the factory.
- Availability of high performance PCs, capable of satisfying the demanding requirements of process control (Section 5.3.1).
- Trend toward *open architecture philosophy* in control systems design, in which vendors of control hardware and software agree to comply with published standards that allow their products to be interoperable. This means that components from different vendors can be interconnected in the same system. The traditional philosophy had been for each vendor to design proprietary systems, requiring the user to purchase the complete hardware and software package from one supplier. Open architecture allows the user a wider choice of products in the design of a given process control application.
- Availability of PC operating systems that facilitate real-time control, multitasking, and networking. At the same time, these systems provide the user friendliness of the desktop PC and most of the power of an engineering workstation. Installed in the factory, a PC equipped with the appropriate software can perform multiple functions simultaneously, such as data logging, trend analysis, and displaying an animated view of the process as it proceeds, all while reserving a portion of its CPU capacity for direct control of the process.

Regarding the factory environment issue, this can be addressed by using industrial-grade PCs, which are equipped with enclosures designed for the rugged plant environment. Compared with the previously discussed PC/PLC configuration, in which the PC is used only as the operator interface, there is a cost savings from installing one PC for direct control rather than a PC plus a PLC. A related issue is data integration: Setting up a data link between a PC and a PLC is more complex than when the data are all in one PC.

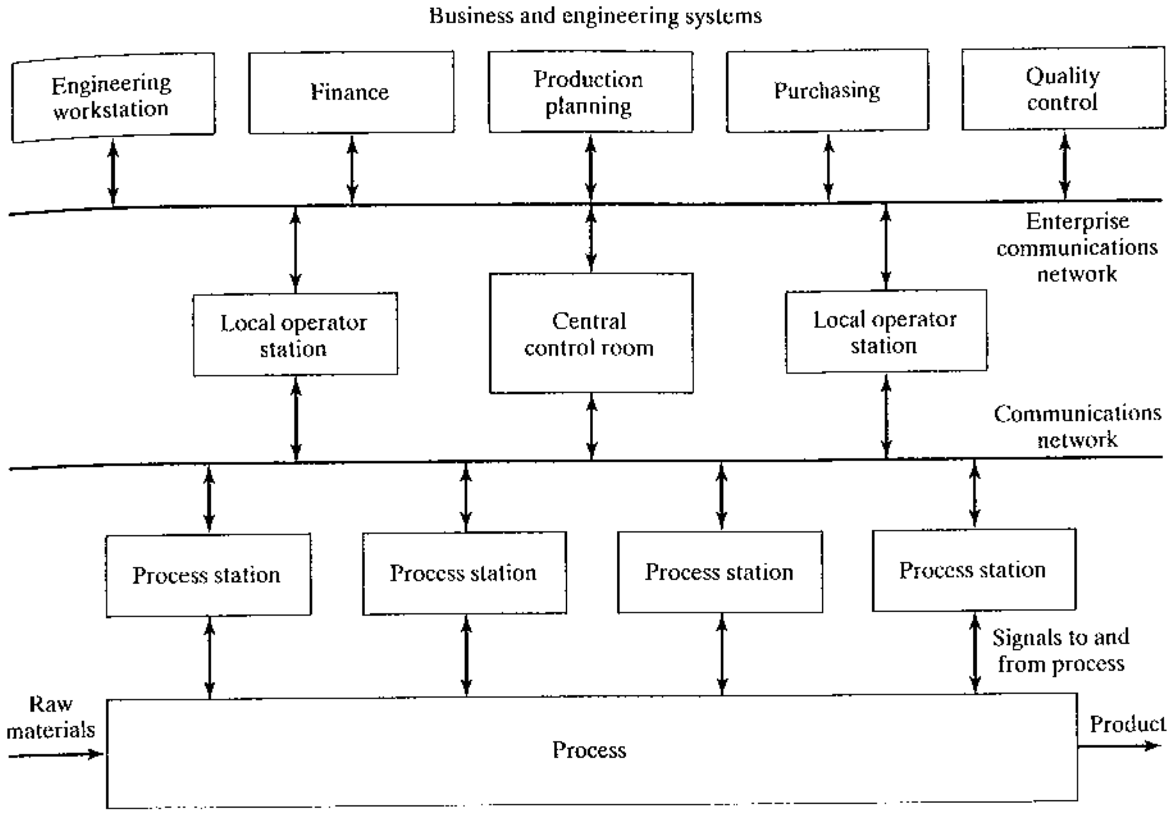


Figure 5.12 Enterprise-wide PC-based DCS.

Enterprise-Wide Integration of Factory Data. The most recent progression in PC-based distributed control is enterprise-wide integration of factory operations data, as depicted in Figure 5.12. This trend is consistent with modern information management and worker empowerment philosophies. These philosophies assume fewer levels of company management and greater responsibilities for front-line workers in sales, order scheduling, and production. The networking technologies that allow such integration are available. The latest PC operating systems provide a number of built-in and optional features for connecting the industrial control system in the factory to enterprise-wide business systems and supporting data exchange between various applications (e.g., allowing data collected in the plant to be used in analysis packages, such as spreadsheets). The term *enterprise resource planning* (ERP) refers to a computer software system that achieves company-wide integration of not only factory data but of all the other data required to execute the business functions of the organization. A key feature of ERP is the use of a single central database that can be accessed from anywhere in the company. Some of the details of ERP are discussed in Chapter 25 (Section 25.6.2).

Following are some of the capabilities that are enabled by making process data available throughout the enterprise:

1. Managers can have more direct access to factory floor operations.
2. Production planners can use the most current data on times and production rates in scheduling future orders.

3. Sales personnel can provide realistic estimates on delivery dates to customers, based on current shop loading.
4. Order trackers are able to provide inquiring customers with current status information on their orders.
5. Quality control personnel are made aware of real or potential quality problems on current orders, based on access to quality performance histories from previous orders.
6. Cost accounting has access to the most recent production cost data.
7. Production personnel can access part and product design details to clarify ambiguities and do their job more effectively.

REFERENCES

[1] ASTROM, K. J., and WITTENMARK, B., *Computer-Controlled Systems—Theory and Design*, 3d ed., Prentice Hall, Upper Saddle River, NJ, 1997.

[2] BATESON, R. N., *Introduction to Control System Technology*, 7th ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[3] BOUCHER, T. O., *Computer Automation in Manufacturing*, Chapman & Hall, London, UK, 1996.

[4] CAWLFIELD, D., "PC-Based Direct Control Flattens Control Hierarchy, Opens Information Flow," *Instrumentation & Control Systems*, September 1997, pp 61–67.

[5] GROOVER, M. P., "Industrial Control Systems," *Maynard's Industrial Engineering Handbook*, 5th ed., K. Zandin (ed.), McGraw-Hill Book Company, NY, 2001.

[6] HIRSH, D., "Acquiring and Sharing Data Seamlessly," *Instrumentation and Control Systems*, October 1997, pp. 25–35.

[7] OLSSON, G., and G. PIANI, *Computer Systems for Automation and Control*, Prentice Hall, London, UK, 1992.

[8] PLATT, G., *Process Control: A Primer for the Nonspecialist and the Newcomer*, 2d ed., Instrument Society of America, Research Triangle Park, NC, 1998.

[9] RULLAN, A., "Programmable Logic Controllers versus Personal Computers for Process Control," *Computers and Industrial Engineering*, Vol. 33, Nos. 1–2, 1997, pp. 421–424.

[10] STENERSON, J., *Fundamentals of Programmable Logic Controllers, Sensors, and Communications*, 3d ed., Pearson/Prentice Hall, Upper Saddle River, NJ, 2004.

REVIEW QUESTIONS

5.1 What is industrial control?

5.2 What is the difference between a continuous variable and a discrete variable?

5.3 Name and briefly define each of the three different types of discrete variables.

5.4 What is the difference between a continuous control system and a discrete control system?

5.5 What is feedforward control?

5.6 What is adaptive control?

5.7 What are the three functions of adaptive control?

5.8 What is the difference between an event-driven change and a time-driven change in discrete control?

5.9 What are the two basic requirements that must be managed by the controller to achieve real-time control?

5.10 What is polling in computer process control?

5.11 What is an interlock? What are the two types of interlocks in industrial control?

5.12 What is an interrupt system in computer process control?

5.13 What is computer process monitoring?

5.14 What is direct digital control (DDC), and why is it no longer used in industrial process control applications?

5.15 Are programmable logic controllers (PLCs) more closely associated with the process industries or the discrete manufacturing industries?

5.16 What is a distributed control system?

5.17 What is the open architecture philosophy in control systems design?