Karnatak Law Society's
Vishwanathrao Deshpande Institute of Technology, Haliyal
(Formerly Known as KLS Vishwanathrao Deshpande Rural Institute of Technology, Haliyal)
(Approved by AICTE, New Delhi. Affiliated to VTU, Belagavi)
Udyog Vidya Nagar, Haliyal – 581329, Dist: Uttar Kannada
Phone: 08284-220861, 220334, 221409, Fax: 08284-220813
Web: www.vdrit.org email: klsvdrit@yahoo.com

Module 5

MANUFACTURING CONTROL AND AUTOMATION

CNC technology - An overview: Introduction to NC/CNC/DNC machine tools, Classification of NC /CNC machine tools, Advantage, disadvantages of NC /CNC machine tools, Application of NC/CNC

Part programming: CNC programming and introduction, Manual part programming: Basic (Drilling, milling, turning etc.), Special part programming, Advanced part programming, Computer aided part programming (APT)

Introduction: Automation in production system principles and strategies of automation, basic Elements of an automated system. Advanced Automation functions. Levels of Automations, introduction to automation productivity

Control Technologies in Automation: Industrial control system. Process industry vs discrete manufacturing industries. Continuous vs discrete control. Continuous process and its forms. Other control system components.

CNC technology - An overview

Introduction to NC/CNC/DNC machine tools

AN OVERVIEW OF CNC MACHINES

Historical Perspective

The word NC which stands for numerical control refer to control of a machine or a process using symbolic codes consisting of characters and numerals. The word CNC came into existence in seventies when microprocessors and microcomputers replaced integrated circuit IC based controls used for NC machines. The development of numerical control owes much to the United States air force. The concept of NC was proposed in the late 1940s by John Parsons who recommended a method of automatic machine control that would guide a milling cutter to produce a curvilinear motion in order to generate smooth profiles on the work-pieces. In 1949, the U.S Air Force awarded Parsons a contract to develop new type of machine tool that would be able to speed up production methods.

Parsons sub-contracted the Massachusetts Institute of Technology (MIT) to develop a practical implementation of his concept. Scientists and engineers at M.I.T built a control system for a two axis milling machine that used a perforated paper tape as the input media. This prototype was produced by retrofitting a conventional tracer mill with numerical control servomechanisms for the three axes of the machine. By 1955, these machines were available to industries with some small modifications.

The machine tool builders gradually began developing their own projects to introduce commercial NC units. Also, certain industry users, especially airframe builders, worked to devise numerical control machines to satisfy their own particular production needs. The Air force continued its encouragement of NC development by sponsoring additional research at MIT to design a part programming language that could be used in controlling N.C. machines.

In a short period of time, all the major machine tool manufacturers were producing some machines with NC, but it was not until late 1970s that computer-based NC became widely used. NC matured as an automation technology when electronics industry developed new products. At first, miniature electronic tubes were developed, but the controls were big, bulky, and not very reliable. Then solid-state circuitry and eventually modular or integrated circuits were developed. The control unit became smaller, more reliable, and less expensive.

ComputerNumericalControl

Computer numerical control (CNC) is the numerical control system in which a dedicated computer is built into the control to perform basic and advanced NC functions. CNC controls are also referred to as soft-wired NC systems because most of their control functions are implemented by the control software programs. CNC is a computer assisted process to control general purpose machines from instructions generated by a processor and stored in a memory system. It is a specific form of control system where position is the principal controlled variable. All numerical control machines manufactured since the seventies are of

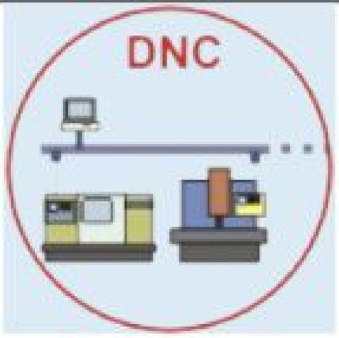
CNC type. The computer allows for the following: storage of additional programs, program editing, running of program from memory, machine and control diagnostics, special routines, inch/metric, incremental/absolute switchability.

CNC machines can be used as stand alone units or in a network of machines such as flexible machine centres. The controller uses a permanent resident program called an executive program to process the codes into the electrical pulses that control the machine. In any CNC machine, executive program resides in ROM and all the NC codes in RAM. The information in ROM is written into the electronic chips and cannot be erased and they become active whenever the machine is on. The contents in RAM are lost when the controller is turned off. Some use special type of RAM called CMOS memory, which retains its contents even when the power is turned off.

CNC milling machine

Direct Numerical Control

In a Direct Numerical Control system (DNC), a mainframe computer is used to coordinate the simultaneous operations of a number NC machines as shown in the figures 21.2 & 21.3. The main tasks performed by the computer are to program and edit part programs as well as download part programs to NC machines. Machine tool controllers have limited memory and a part program may contain few thousands of blocks.So the program is stored in a separate computer and sent directly to the machine, one block at a time. First DNC system developed was Molins System 24 in 1967 by Cincinnati Milacron and General Electric. They are now referred to as flexible manufacturing systems (FMS). The computers that were used at those times were quite expensive.



DNC system

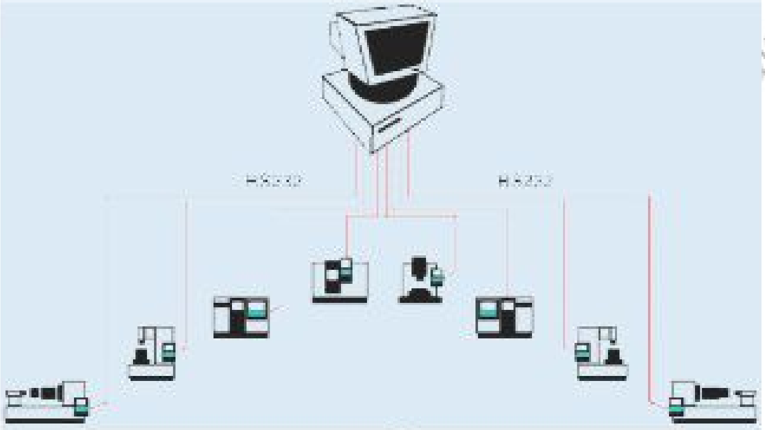


Figure 21.3: DNC system

Advantages & Disadvantages of CNC machine tools

Manually operated milling

Computer controlled machine milling machine

Some of the dominant advantages of the CNC machines are:

- CNC machines can be used continuously and only need to be switched off for occasional maintenance.
- These machines require less skilled people to operate unlike manual lathes / milling machines etc.
- CNC machines can be updated by improving the software used to drive the machines.
- Training for the use of CNC machines can be done through the use of 'virtual software'.
- The manufacturing process can be simulated virtually and no need to make a prototype or a model. This saves time and money.
- Once programmed, these machines can be left and do not require any human intervention, except for work loading and unloading.
- These machines can manufacture several components to the required accuracy without any fatigue as in the case of manually operated machines.
- Savings in time that could be achieved with the CNC machines are quite significant.

Some of the disadvantages of the CNC machines are:

CNC machines are generally more expensive than manually operated machines.
The CNC machine operator only needs basic training and skills, enough to supervise several machines.
Increase in electrical maintenance, high initial investment and high per hour operating costs than the traditional systems.
Fewer workers are required to operate CNC machines compared to manually operated machines. Investment in CNC machines can lead to unemployment.
Applications of NC/CNC machine tools

CNC was initially applied to metal working machinery: Mills, Drills, boring machines, punch presses etc and now expanded to robotics, grinders, welding machinery, EDM's, flame cutters and also for inspection equipment etc. The machines controlled by CNC can be classified into the following categories: CNC mills and machining centres.

CNC lathes and turning centers
CNC EDM
CNC grinding machines
CNC cutting machines (laser, plasma, electron, or flame)
CNC fabrication machines (sheet metal punch press, bending machine, or press brake)
CNC welding machines
CNC coordinate measuring machines

CNC Coordinate Measuring Machines:

A coordinate measuring machine is a dimensional measuring device, designed to move the measuring probe to determine the coordinates along the surface of the work piece. Apart from dimensional measurement, these machines are also used for profile measurement, angularity, digitizing or imaging.

A CMM consists of four main components: the machine, measuring probe, control system and the measuring software. The control system in a CMM performs the function of a live interaction between various machine drives, displacement transducers, probing systems and the peripheral devices. Control systems can be classified according to the following groups of CMMs.

1. Manually driven CMMs
2. Motorized CMMs with automatic probing systems
3. Direct computer controlled (DCC) CMMs
4. CMMs linked with CAD, CAM and FMS etc.

The first two methods are very common and self explanatory. In the case of DCC CMMs, the computer control is responsible for the movement of the slides, readout from displacement transducers and data communication. CMM are of different configurations-fixed bridge, moving bridge, cantilever arm figure 21.5(a), horizontal arm and gantry type CMM as shown in figure 21.5(b).

Figure 21.5(a) Cantilever type CMM

Figure 21.5(b) Gantry type CMM

CNC welding machines:

Figure 21.6 4 axis CNC Tig welding machine

The salient features of CNC welding machines are:

- Superior quality and weld precision.
- These machines are also equipped with rotary tables.
- Weld moves, welding feed rate, wire feed, torch heights & welding current can be programmed.
- CNC welding machines are used for laser welding, welding of plastics, submerged arc welding, wire welding machines, butt welding, flash butt welding etc.
- These machines are generally used in automobile work shops
- Cost of these machines will be twice than the conventional welding machines.

CNC EDM & WEDM machines:

EDM is a nontraditional machining method primarily used to machine hard metals that could not be machined by traditional machining methods. Material removal will be taking place by a series of electric arcs discharging across the gap between the electrode and the work piece. There are two main types- ram EDM & wire cut EDM. In wire-cut EDM, a thin wire is fed through the work piece and is constantly fed from a spool and is held between upper and lower guides. These guides move in the x-y plane and are precisely controlled by the CNC. Wire feed rate is also controlled by the CNC.

Figure 21.6 (a) Ram EDM

Figure 21.6 (b) Wire cut EDM

CLASSIFICATION OF CNC MACHINE TOOLS

(1) Based on the motion type ' **Point-to-point & Contouring systems**

There are two main types of machine tools and the control systems required for use with them differ because of the basic differences in the functions of the machines to be controlled. They are known as point-to-point and contouring controls.

(1.1) Point-to-point systems

Some machine tools for example drilling, boring and tapping machines etc, require the cutter and the work piece to be placed at a certain fixed relative positions at which they must remain while the cutter does its work. These machines are known as point-to-point machines as shown in figure 22.1 (a) and the control equipment for use with them are known as point-to-point control equipment. Feed rates need not to be programmed. In theses machine tools, each axis is driven separately. In a point-to-point control system, the dimensional information that must be given to the machine tool will be a series of required position of the two slides. Servo systems can be used to move the slides and no attempt is made to move the slide until the cutter has been retracted back.

(1.2) Contouring systems (Continuous path systems)

Other type of machine tools involves motion of work piece with respect to the cutter while cutting operation is taking place. These machine tools include milling, routing machines etc. and are known as contouring machines as shown in figure 22.1 (b) and the controls required for their control are known as contouring control.

Contouring machines can also be used as point-to-point machines, but it will be uneconomical to use them unless the work piece also requires having a contouring operation to be performed on it. These machines require simultaneous control of axes. In contouring machines, relative positions of the work piece and the tool should be continuously controlled.

The control system must be able to accept information regarding velocities and positions of the machines slides. Feed rates should be programmed.

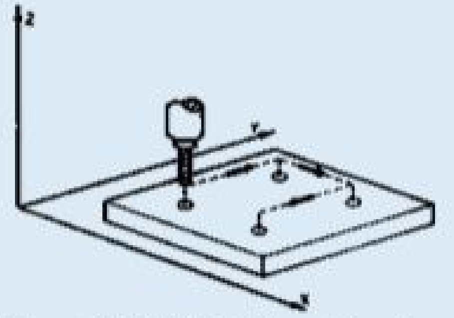


Figure 22.1 (a) Point-to-point system

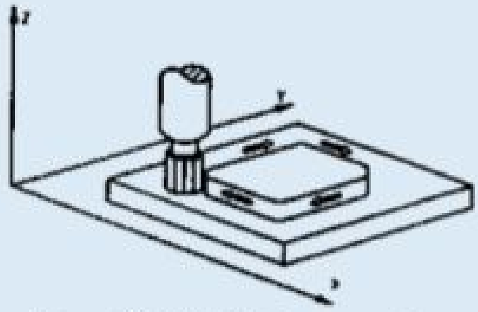


Figure 22.1 (b) Contouring system

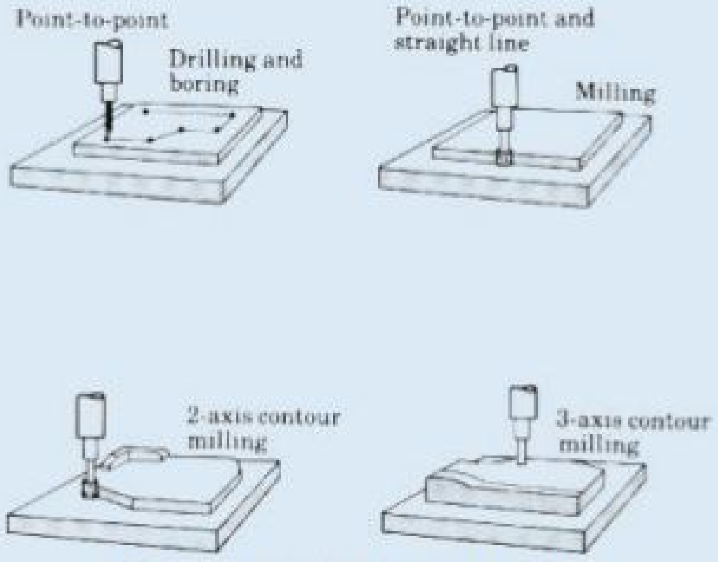


Figure 22.1 (c) Contouring systems

Based on the control loops ' Open loop & Closed loop systems
Open loop systems:

Programmed instructions are fed into the controller through an input device. These instructions are then converted to electrical pulses (signals) by the controller and sent to the servo amplifier to energize the servo motors. The primary drawback of the open-loop system is that there is no feedback system to check whether the program position and velocity has been achieved. If the system performance is affected by load, temperature, humidity, or lubrication then the actual output could deviate from the desired output. For these reasons the open -loop system is generally used in point-to-point systems where the accuracy requirements are not critical. Very few continuous-path systems utilize open-loop control.

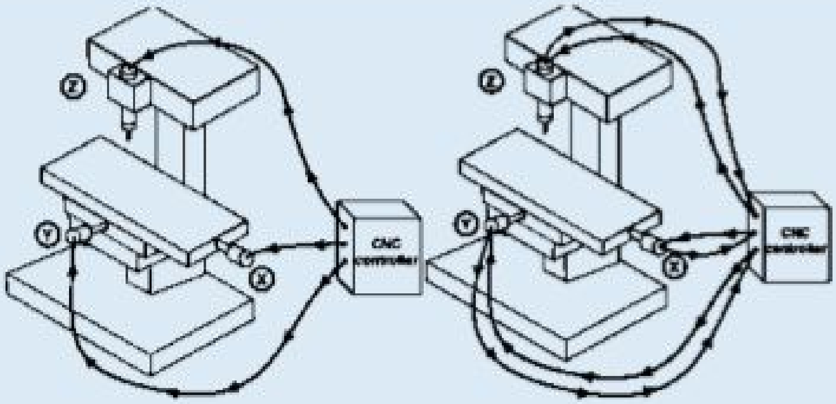


Figure 22.2 (a) Open loop control system Figure 22.2 (b) Closed loop control system

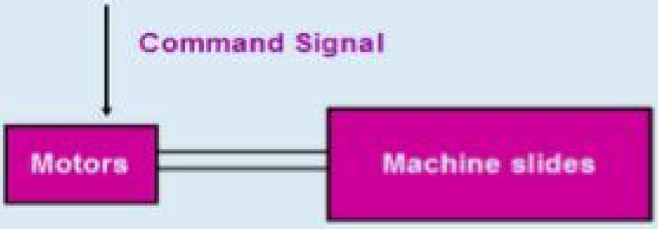


Figure 22.2 (c) Open loop system

Closed loop systems:

The closed-loop system has a feedback subsystem to monitor the actual output and correct any discrepancy from the programmed input. These systems use position and velocity feed back. The feedback system could be either analog or digital. The analog systems measure the variation of physical variables such as position and velocity in terms of voltage levels. Digital systems monitor output variations by means of electrical pulses. To control the dynamic behavior and the final position of the machine slides, a variety of position transducers are employed. Majority of CNC systems operate on servo mechanism, a closed loop principle. If a discrepancy is revealed between where the machine element should be and where it actually is, the sensing device signals the driving unit to make an adjustment, bringing the movable component to the required location.

Closed-loop systems are very powerful and accurate because they are capable of monitoring operating conditions through feedback subsystems and automatically compensating for any variations in real-time.

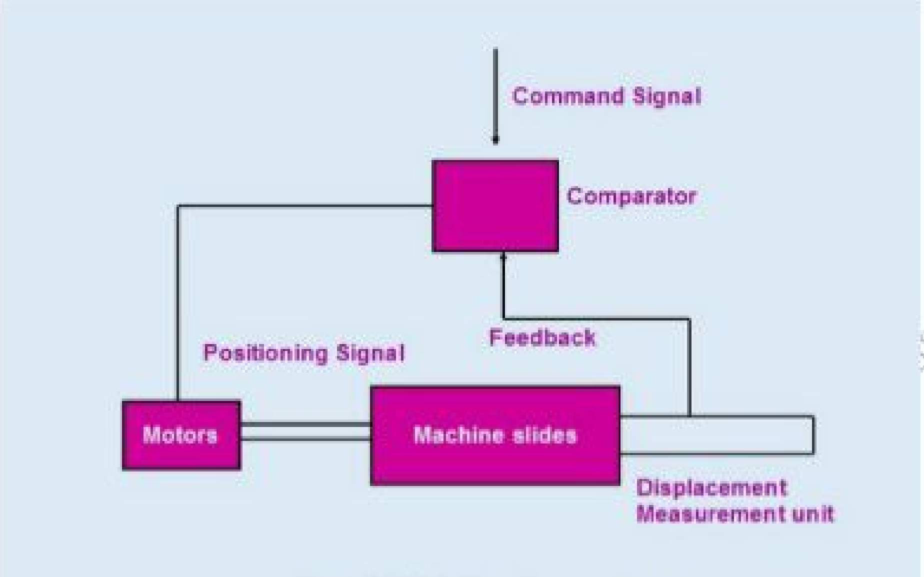


Figure 22.2 (d) Closed loop system

Based on the number of axes ' 2, 3, 4 & 5 axes CNC machines.

(3.1) 2& 3 axes CNC machines:

CNC lathes will be coming under 2 axes machines. There will be two axes along which motion takes place. The saddle will be moving longitudinally on the bed (Z-axis) and the cross slide moves transversely on the saddle (along X-axis). In 3-axes machines, there will be one more axis, perpendicular to the above two axes. By the simultaneous control of all the 3 axes, complex surfaces can be machined.

(3.2) 4 & 5 axes CNC machines:

4 and 5 axes CNC machines provide multi-axis machining capabilities beyond the standard 3-axis CNC tool path movements. A 5-axis milling centre includes the three X, Y, Z axes, the A axis which is rotary tilting of the spindle and the B-axis, which can be a rotary index table.

Figure 22.3 Five axes CNC machine

Five axes CNC machine

Importance of higher axes machining :
Reduced cycle time by machining complex components using a single setup. In addition to time savings, improved accuracy can also be achieved as positioning errors between setups are eliminated.

- Improved surface finish and tool life by tilting the tool to maintain optimum tool to part contact all the times.
- Improved access to under cuts and deep pockets. By tilting the tool, the tool can be made normal to the work surface and the errors may be reduced as the major component of cutting force will be along the tool axis.
- Higher axes machining has been widely used for machining sculptures surfaces in aerospace and automobile industry.

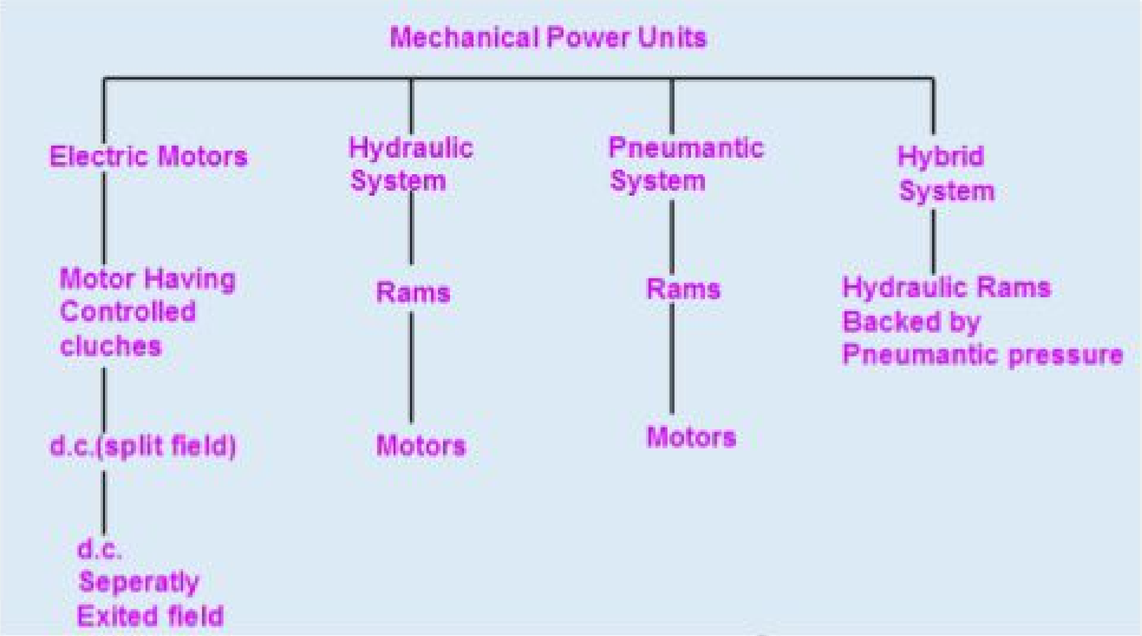
Turning centre:

Traditional centre lathes have horizontal beds. The saddle moves longitudinally and the cross slide moves transversely. Although the tools can be clearly seen, the operator must lean over the tool post to position them accurately. Concentration of chips may be creating a heat source and there may be temperature gradients in the machine tool. Keeping the above points in view, developments in the structure of the turning centres lead to the positioning the saddle and the cross slide behind the spindle on a slant bed as shown in the figure 22.4. Chips fall freely because of slant bed configuration which is more ergonomically acceptable from operator's point of view.

Figure 22.4 Slant bed turning centre

Based on the power supply ' Electric, Hydraulic & Pneumatic systems

Mechanical power unit refers to a device which transforms some form of energy to mechanical power which may be used for driving slides, saddles or gantries forming a part of machine tool. The input power may be of electrical, hydraulic or pneumatic.



22.4.1 Electric systems:

Electric motors may be used for controlling both positioning and contouring machines. They may be either a.c. or d.c. motor and the torque and direction of rotation need to be controlled. The speed of a d.c. motor can be controlled by varying either the field or the armature supply. The clutch-controlled motor can either be an a.c. or d.c. motor. They are generally used for small machine tools because of heat losses in the clutches. Split field motors are the simplest form of motors and can be controlled in a manner according to the machine tool. These are small and generally run at high maximum speeds and so require reduction gears of high ratio. Separately excited motors are used with control systems for driving the slides of large machinetools.

22.4.2 Hydraulic systems:

These hydraulic systems may be used with positioning and contouring machine tools of all sizes. These systems may be either in the form of rams or motors. Hydraulic motors are smaller than electric motors of equivalent power. There are several types of hydraulic motors. The advantage of using hydraulic motors is that they can be very small and have considerable torque. This means that they may be incorporated in servosystems which require having a rapid response.

Different components related to CNC machine tools

Any CNC machine tool essentially consists of the following parts:

(1.1) Part program:

A part program is a series of coded instructions required to produce a part. It controls the movement of the machine tool and on/off control of auxiliary functions such as spindle rotation and coolant. The coded instructions are composed of letters, numbers and symbols.

(1.2) Program input device:

The program input device is the means for part program to be entered into the CNC control. Three commonly used program input devices are punch tape reader, magnetic tape reader, and computer via RS-232-C communication.

(1.3) Machine Control Unit:

The machine control unit (MCU) is the heart of a CNC system. It is used to perform the following functions:

- To read the coded instructions.
- To decode the coded instructions.
- To implement interpolations (linear, circular, and helical) to generate axis motion commands.
- To feed the axis motion commands to the amplifier circuits for driving the axis mechanisms.
- To receive the feedback signals of position and speed for each drive axis.
- To implement auxiliary control functions such as coolant or spindle on/off and tool change.

(1.4) Drive System:

A drive system consists of amplifier circuits, drive motors, and ball lead-screws. The MCU feeds the control signals (position and speed) of each axis to the amplifier circuits. The control signals are augmented to actuate drive motors which in turn rotate the ball lead-screws to position the machine table.

(1.5) Machine Tool:

CNC controls are used to control various types of machine tools. Regardless of which type of machine tool is controlled, it always has a slide table and a spindle to control of position and speed. The machine table is controlled in the X and Y axes, while the spindle runs along the Z axis.

(1.6) Feed Back System:

The feedback system is also referred to as the measuring system. It uses position and speed transducers to continuously monitor the position at which the cutting tool is located at any particular instant. The MCU uses the difference between reference signals and feedback signals to generate the control signals for correcting position and speed errors.

ADVANTAGE

- CNC machines can be used continuously 24 hours a day, 365 days a year and only need to be switched off for occasional maintenance.
- CNC machines are programmed with a design which can then be manufactured hundreds or even thousands of times. Each manufactured product will be exactly the same.
- Less skilled/trained people can operate CNCs unlike manual lathes / milling machines etc.. which need skilled engineers.
- CNC machines can be updated by improving the software used to drive the machines
- Training in the use of CNCs is available through the use of 'virtual software'. This is software that allows the operator to practice using the CNC machine on the screen of a computer. The software is similar to a computer game.
- CNC machines can be programmed by advanced design software such as Pro/DESKTOP®, enabling the manufacture of products that cannot be made by manual machines, even those used by skilled designers / engineers.
- Modern design software allows the designer to simulate the manufacture of his/her idea. There is no need to make a prototype or a model. This saves time and money.
- One person can supervise many CNC machines as once they are programmed they can usually be left to work by themselves. Sometimes only the cutting tools need replacing occasionally.
- A skilled engineer can make the same component many times. However, if each component is carefully studied, each one will vary slightly. A CNC machine will manufacture each component as an exact match.

DISADVANTAGES

- CNC machines are more expensive than manually operated machines, although costs are slowly coming down.
- The CNC machine operator only needs basic training and skills, enough to supervise several machines. In years gone by, engineers needed years of training to operate centre lathes, milling machines and other manually operated machines. This means many of the old skills are been lost.
- Less workers are required to operate CNC machines compared to manually operated machines. Investment in CNC machines can lead to unemployment.
- Many countries no longer teach pupils / students how to use manually operated lathes / milling machines etc... Pupils / students no longer develop the detailed skills required by engineers of the past. These include mathematical and engineering skills.

APPLICATION

CNC machine is used

- *In the metal removal industry*
- *In the metal fabrication industry*
- *In the electrical discharge machining industry*
- *In the woodworking industry*
- *Laser welding in automobile industry*
- *Laser machining and Cutting*

Other Industries where CNC machines are used:

Many forms of lettering and engraving systems use CNC technology. Water jet machining uses a high pressure water jet stream to cut through plates of material. CNC is even used in the manufacturing of many electrical components. For example, there are CNC coil winders, and CNC terminal location and soldering.

CNC programming and introduction

- The part-program is a collection of all data required to produce the part. It is arranged in the form of blocks of information.

Each block contains the numerical data required for processing a segment of the work piece.

Part program is of two type:

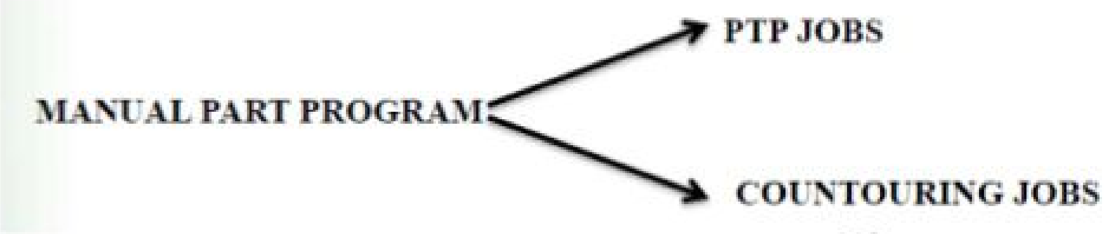
MANUAL PART PROGRAM

Dept. of mechanical Engineering, GIET,BBSR

Manual part program is written in a special set of instruction called manuscript.

The instruction are prepared in précised manner because the typist prepare the NC tape directly from the manuscript

Manuscript includes instructions and also other data such as other preparatory commands, miscellaneous instructions and speed/feed specifications all of which need to operate the machine under tape control .



COMPUTER ASSISTED PART PROGRAMMING

Programmers Activities:

Geometry workspace entry

Entry of sequence of operation

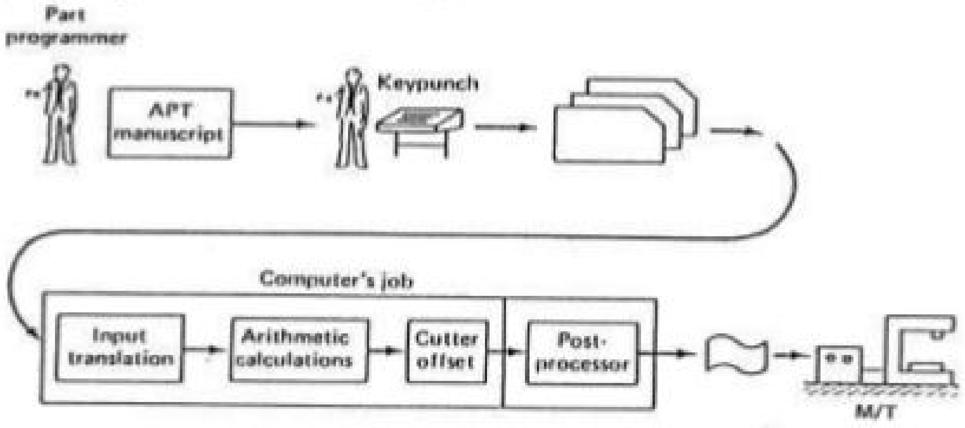
Computer Activities:

Input Translation

Advance computation

Cutting tool offset calculation

Post processing



Preparatory command (G code)

The G codes prepare the MCU for a given operation, typically involving a cutter motion.

G00 rapid motion, point-to-point positioning

G01 linear interpolation (generating a sloped or straight cut)

G06 parabolic interpolation (produces a segment of a parabola)

G17 XY plane selection

G20 circular interpolation

G28 automatic return to reference point

G33 thread cutting

Miscellaneous commands (M code)

M00 program stop
M03 start spindle rotation (cw)
M06 tool change
M07 turn coolant on

Feed commands (F code)

Used to specify the cutter feed rates in inch per minute.

Speed commands (S code)

Used to specify the spindle speed in rpm.

Tool commands (T code)

Specifies which tool to be used, machines with automatic tool changer.

Computer Tasks in Computer-Assisted Part Programming. The computer's role in computer-assisted part programming consists of the following tasks, performed more or less in the sequence noted: (1) input translation, (2) arithmetic and cutter offset computations, (3) editing, and (4) postprocessing. The first three tasks are carried out under the supervision of the language processing program. For example, the APT language uses a processor designed to interpret and process the words, symbols, and numbers written in APT. Other languages require their own processors. The fourth task, postprocessing, requires a separate computer program. The sequence and relationship of the tasks of the part programmer and the computer are portrayed in Figure 6.19.

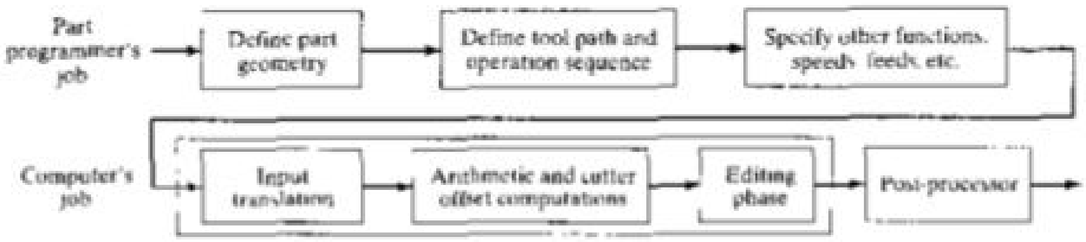


Figure 6.19 Tasks in computer-assisted part programming.

The part programmer enters the program using APT or some other high-level part programming language. The *input translation* module converts the coded instructions contained in the program into computer-usable form, preparatory to further processing. In APT, input translation accomplishes the following tasks: (1) syntax check of the input code to identify errors in format, punctuation, spelling, and statement sequence; (2) assigning a sequence number to each APT statement in the program; (3) converting geometry elements into a suitable form for computer processing; and (4) generating an intermediate file called PROFIL that is utilized in subsequent arithmetic calculations.

Part Programming with APT

In this section, we present some of the basic principles and vocabulary of the APT language. APT is an acronym that stands for Automatically Programmed Tooling. It is a three-dimensional NC part programming system that was developed in the late 1950s and early 60s (Historical Note 6.3). Today it remains an important and widely used language in the United States and around the world. APT is also important because many of the concepts incorporated into it formed the basis for other subsequently developed languages. APT was originally intended as a contouring language, but modern versions can be used for both point-to-point and contouring operations in up to five axes. Our discussion will be limited to the three linear axes, x, y, and z. APT can be used for a variety of machining operations. Our coverage will concentrate on drilling (point-to-point) and milling (contouring) operations. There are more than 500 words in the APT vocabulary. Only a small (but important) fraction of the total lexicon will be covered here. The Appendix to this chapter lists some of these important APT words.

APT is not only a language; it is also the computer program that processes the APT statements to calculate the corresponding cutter positions and generate the machine tool control commands. To program in APT, the part geometry must first be defined. Then the tool is directed to various point locations and along surfaces of the workpart to accomplish the required machining operations. The viewpoint of the programmer is that the workpiece remains stationary, and the tool is instructed to move relative to the part. To complete the program, speeds and feeds must be specified, tools must be called, tolerances must be given for circular interpolation, and so forth. Thus, there are four basic types of statements in the APT language:

1. *Geometry statements*, also called *definition statements*, are used to define the geometry elements that comprise the part.
2. *Motion commands* are used to specify the tool path.
3. *Postprocessor statements* control the machine tool operation, for example, to specify speeds and feeds, set tolerance values for circular interpolation, and actuate other capabilities of the machine tool.
4. *Auxiliary statements*, a group of miscellaneous statements used to name the part program, insert comments in the program and accomplish similar functions.

These statements are constructed of APT vocabulary words, symbols, and numbers, all arranged using appropriate punctuation. APT vocabulary words consist of six or fewer characters. The characters are almost always letters of the alphabet. Only a very few APT vocabulary words contain numerical digits-so few in fact that we will not encounter any of them in our treatment of APT in this chapter. Most APT statements include a slash (/) as part of the punctuation. APT vocabulary words that immediately precede the slash are called *major words*, whereas those that follow the slash are called *minor words*.

Geometry Statements. The geometry of the part must be defined to identify the surfaces and features that are to be machined. Accordingly, the points, lines, and surfaces must be defined in the program prior to specifying the motion statements. The general form of an APT geometry statement is the following:

$$\text{SYMBOL} = \text{GEOMETRY TYPE/descriptive data} \tag{6.3}$$

An example of such a statement is

$$\text{P1} = \text{POINT/20.0, 40.0, 60.0}$$

An APT geometry statement consists of three sections. The first is the symbol used to identify the geometry element. A symbol can be any combination of six or fewer alphabetical and numerical characters, at least one of which must be alphabetical. Also, the symbol cannot be an APT vocabulary word. Some examples are presented in Table 6.12 to illustrate what is permissible as a symbol and what is not. The second section of the APT geometry statement is an APT major word that identifies the type of geometry element. Examples are POINT, LINE, CIRCLE, and PLANE. The third section of the APT geometry statement provides the descriptive data that define the element precisely, completely, and uniquely. These data may include numerical values to specify dimensional and position data, previously defined geometry elements, and APT minor words.

Punctuation in an APT geometry statement is indicated in Eq. (6.3). The definition statement is written as an equation, the symbol being equated to the geometry element type, followed by a slash with descriptive data to the right of the slash. Commas are used to separate the words and numerical values in the descriptive data.

PART I
Automation and Control Technologies

chapter 3

Introduction to Automation

CHAPTER CONTENTS

Automation is the technology by which a process or procedure is accomplished without human assistance. It is implemented using a *program of instructions* combined with a *control system* that executes the instructions. To automate a process, *power* is required, both to drive the process itself and to operate the program and control system. Although automation can be applied in a wide variety of areas, it is most closely associated with the manufacturing industries. It was in the context of manufacturing that the term was originally coined by an engineering manager at Ford Motor Company in 1946 to describe the variety of automatic transfer devices and feed mechanisms that had been installed in Ford's production plants (Historical Note 3.1). It is ironic that nearly all modern applications of automation are controlled by computer technologies that were not available in 1946.

In this part of the book, we examine technologies that have been developed to automate manufacturing operations. The position of automation and control technologies in the larger production system is shown in Figure 3.1. In the present chapter, we provide an

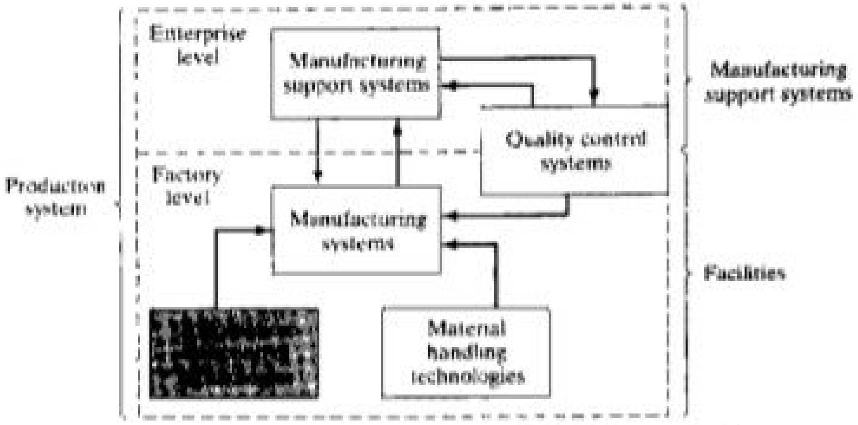


Figure 3.1 Automation and control technologies in the production system.

overview of automation: What are the elements of an automated system? What are some of the advanced features beyond the basic elements? And what are the levels in an enterprise where automation can be applied? In the following two chapters, we discuss industrial control systems and the hardware components of these systems. These two chapters serve as a foundation for the remaining chapters in our coverage of automation and control technologies. These technologies are: (1) numerical control (Chapter 6), (2) industrial robotics (Chapter 7), and (3) programmable logic controllers (Chapter 8).

Historical Note 3.1 History of automation[1]

The history of automation can be traced to the development of basic mechanical devices, such as the wheel (circa 3200 B.C.), lever, winch (circa 600 B.C.), cam (circa A.D. 1000), screw (A.D. 1405), and gear in ancient and medieval times. These basic devices were refined and used to construct the mechanisms in waterwheels, windmills (circa A.D. 650), and steam engines (A.D. 1765). These machines generated the power to operate other machinery of various kinds, such as flour mills (circa 85 B.C.), weaving machines (flying shuttle, 1733), machine tools (boring mill, 1775), steamboats (1787), and railroad locomotives (1803). *Power*, and the capacity to generate it and transmit it to operate a process, is one of the three basic elements of an automated system.

After his first steam engine in 1765, James Watt and his partner, Matthew Boulton, made several improvements in the design. One of the improvements was the flying-ball governor (around 1785), which provided feedback to control the throttle of the engine. The governor consisted of a ball on the end of a hinged lever attached to the rotating shaft. The lever was connected to the throttle valve. As the speed of the rotating shaft increased, the ball was forced to move outward by centrifugal force; this in turn caused the lever to reduce the valve opening and slow the motor speed. As rotational speed decreased, the ball and lever relaxed, thus allowing the valve to open. The flying-ball governor was one of the first examples in engineering of feedback control, an important type of *control system*—the second basic element of an automated system.

The third basic element of an automated system is for the actions of the system or machine to be directed by a *program of instructions*. One of the first examples of machine pro-

---

[1] Sources of most of the dates in this Historical Note: (1) R. Platt, ***Smithsonian Visual Timeline of Inventions*** (London: Dorling Kindersley Ltd., 1994); and (2) "The Power of Invention," ***Newsweek Special Issue***, Winter 1997–98 (pp. 6-79).

gramming was the Jacquard loom, invented around 1800. This loom was a machine for weaving cloth from yarn. The program of instructions that determined the weaving pattern of the cloth consisted of a metal plate containing holes. The hole pattern in the plate directed the shuttle motions of the loom, which in turn determined the weaving pattern. Different hole patterns yielded different cloth patterns. Thus, the Jacquard loom was a programmable machine, one of the first of its kind.

By the early 1800s, the three basic elements of automated systems—power source, controls, and programmable machines—had been developed, although these elements were primitive by today's standards. It took many years of refinement and many new inventions and developments, both in these basic elements as well as in the enabling infrastructure of the manufacturing industries, before fully automated production systems were to become a common reality. Important examples of these inventions and developments include *interchangeable parts* (circa 1800, Historical Note 2.1); *electrification* (starting in 1881); the *moving assembly line* (1913, Historical Note 17.1); mechanized *transfer lines* for mass production, whose programs were fixed by their hardware configuration (1924, Historical Note 18.1); a mathematical theory of *control systems* (1930s and 1940s); and the MARK I electromechanical *computer* at Harvard University (1944). These inventions and developments had all been realized by the end of World War II.

Since 1945, many new inventions and developments have contributed significantly to automation technology. Del Harder coined the word *automation* around 1946 in reference to the many automatic devices that the Ford Motor Company had developed for its production lines. The first electronic digital computer was developed at University of Pennsylvania in 1946. The first *numerical control* machine tool was developed and demonstrated in 1952 at Massachusetts Institute of Technology based on a concept proposed by John Parsons and Frank Stulen (Historical Note 6.1). By the late 1960s and early 1970s, digital computers were being connected to machine tools. In 1954, the first *industrial robot* was designed and patented (issued 1961) by George Devol (Historical Note 7.1). The first commercial robot was installed to unload parts in a die casting operation in 1961. In the late 1960s, the first *flexible manufacturing system* in the United States was installed at Ingersoll Rand Company to perform machining operations on a variety of parts (Historical Note 16.1). Around 1969, the first *programmable logic controller* was introduced (Historical Note 8.1). In 1978, the first commercial *personal computer* (PC) had been introduced by Apple Computer, although a similar product had been introduced in kit form as early as 1975.

Developments in computer technology were made possible by advances in electronics, including the *transistor* (1948), *hard disk* for computer memory (1956), *integrated circuits* (1960), the *microprocessor* (1971), *random access memory* (1984), megabyte capacity memory chips (circa 1990), and the *Pentium* microprocessors (1993). Software developments related to automation have been equally important, including the *FORTRAN* computer programming language (1955), the *APT* programming language for numerical control (NC) machine tools (1961), the *UNIX* operating system (1969), the *VAL* language for robot programming (1979), Microsoft *Windows* (1985), and the *JAVA* programming language (1995). Advances and enhancements in these technologies continue.

3.1 BASIC ELEMENTS OF AN AUTOMATED SYSTEM

An automated system consists of three basic elements: (1) *power* to accomplish the process and operate the system, (2) a *program of instructions* to direct the process, and (3) a *control system* to actuate the instructions. The relationship amongst these elements is illustrated in Figure 3.2. All systems that qualify as being automated include these three basic elements in one form or another.

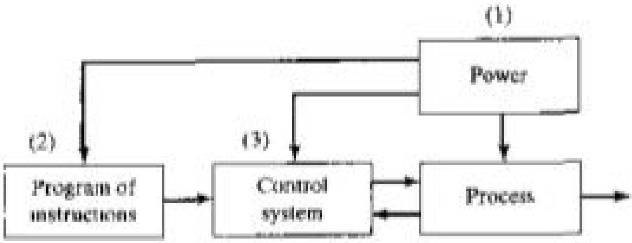


Figure 3.2 Elements of an automated system: (1) power, (2) program of instructions, and (3) control systems.

3.1.1 Power to Accomplish the Automated Process

An automated system is used to operate some process, and power is required to drive the process as well as the controls. The principal source of power in automated systems is electricity. Electric power has many advantages in automated as well as nonautomated processes:

- Electrical power is widely available at moderate cost. It is an important part of our industrial infrastructure.
- Electrical power can be readily converted to alternative energy forms: mechanical, thermal, light, acoustic, hydraulic, and pneumatic.
- Electrical power at low levels can be used to accomplish functions such as signal transmission, information processing, and data storage and communication.
- Electrical energy can be stored in long-life batteries for use in locations where an external source of electrical power is not conveniently available.

Alternative power sources include fossil fuels, solar energy, water, and wind. However, their exclusive use is rare in automated systems. In many cases when alternative power sources are used to drive the process itself, electrical power is used for the controls that automate the operation. For example, in casting or heat treatment, the furnace may be heated by fossil fuels, but the control system to regulate temperature and time cycle is electrical. In other cases, the energy from these alternative sources is converted to electric power to operate both the process and its automation. When solar energy is used as a power source for an automated system, it is generally converted in this way.

Power for the Process. In production, the term *process* refers to the manufacturing operation that is performed on a work unit. In Table 3.1, a list of common manufacturing processes is compiled along with the form of power required and the resulting action on the work unit. Most of the power in manufacturing plants is consumed by these kinds of operations. The "power form" indicated in the middle column of the table refers to the energy that is applied directly to the process. As indicated above, the power source for each operation is usually converted from electricity.

In addition to driving the manufacturing process itself, power is also required for the following material handling functions:

- *Loading and unloading the work unit.* All of the processes listed in Table 3.1 are accomplished on discrete parts. These parts must be moved into the proper position

TABLE 3.1 Common Manufacturing Processes and Their Power Requirements

Process	*Power Form*	*Action Accomplished*
Casting	Thermal	Melting the metal before pouring into a mold cavity where solidification occurs.
Electric discharge machining (EDM)	Electrical	Metal removal is accomplished by a series of discrete electrical discharges between electrode (tool) and workpiece. The electric discharges cause very high localized temperatures that melt the metal.
Forging	Mechanical	Metal workpart is deformed by opposing dies. Workparts are often heated in advance of deformation, thus thermal power is also required.
Heat treating	Thermal	Metallic work unit is heated to temperature below melting point to effect microstructural changes.
Injection molding	Thermal and mechanical	Heat is used to raise temperature of polymer to highly plastic consistency, and mechanical force is used to inject the polymer melt into a mold cavity.
Laser beam cutting	Light and thermal	A highly coherent light beam is used to cut material by vaporization and melting.
Machining	Mechanical	Cutting of metal is accomplished by relative motion between tool and workpiece.
Sheet metal punching and blanking	Mechanical	Mechanical power is used to shear metal sheets and plates.
Welding	Thermal (maybe mechanical)	Most welding processes use heat to cause fusion and coalescence of two (or more) metal parts at their contacting surfaces. Some welding processes also apply mechanical pressure to the surfaces.

and orientation for the process to be performed, and power is required for this transport and placement function. At the conclusion of the process, the work unit must similarly be removed. If the process is completely automated, then some form of mechanized power is used. If the process is manually operated or semiautomated, then human power may be used to position and locate the work unit.

- *Material transport between operations.* In addition to loading and unloading at a given operation, the work units must be moved between operations. We consider the material handling technologies associated with this transport function in Chapter 10.

Power for Automation. Above and beyond the basic power requirements for the manufacturing operation, additional power is required for automation. The additional power is used for the following functions:

- *Controller unit.* Modern industrial controllers are based on digital computers, which require electrical power to read the program of instructions, make the control calculations, and execute the instructions by transmitting the proper commands to the actuating devices.
- *Power to actuate the control signals.* The commands sent by the controller unit are carried out by means of electromechanical devices, such as switches and motors, called *actuators* (Section 5.2). The commands are generally transmitted by means of low-voltage control signals. To accomplish the commands, the actuators require more power,

and so the control signals must be amplified to provide the proper power level for the actuating device.

- *Data acquisition and information processing*. In most control systems, data must be collected from the process and used as input to the control algorithms. In addition, a requirement of the process may include keeping records of process performance or product quality. These data acquisition and record keeping functions require power, although in modest amounts.

3.1.2 Program of Instructions

The actions performed by an automated process are defined by a program of instructions. Whether the manufacturing operation involves low, medium, or high production (Section 1.1), each part or product style made in the operation requires one or more processing steps that are unique to that style. These processing steps are performed during a work cycle. A new part is completed during each work cycle (in some manufacturing operations, more than one part is produced during the work cycle; e.g., a plastic injection molding operation may produce multiple parts each cycle using a multiple cavity mold). The particular processing steps for the work cycle are specified in a *work cycle program*. Work cycle programs are called *part programs* in numerical control (Chapter 6). Other process control applications use different names for this type of program.

Work Cycle Programs. In the simplest automated processes, the work cycle consists of essentially one step, which is to maintain a single process parameter at a defined level, for example, maintain the temperature of a furnace at a designated value for the duration of a heat treatment cycle. (We assume that loading and unloading of the work units into and from the furnace is performed manually and is therefore not part of the automatic cycle.) In this case, programming simply involves setting the temperature dial on the furnace. To change the program, the operator simply changes the temperature setting. An extension of this simple case is when the single-step process is defined by more than one process parameter, for example, a furnace in which both temperature and atmosphere are controlled.

In more complicated systems, the process involves a work cycle consisting of multiple steps that are repeated with no deviation from one cycle to the next. Most discrete part manufacturing operations are in this category. A typical sequence of steps (simplified) is: (1) load the part into the production machine, (2) perform the process, and (3) unload the part. During each step, there are one or more activities that involve changes in one or more process parameters. *Process parameters* are inputs to the process, such as temperature setting of a furnace, coordinate axis value in a positioning system, valve opened or closed in a fluid flow system, and motor on or off. Process parameters are distinguished from *process variables*, which are outputs from the process; for example, the actual temperature of the furnace, the actual position of the axis, the actual flow rate of the fluid in the pipe, and the rotational speed of the motor. As our list of examples suggests, the changes in process parameter values may be continuous (gradual changes during the processing step; for example, gradually increasing temperature during a heat treatment cycle) or discrete (stepwise changes; for example, on/off). Different process parameters may be involved in each step.

EXAMPLE 3.1 An Automated Turning Operation

Consider an automated turning operation in which a cone-shaped geometry is generated. Assume the system is automated and that a robot is used to load and unload the work unit. The work cycle consists of the following steps: (1) load starting workpiece, (2) position cutting tool prior to turning, (3) turn, (4) reposition tool to a safe location at end of turning, and (5) unload finished workpiece. Identify the activity(ies) and process parameter(s) in each step of the operation.

Solution: In step (1), the activities consist of the robot manipulator reaching for the raw workpart, lifting and positioning the part into the chuck jaws of the lathe, then removing the manipulator to a safe position to await unloading. The process parameters for these activities are the axis values of the robot manipulator (which change continuously), the gripper value (open or closed), and the chuck jaw value (open or closed).

In step (2), the activity involves the movement of the cutting tool to a "ready" position. The process parameters associated with this activity are the x- and z-axis position of the tool.

Step (3) is the turning operation. It requires the simultaneous control of three process parameters: rotational speed of the workpiece (rev/min), feed (mm/rev), and radial distance of the cutting tool from the axis of rotation. To cut the conical shape, radial distance must be changed continuously at a constant rate for each revolution of the workpiece. For a consistent finish on the surface, the rotational speed must be continuously adjusted to maintain a constant surface speed (m/min); and for equal feed marks on the surface, the feed must be set at a constant value. Depending on the angle of the cone, multiple turning passes may be required to gradually generate the desired contour. Each pass represents an additional step in the sequence.

Steps (4) and (5) involve the reverse activities as steps (2) and (1), respectively, and the process parameters are the same.

Many production operations consist of multiple steps, sometimes more complicated than our turning example. Examples of these operations include automatic screw machine cycles, sheet metal stamping operations, plastic injection molding, and die casting. Each of these manufacturing processes has been used for many decades. In earlier versions of these operations, the work cycles were controlled by hardware components, such as limit switches, timers, cams, and electromechanical relays. In effect, the hardware components and their arrangements served as the program of instructions that directed the sequence of steps in the processing cycle. Although these devices were quite adequate in performing their sequencing function, they suffered from the following disadvantages: (1) They often required considerable time to design and fabricate, thus forcing the production equipment to be used for batch production only; (2) making even minor changes in the program was difficult and time consuming; and (3) the program was in a physical form that is not readily compatible with computer data processing and communication.

Modern controllers used in automated systems are based on digital computers. Instead of cams, timers, relays, and other hardware devices, the programs for computer-controlled equipment are contained in magnetic tape, diskettes, compact disks (CD-ROMs), computer memory, and other modern storage technologies. Virtually all new equipment that perform the above mass production operations are designed with some type of computer controller to execute their respective processing cycles. The use of digital computers as the process controller allows improvements and upgrades to be made in the control programs, such as the addition of control functions not foreseen during initial equipment design. These kinds of control changes are often difficult to make with the previous hardware devices.

The work cycle may include manual steps, where the operator performs certain activities during the work cycle, and the automated system performs the rest. A common example is the loading and unloading of parts by the operator into and from a numerical control machine between machining cycles, where the machine performs the cutting operation under part program control. Initiation of the cutting operation of each cycle is triggered by the operator activating a "start" button after the part has been loaded.

Decision-Making in the Programmed Work Cycle. In our previous discussion of automated work cycles, the only two features of the work cycle are (1) the number and sequence of processing steps and (2) the process parameter changes in each step. Each work cycle consists of the same steps and associated process parameter changes with no variation from one cycle to the next. The program of instructions is repeated each work cycle without deviation. In fact, many automated manufacturing operations require decisions to be made during the programmed work cycle to cope with variations in the cycle. In many cases, the variations are routine elements of the cycle, and the corresponding instructions for dealing with them are incorporated into the regular part program. These cases include:

- *Operator interaction.* Although the program of instructions is intended to be carried out without human interaction, the controller unit may require input data from a human operator in order to function. For example, in an automated engraving operation, the operator may have to enter the alphanumeric characters that are to be engraved on the work unit (e.g., plaque, trophy, belt buckle). Having entered the characters, the engraving operation is accomplished automatically by the system. (An everyday example of operator interaction with an automated system is a bank customer using an automated teller machine. The customer must enter the codes indicating what transaction is to be accomplished by the teller machine.)
- *Different part or product styles processed by the system.* In this instance, the automated system is programmed to perform different work cycles on different part or product styles. An example is an industrial robot that performs a series of spot welding operations on car bodies in a final assembly plant. These plants are often designed to build different body styles on the same automated assembly line, such as two-door and four-door sedans. As each car body enters a given welding station on the line, sensors identify which style it is, and the robot performs the correct series of welds for that style.
- *Variations in the starting work units.* In many manufacturing operations the starting work units are not consistent. A good example is a sand casting as the starting work unit in a machining operation. The dimensional variations in the raw castings sometimes necessitate an extra machining pass to bring the machined dimension to the specified value. The part program must be coded to allow for the additional pass when necessary.

In all of these examples, the routine variations can be accommodated in the regular work cycle program. The program can be designed to respond to sensor or operator inputs by executing the appropriate subroutine corresponding to the input. In other cases, the variations in the work cycle are not routine at all. They are infrequent and unexpected, such as the failure of an equipment component. In these instances, the program must include contingency procedures or modifications in the sequence to cope with conditions that lie outside the normal routine. We discuss these measures later in the chapter in the context of advanced automation functions (Section 3.2).

TABLE 3.2 Features of Work Cycle Programs Used in Automated Systems

Program Feature	*Examples or Alternatives*
Steps in work cycle	Example: • Typical sequence of steps: (1) load, (2), process, (3) unload
Process parameters (inputs) in each step	Alternatives: • One parameter versus multiple parameters that must be changed during the step • Continuous parameters versus discrete parameters • Parameters that change during the step; for example, a positioning system whose axes values change during the processing step
Manual steps in work cycle	Alternatives: • Manual steps versus no manual steps (completely automated work cycle) Example: • Operator loading and unloading parts to and from machine
Operator interaction	Alternatives: • Operator interaction versus completely automated work cycle Example: • Operator entering processing information for current workpart
Different part or product styles	Alternatives: • Identical part or product style each cycle (mass or batch production) versus different part or product styles each cycle (flexible automation)
Variations in starting work units	Example: • Variations in starting dimensions or part features

A variety of production situations and work cycle programs has been discussed here. The features of work cycle programs (part programs) used to direct the operations of an automated system are summarized as in Table 3.2.

3.1.3 Control System

The control element of the automated system executes the program of instructions. The control system causes the process to accomplish its defined function, which for our purpose is to carry out some manufacturing operation. Let us provide a brief introduction to control systems here. The following chapter describes this important industrial technology in more detail.

The controls in an automated system can be either closed loop or open loop. A *closed loop control system*, also known as a *feedback control system*, is one in which the output variable is compared with an input parameter, and any difference between the two is used to drive the output into agreement with the input. As shown in Figure 3.3, a closed loop control system consists of six basic elements: (1) input parameter, (2) process, (3) output variable, (4) feedback sensor, (5) controller, and (6) actuator. The *input parameter*, often referred to as the *set point*, represents the desired value of the output. In a home temperature control system, the set point is the desired thermostat setting. The *process* is the operation or function being controlled. In particular, it is the *output variable* that is being controlled in the loop. In the present discussion, the process of interest is usually a manufacturing operation, and the output variable is some process variable, perhaps a critical performance

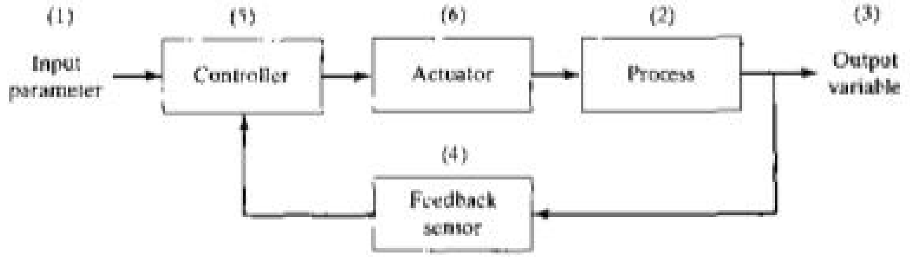


Figure 3.3 A feedback control system.

measure in the process, such as temperature or force or flow rate. A *sensor* is used to measure the output variable and close the loop between input and output. Sensors perform the feedback function in a closed loop control system. The *controller* compares the output with the input and makes the required adjustment in the process to reduce the difference between them. The adjustment is accomplished using one or more *actuators*, which are the hardware devices that physically carry out the control actions, such as an electric motor or a flow valve. It should be mentioned that our model in Figure 3.3 shows only one loop. Most industrial processes require multiple loops, one for each process variable that must be controlled.

In contrast to the closed loop control system, an *open loop control system* operates without the feedback loop, as in Figure 3.4. In this case, the controls operate without measuring the output variable, so no comparison is made between the actual value of the output and the desired input parameter. The controller relies on an accurate model of the effect of its actuator on the process variable. With an open loop system, there is always the risk that the actuator will not have the intended effect on the process, and that is the disadvantage of an open loop system. Its advantage is that it is generally simpler and less expensive than a closed loop system. Open loop systems are usually appropriate when the following conditions apply: (1) The actions performed by the control system are simple, (2) the actuating function is very reliable, and (3) any reaction forces opposing the actuation are small enough to have no effect on the actuation. If these characteristics are not applicable, then a closed loop control system may be more appropriate.

Consider the difference between a closed loop and open loop system for the case of a positioning system. Positioning systems are common in manufacturing to locate a workpart relative to a tool or workhead. Figure 3.5 illustrates the case of a closed loop posi-

Figure 3.4 An open loop control system.

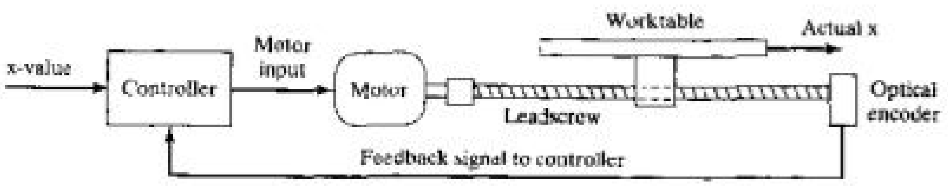


Figure 3.5 A (one-axis) positioning system consisting of a leadscrew driven by a dc servomotor.

tioning system. In operation, the system is directed to move the worktable to a specified location as defined by a coordinate value in a Cartesian (or other) coordinate system. Most positioning systems have at least two axes (e.g., an $x - y$ positioning table) with a control system for each axis, but our diagram only illustrates one of these axes. A dc servomotor connected to a leadscrew is a common actuator for each axis. A signal indicating the coordinate value (e.g., x-value) is sent from the controller to the motor that drives the leadscrew, whose rotation is converted into linear motion of the positioning table. As the table moves closer to the desired x-coordinate value, the difference between the actual x-position and the input x-value is reduced. The actual x-position is measured by a feedback sensor (e.g., an optical encoder). The controller continues to drive the motor until the actual table position corresponds to the input position value.

For the open loop case, the diagram for the positioning system would be similar to the preceding, except that no feedback loop is present and a stepper motor is used in place of the dc servomotor. A stepper motor is designed to rotate a precise fraction of a turn for each pulse received from the controller. Since the motor shaft is connected to the leadscrew, and the leadscrew drives the worktable, each pulse converts into a small constant linear movement of the table. To move the table a desired distance, the number of pulses corresponding to that distance is sent to the motor. Given the proper application, whose characteristics match the preceding list of operating conditions, an open loop positioning system works with high reliability.

We consider the engineering analysis of closed loop and open loop positioning systems in the context of numerical control in a subsequent chapter (Section 6.6).

3.2 ADVANCED AUTOMATION FUNCTIONS

In addition to executing work cycle programs, an automated system may be capable of executing advanced functions that are not specific to a particular work unit. In general, the functions are concerned with enhancing the performance and safety of the equipment. Advanced automation functions include the following: (1) safety monitoring, (2) maintenance and repair diagnostics, and (3) error detection and recovery.

Advanced automation functions are made possible by special subroutines included in the program of instructions. In some cases, the functions provide information only and do not involve any physical actions by the control system. An example of this case includes reporting a list of preventive maintenance tasks that should be accomplished. Any actions taken on the basis of this report are decided by the human operators and managers of the system and not by the system itself. In other cases, the program of instructions must be physically executed by means of the control system using available actuators. A simple example of this case is a safety monitoring system that sounds an alarm when a human worker gets dangerously close to the automated system.

3.2.1 Safety Monitoring

One of the significant reasons for automating a manufacturing operation is to remove worker(s) from a hazardous working environment. An automated system is often installed to perform a potentially dangerous operation that would otherwise be accomplished manually by human workers. However, even in automated systems, workers are still needed to service the system, at periodic time intervals if not full-time. Accordingly, it is important that the automated system be designed to operate safely when workers are in attendance. In addition, it is essential that the automated system carry out its process in a way that is not self-destructive. Thus, there are two reasons for providing an automated system with a safety monitoring capability: (1) to protect human workers in the vicinity of the system and (2) to protect the equipment associated with the system.

Safety monitoring means more than the conventional safety measures taken in a manufacturing operation, such as protective shields around the operation or the kinds of manual devices that might be utilized by human workers, such as emergency stop buttons. *Safety monitoring* in an automated system involves the use of sensors to track the system's operation and identify conditions and events that are unsafe or potentially unsafe. The safety monitoring system is programmed to respond to unsafe conditions in some appropriate way. Possible responses to various hazards might include one or more of the following:

- complete stoppage of the automated system
- sounding an alarm
- reducing the operating speed of the process
- taking corrective actions to recover from the safety violation

This last response is the most sophisticated and is suggestive of an intelligent machine performing some advanced strategy. This kind of response is applicable to a variety of possible mishaps, not necessarily confined to safety issues, and is called error detection and recovery (Section 3.2.3).

Sensors for safety monitoring range from very simple devices to highly sophisticated systems. The topic of sensor technology is discussed in Chapter 5 (Section 5.1). The following list suggests some of the possible sensors and their applications for safety monitoring:

- Limit switches to detect proper positioning of a part in a workholding device so that the processing cycle can begin.
- Photoelectric sensors triggered by the interruption of a light beam; this could be used to indicate that a part is in the proper position or to detect the presence of a human intruder into the work cell.
- Temperature sensors to indicate that a metal workpart is hot enough to proceed with a hot forging operation. If the workpart is not sufficiently heated, then the metal's ductility may be too low, and the forging dies might be damaged during the operation.
- Heat or smoke detectors to sense fire hazards.
- Pressure-sensitive floor pads to detect human intruders into the work cell.
- Machine vision systems to supervise the automated system and its surroundings.

It should be mentioned that a given safety monitoring system is limited in its ability to respond to hazardous conditions by the possible irregularities that have been foreseen by the system designer. If the designer has not anticipated a particular hazard, and consequently has not provided the system with the sensing capability to detect that hazard, then the safety monitoring system cannot recognize the event if and when it occurs.

3.2.2 Maintenance and Repair Diagnostics

Modern automated production systems are becoming increasingly complex and sophisticated, thus complicating the problem of maintaining and repairing them. *Maintenance and repair diagnostics* refers to the capabilities of an automated system to assist in the identification of the source of potential or actual malfunctions and failures of the system. Three modes of operation are typical of a modern maintenance and repair diagnostics subsystem:

1. *Status monitoring.* In the status monitoring mode, the diagnostic subsystem monitors and records the status of key sensors and parameters of the system during normal operation. On request, the diagnostics subsystem can display any of these values and provide an interpretation of current system status, perhaps warning of an imminent failure.
2. *Failure diagnostics.* The failure diagnostics mode is invoked when a malfunction or failure occurs. Its purpose is to interpret the current values of the monitored variables and to analyze the recorded values preceding the failure so that the cause of the failure can be identified.
3. *Recommendation of repair procedure.* In the third mode of operation, the subsystem provides a recommended procedure to the repair crew as to the steps that should be taken to effect repairs. Methods for developing the recommendations are sometimes based on the use of expert systems in which the collective judgments of many repair experts are pooled and incorporated into a computer program that uses artificial intelligence techniques.

Status monitoring serves two important functions in machine diagnostics: (1) providing information for diagnosing a current failure and (2) providing data to predict a future malfunction or failure. First, when a failure of the equipment has occurred, it is usually difficult for the repair crew to determine the reason for the failure and what steps should be taken to make repairs. It is often helpful to reconstruct the events leading up to the failure. The computer is programmed to monitor and record the variables and to draw logical inferences from their values about the reason for the malfunction. This diagnosis helps the repair personnel make the necessary repairs and replace the appropriate components. This is especially helpful in electronic repairs where it is often difficult to determine on the basis of visual inspection which components have failed.

The second function of status monitoring is to identify signs of an impending failure, so that the affected components can be replaced before failure actually causes the system to go down. These part replacements can be made during the night shift or other time when the process is not operating, with the result that the system experiences no loss of regular operation.

3.2.3 Error Detection and Recovery

In the operation of any automated system, there are hardware malfunctions and unexpected events that occur during operation. These events can result in costly delays and loss of production until the problem has been corrected and regular operation is restored. Traditionally, equipment malfunctions are corrected by human workers, perhaps with the aid of a maintenance and repair diagnostics subroutine. With the increased use of computer control for manufacturing processes, there is a trend toward using the control computer not only to diagnose the malfunctions but also to automatically take the necessary corrective action to restore the system to normal operation. The term *error detection and recovery* is used when the computer performs these functions.

Error Detection. As indicated by the term, error detection and recovery consists of two steps: (1) error detection and (2) error recovery. The *error detection* step uses the automated system's available sensor systems to determine when a deviation or malfunction has occurred, correctly interpret the sensor signal(s), and classify the error. Design of the error detection subsystem must begin with a classification of the possible errors that can occur during system operation. The errors in a manufacturing process tend to be very application specific. They must be anticipated in advance in order to select sensors that will enable their detection.

In analyzing a given production operation, the possible errors can be classified into one of three general categories: (1) random errors, (2) systematic errors, and (3) aberrations. *Random errors* occur as a result of the normal stochastic nature of the process. These errors occur when the process is in *statistical control* (Section 21.1). Large variations in part dimensions, even when the production process is in statistical control, can cause problems in downstream operations. By detecting these deviations on a part-by-part basis, corrective action can be taken in subsequent operations. *Systematic errors* are those that result from some assignable cause such as a change in raw material properties or a drift in an equipment setting. These errors usually cause the product to deviate from specifications so as to be unacceptable in quality terms. Finally, the third type of error, *aberrations*, results from either an equipment failure or a human mistake. Examples of equipment failures include fracture of a mechanical shear pin, bursts in a hydraulic line, rupture of a pressure vessel, and sudden failure of a cutting tool. Examples of human mistakes include errors in the control program, improper fixture setups, and substitution of the wrong raw materials.

The two main design problems in error detection are: (1) to anticipate all of the possible errors that can occur in a given process and (2) to specify the appropriate sensor systems and associated interpretive software so that the system is capable of recognizing each error. Solving the first problem requires a systematic evaluation of the possibilities under each of the three error classifications. If the error has not been anticipated, then the error detection subsystem cannot correctly detect and identify it.

EXAMPLE 3.2 Error Detection in an Automated Machining Cell

Consider an automated cell consisting of a CNC machine tool, a parts storage unit, and a robot for loading and unloading the parts between the machine and the storage unit. Possible errors that might affect this system can be divided into the following categories: (1) machine and process, (2) cutting tools, (3) workholding fixture, (4) part storage unit, and (5) load/unload robot. Develop a list of possible errors (deviations and malfunctions) that might be included in each of these five categories.

Solution: A list of possible errors in the machining cell is presented in Table 3.3.

Error Recovery. *Error recovery* is concerned with applying the necessary corrective action to overcome the error and bring the system back to normal operation. The problem of designing an error recovery system focuses on devising appropriate strategies and procedures that will either correct or compensate for the variety of errors that can occur in the process. Generally, a specific recovery strategy and procedure must be designed for each different error. The types of strategies can be classified as follows:

1. *Make adjustments at the end of the current work cycle.* When the current work cycle is completed, the part program branches to a corrective action subroutine specifically

TABLE 3.3 Error Detection Step in an Automated Machining Cell: Error Categories and Possible Malfunctions Within Each Category

Error Categories	*Possible Malfunctions*
1. Machine and process	Loss of power, power overload, thermal deflection, cutting temperature too high, vibration, no coolant, chip fouling, wrong part program, defective part
2. Cutting tools	Tool breakage, tool wear-out, vibration, tool not present, wrong tool
3. Workholding fixture	Part not in fixture, clamps not actuated, part dislodged during machining, part deflection during machining, part breakage, chips causing location problems
4. Part storage unit	Workpart not present, wrong workpart, oversized or undersized workpart
5. Load/unload robot	Improper grasping of workpart, robot drops workpart, no part present at pickup

designed for the error detected, executes the subroutine, and then returns to the work cycle program. This action reflects a low level of urgency and is most commonly associated with random errors in the process.

2. *Make adjustments during the current cycle.* This generally indicates a higher level of urgency than the preceding type. In this case, the action to correct or compensate for the detected error is initiated as soon as the error is detected. However, it must be possible to accomplish the designated corrective action while the work cycle is still being executed.
3. *Stop the process to invoke corrective action.* In this case, the deviation or malfunction requires that the execution of the work cycle be suspended during corrective action. It is assumed that the system is capable of automatically recovering from the error without human assistance. At the end of the corrective action, the regular work cycle is continued.
4. *Stop the process and call for help.* In this case, the error requiring stoppage of the process cannot be resolved through automated recovery procedures. This situation arises because: (1) the automated cell is not enabled to correct the problem or (2) the error cannot be classified into the predefined list of errors. In either case, human assistance is required to correct the problem and restore the system to fully automated operation.

Error detection and recovery requires an interrupt system (Section 4.3.2). When an error in the process is sensed and identified, an interrupt in the current program execution is invoked to branch to the appropriate recovery subroutine. This is done either at the end of the current cycle (type 1 above) or immediately (types 2, 3, and 4). At the completion of the recovery procedure, program execution reverts back to normal operation.

EXAMPLE 3.3 Error Recovery in an Automated Machining Cell

For the automated cell of Example 3.2, develop a list of possible corrective actions that might be taken by the system to address certain of the errors.

Solution: A list of possible corrective actions is presented in Table 3.4.

TABLE 3.4 Error Recovery in an Automated Machining Cell: Possible Corrective Actions That Might Be Taken in Response to Errors Detected During the Operation

Errors Detected	*Possible Corrective Actions to Recover*
Part dimensions deviating due to thermal deflection of machine tool	Adjust coordinates in part program to compensate (category 1 corrective action)
Part dropped by robot during pickup	Reach for another part (category 2 corrective action)
Part is dimensionally oversized	Adjust part program to take a preliminary machining pass across the work surface (category 2 corrective action)
Chatter (tool vibration)	Increase or decrease cutting speed to change harmonic frequency (category 2 corrective action)
Cutting temperature too high	Reduce cutting speed (category 2 corrective action)
Failure of cutting tool	Replace cutting tool with another sharp tool (category 3 corrective action).
No more parts in parts storage unit	Call operator to resupply starting workparts (category 4 corrective action)
Chips fouling machining operation	Call operator to clear chips from work area (category 4 corrective action)

3.3 LEVELS OF AUTOMATION

The concept of automated systems can be applied to various levels of factory operations. One normally associates automation with the individual production machines. However, the production machine itself is made up of subsystems that may themselves be automated. For example, one of the important automation technologies we discuss in this part of the book is numerical control (Chapter 6). A modern numerical control (NC) machine tool is an automated system. However, the NC machine itself is composed of multiple control systems. Any NC machine has at least two axes of motion, and some machines have up to five axes. Each of these axes operates as a positioning system, as described in Section 3.1.3, and is, in effect, itself an automated system. Similarly, a NC machine is often part of a larger manufacturing system, and the larger system may itself be automated. For example, two or three machine tools may be connected by an automated part handling system operating under computer control. The machine tools also receive instructions (e.g., part programs) from the computer. Thus we have three levels of automation and control included here (the positioning system level, the machine tool level, and the manufacturing system level). For our purposes in this text, we can identify five possible levels of automation in a production plant. They are defined next, and their hierarchy is depicted in Figure 3.6.

1. *Device level.* This is the lowest level in our automation hierarchy. It includes the actuators, sensors, and other hardware components that comprise the machine level. The devices are combined into the individual control loops of the machine; for example, the feedback control loop for one axis of a CNC machine or one joint of an industrial robot.
2. *Machine level.* Hardware at the device level is assembled into individual machines. Examples include CNC machine tools and similar production equipment, industrial robots, powered conveyors, and automated guided vehicles. Control functions at this

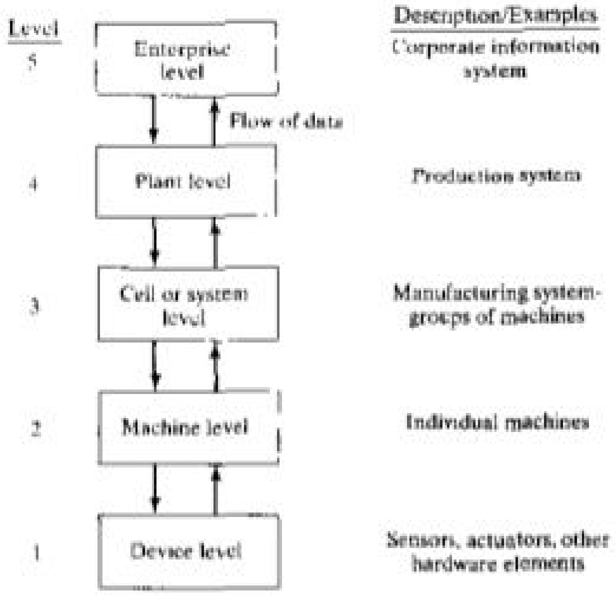


Figure 3.6 Five levels of automation and control in manufacturing.

level include performing the sequence of steps in the program of instructions in the correct order and making sure that each step is properly executed.

3. *Cell or system level.* This is the manufacturing cell or system level, which operates under instructions from the plant level. A manufacturing cell or system is a group of machines or workstations connected and supported by a material handling system, computer, and other equipment appropriate to the manufacturing process. Production lines are included in this level. Functions include part dispatching and machine loading, coordination among machines and material handling system, and collecting and evaluating inspection data.
4. *Plant level.* This is the factory or production systems level. It receives instructions from the corporate information system and translates them into operational plans for production. Likely functions include: order processing, process planning, inventory control, purchasing, material requirements planning, shop floor control, and quality control.
5. *Enterprise level.* This is the highest level, consisting of the corporate information system. It is concerned with all of the functions necessary to manage the company: marketing and sales, accounting, design, research, aggregate planning, and master production scheduling.

Most of the technologies discussed in this part of the book are at level 2 (the machine level), although we discuss level 1 automation technologies (the devices that make up a control system) in Chapter 5. The level 2 technologies include the individual controllers (e.g., programmable logic controllers and digital computer controllers), numerical control machines, and industrial robots. The material handling equipment discussed in Part II also represent technologies at level 2, although some of the handling equipment are themselves sophisticated automated systems. The automation and control issues at level 2 are concerned with the basic operation of the equipment and the physical processes they perform.

Controllers, machines, and material handling equipment are combined into manufacturing cells, or production lines, or similar systems, which make up level 3, considered in Part III. A *manufacturing system* is defined in this book as a collection of integrated equipment designed for some special mission, such as machining a defined part family or assembly of a certain product. Manufacturing systems also include people. Certain highly automated manufacturing systems can operate for extended periods of time without humans present to attend to their needs. But most manufacturing systems include workers as important elements of the system: for example, assembly workers on a conveyorized production line or part loaders/unloaders in a machining cell. Thus, manufacturing systems are designed with varying degrees of automation; some are highly automated, others are completely manual, and there is a wide range between.

The manufacturing systems in a factory are components of a larger system, which we refer to as a production system. We define a *production system* as the people, equipment, and procedures that are organized for the combination of materials and processes that comprise a company's manufacturing operations. Production systems are at level 4, the plant level, while manufacturing systems are at level 3 in our automation hierarchy. Production systems include not only the groups of machines and workstations in the factory but also the support procedures that make them work. These procedures include production control, inventory control, material requirements planning, shop floor control, and quality control. These systems are discussed in Parts IV and V. They are often implemented not only at the plant level but also at the corporate level (level 5).

REFERENCES

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

[2] GROOVER, M. P., "Automation," *Encyclopaedia Britannica, Macropaedia*, 15th Edition, Chicago, 1992. Vol. 14, pp. 548–557.

[3] GROOVER, M. P., "Automation," *Handbook of Design, Manufacturing, and Automation*, R. C. Dorf and A. Kusiak, Editors, John Wiley & Sons, Inc., New York, 1994, pp. 3–21.

[4] GROOVER, M. P., "Industrial Control Systems," *Maynard's Industrial Engineering Handbook*, Fifth Edition, Kjell Zandin, Editor, McGraw-Hill Book Company, New York, (forthcoming, 2001).

chapter 4

Industrial Control Systems

CHAPTER CONTENTS

The control system is one of the three basic components of an automation system (Section 3.1). 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 industrial control as it is applied in the processing industries and how it is applied in the discrete manufacturing industries.

4.1 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 (Section 2.1). 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 4.1.

4.1.1 Levels of Automation in the Two Industries

The levels of automation (Section 3.3) in the two industries are compared in Table 4.2. 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 above, 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.

TABLE 4.1 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
Heating	Mechanical assembly
Mixing and blending of ingredients	Plastic molding
Separation of ingredients	Sheet metal stamping

TABLE 4.2 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

4.1.2 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 3.1.2) 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 4.1.

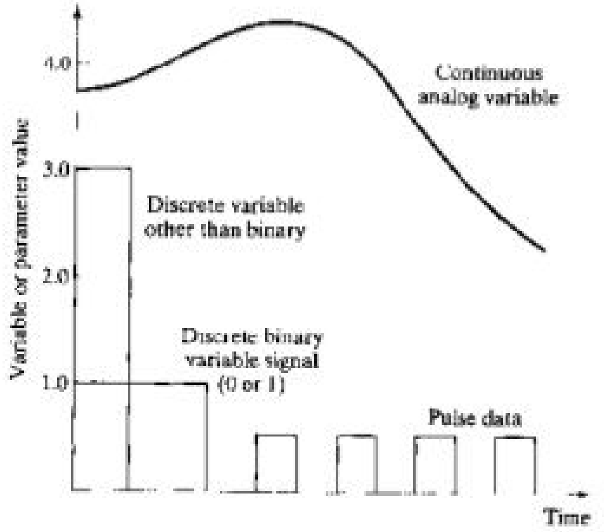


Figure 4.1 Continuous and discrete variables and parameters in manufacturing operations.

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 (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 variables 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 4.1. 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.

4.2 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 the controllers used here 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 4.3.

In reality, most operations in the process and discrete manufacturing industries tend to include both continuous as well as 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 5, 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, with the substitution of the digital computer to replace analog controllers in continuous process control applications starting around 1960 (Historical Note 4.1), 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 vari-

TABLE 4.3 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

ables 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.

4.2.1 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 3.1.3). 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 approaches by which the control objective is achieved 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

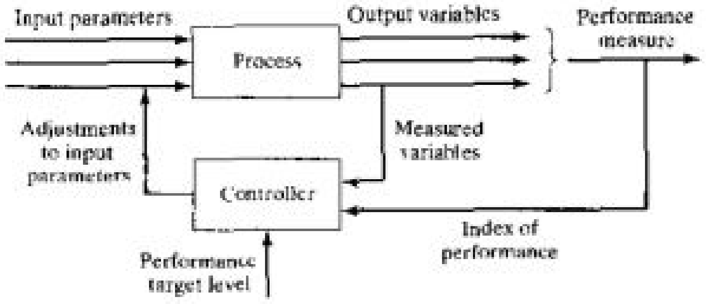


Figure 4.2 Regulatory control.

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 4.2.

The trouble with regulatory control (the same problem exists 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 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 4.3, 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.

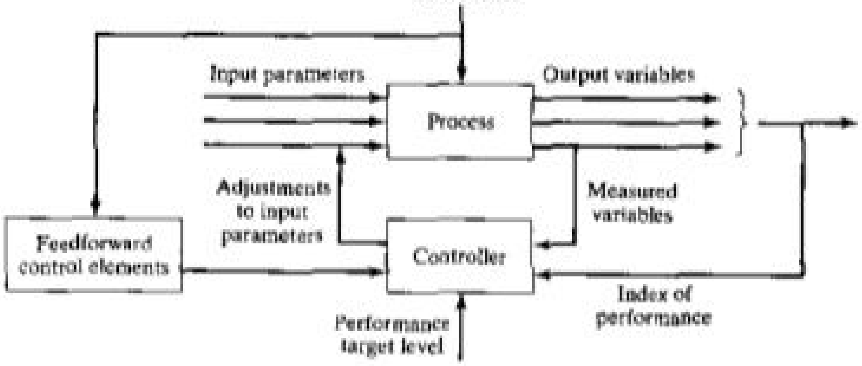


Figure 4.3 Feedforward control, combined with feedback control.

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 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 4.4. Several mathematical techniques are available for solving steady-state optimal control problems, including differential calculus, calculus of variations, and a variety of 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 day-to-day variations in 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 4.5. 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

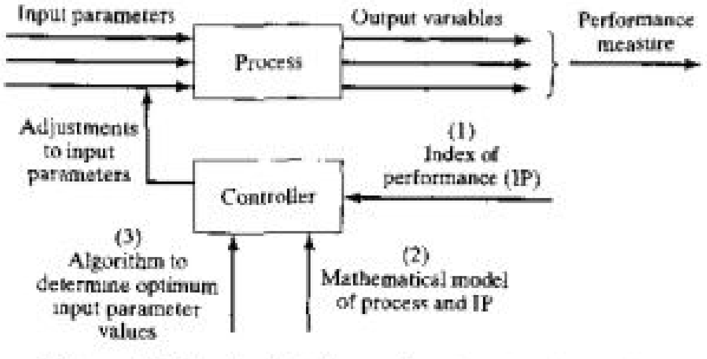


Figure 4.4 Steady-state (open-loop) optimal control.

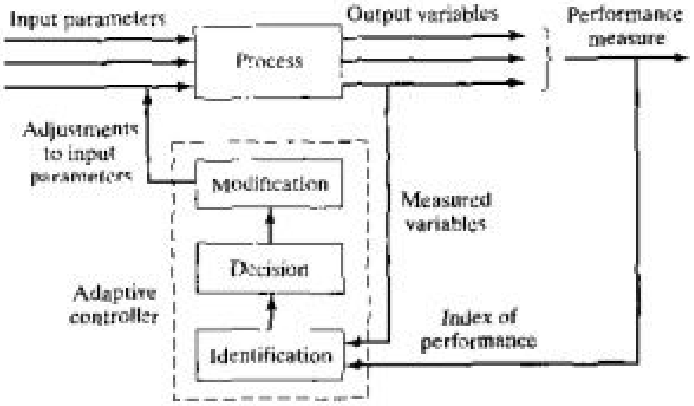


Figure 4.5 Configuration of an adaptive control system.

the environment changes over time, system performance also 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 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 4.2). Adaptive control has been the subject of research and development for several decades, 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 effort is adaptive control machining.

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 what effect these changes will have 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.

4.2.2 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 3.1.2). 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 [3].

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 triggers a valve to open that starts 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 has been 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 8.

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), stand-alone production machines (Chapter 14), flexible manufacturing systems (Chapter 16), automated transfer lines (Chapter 18), and automated assembly 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, which are well-suited to discrete control.

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.

4.3 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 4.1). Prior to then, 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 4.4.6.

Historical Note 4.1 Computer process control [2], [12].

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 chemicals, 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 Wooldridge (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, 3 pressures, and 3 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 4.3.2), 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 4.4.2), 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 from elimination of analog instrumentation for large systems, (2) simplified operator display panels, and (3) flexibility through 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 8.1). Also, *numerical control* (NC) machine tools (Historical Note 6.1) and industrial *robots* (Historical Note 7.1), 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 being 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.

Let us consider 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 observe the hierarchical structure of the functions performed by the control computer.

4.3.1 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 able to respond to the process within a short enough time period that process performance is not degraded. Factors that determine whether a computer controller can operate in real-time include: (1) the speed of the controller's central processing unit (CPU) and its interfaces, (2) the controller's operating system, (3) the design of the application software, and (4) the number of different input/output events to which the controller is designed to respond. 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:

3. *Computer commands to process.* In addition to incoming signals from the process, the control computer must be able to send control signals to the process to accom-

plish 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* is when some non-process-related action is called for in the program, 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) request for process data; and (5) emergency stop.

4.3.2 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 5. Other names used 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 when data are collected.
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 normally 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 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 its 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.

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

TABLE 4.4 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)

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 though it may be 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 4.4. Of course, the priority system may have more or less 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 4.1 illustrates the difference between the single-level and multilevel interrupt systems.

EXAMPLE 4.1 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. Shortly later, higher priority Task 2 arrives. And shortly later, 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 4.6.

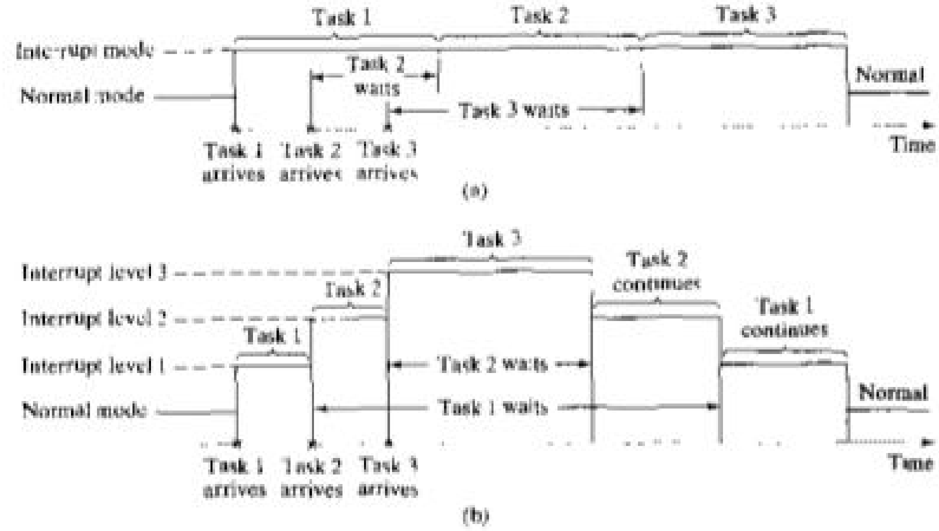


Figure 4.6 Response of the computer control system in Example 4.1 to three different priority interrupts for (a) a single-level interrupt system and (b) a multilevel interrupt system. Task 3 has the highest level priority. Task 1 has the lowest level. Tasks arrive for servicing in the order 1, then 2, 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.

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 3.2.3).

4.3.3 Levels of Industrial Process Control

In general, industrial control systems possess a hierarchical structure consisting of multiple levels of functions, similar to our levels of automation described in the previous chapter (Table 4.2). *ANSI/ISA-S88.01-1995*[1] [1] divides process control functions into three

---

[1] This standard [1] was prepared for batch process control but most of the concepts and terminology are applicable to discrete parts manufacturing and continuous process control.

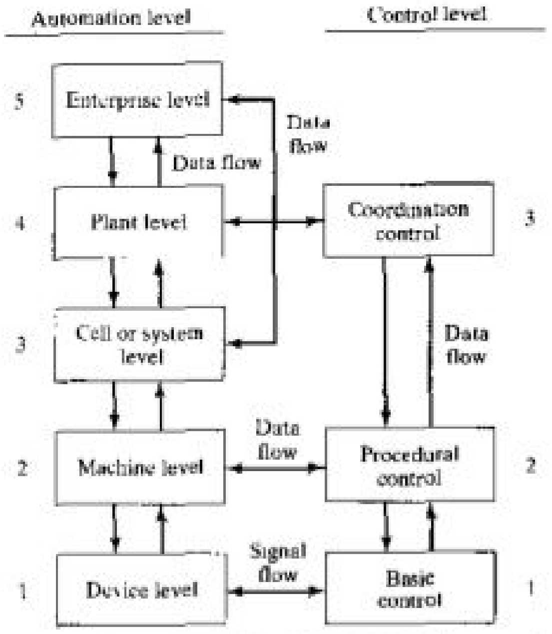


Figure 4.7 Mapping of *ANSI/ISA S88.01-1995* [1] control levels into the levels of automation in a factory.

levels: (1) basic control, (2) procedural control, and (3) coordination control. These control levels map into our automation hierarchy as shown in Figure 4.7. We now describe the three control levels, perhaps adapting the standard to fit our own models of continuous and discrete control (the reader is referred to the original standard [1], available from the Instrument Society of America).

Basic Control. This is the lowest level of control defined in the standard, corresponding to the device level in our automation hierarchy. In the process industries, this level is concerned with feedback control in the basic control loops. In the discrete manufacturing industries, basic control is concerned with directing the servomotors and other actuators of the production machines. Basic control includes functions such as feedback control, polling, interlocking, interrupts, and certain exception handling actions. Basic control functions may be activated, deactivated, or modified by either of the higher control levels (procedural or coordination control) or by operator commands.

Procedural Control. This intermediate level of control maps into regulatory control of unit operations in the process industries and into the machine level in discrete manufacturing automation (Table 4.2). In continuous control, procedural control functions include using data collected during polling to compute some process parameter value, changing setpoints and other process parameters in basic control, and changing controller gain constants. In discrete control, the functions are concerned with executing the work cycle program, that is, directing the machine to perform actions in an ordered sequence to accomplish some productive task. Procedural control may also involve executing error detection and recovery procedures and making decisions regarding safety hazards that occur during the process.

Coordination Control. This is the highest level in the control hierarchy in the ANSI/ISA standard. It corresponds to the supervisory level in the process industries and the cell or system level in discrete manufacturing. It is also likely to involve the plant and possibly the enterprise levels of automation. Coordination control initiates, directs, or alters the execution of programs at the procedural control level. Its actions and outcomes change over time, as in procedural control, but its control algorithms are not structured for a specific process-oriented task. It is more reactive and adaptive. Functions of coordination control at the cell level include: coordinating the actions of groups of equipment or machines, coordinating material handling activities between machines in a cell or system, allocating production orders to machines in the cell, and selecting among alternative work cycle programs.

At the plant and enterprise levels, coordination control is concerned with manufacturing support functions, including production planning and scheduling; coordinating common resources, such as equipment used in more than one production cell; and supervising availability, utilization, and capacity of equipment. These control functions are accomplished through the company's integrated computer and information system.

4.4 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 4.8. 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 require no 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.

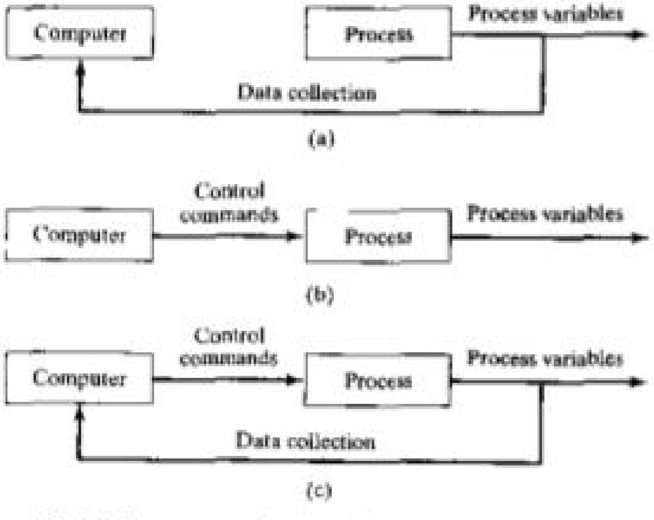


Figure 4.8 (a) Process monitoring, (b) open-loop process control, and (c) closed-loop process control.

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. The survey covers the following categories. (1) computer process monitoring, (2) direct digital control, (3) numerical control and robotics, (4) programmable logic controllers, (5) supervisory control, and (6) distributed control systems and personal computers. The second category, direct digital control, represents a transitory phase in the evolution of computer control technology. In its pure form, it is no longer used today. However, we briefly describe DDC to expose the opportunities it contributed. The sixth category, distributed control systems and personal computers, represents the most recent means of implementing computer process control.

4.4.1 Computer Process Monitoring

Computer process monitoring is one of the ways in which the computer can be interfaced with a process. *Computer process monitoring* 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.
2. *Equipment data.* These data indicate the status of the equipment in the work cell. Functions served by the data include monitoring machine utilization, scheduling tool changes, avoiding machine breakdowns, diagnosing equipment malfunctions, and planning 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 5 (Section 5.1). Bar codes and similar *automatic identification* 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*, explained in Chapter 26.

4.4.2 Direct Digital Control

Direct digital control 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. *Direct digital control* (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 4.9 and 4.10. 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 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, amplifier, and actuator that determines the input parameter to the process.

In the DDC system (Figure 4.10), 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

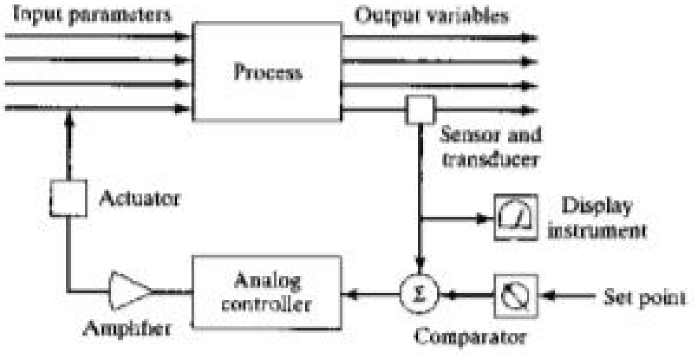


Figure 4.9 A typical analog control loop.

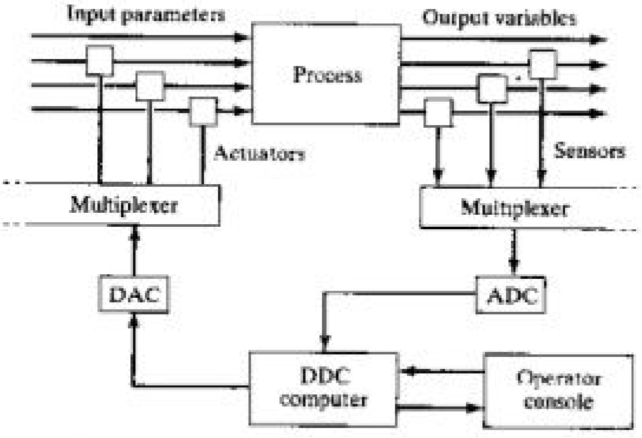


Figure 4.10 Components of a DDC system.

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.
- *Editing the control programs.* Using a digital computer makes it relatively easy to change the control algorithm if that becomes 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.

4.4.3 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 that specifies 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 must 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, we devote Chapter 6 to the topic of NC.

Closely related to NC is industrial robotics, in which the joints of the manipulator (robot arm) are controlled to move the end-of-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 7.

4.4.4 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-era 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 8.

4.4.5 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 the cell or system level. Thus, supervisory control coincides closely with coordination control in the ANSI/ISA-S88 Standard (Section 4.3.3). Supervisory control represents a higher level of control than the preceding forms of process control that we have surveyed in this section (i.e., 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 4.11.

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.

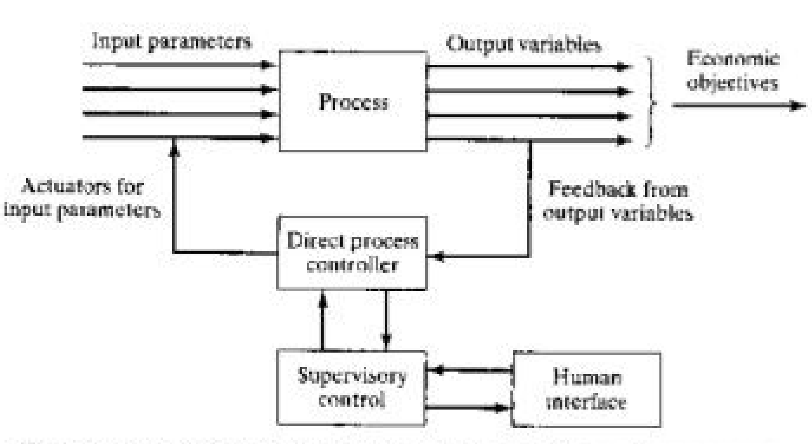


Figure 4.11 Supervisory control superimposed on other process-level control systems.

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: to minimize part or product costs by determining optimum operating conditions, to maximize machine utilization through efficient scheduling, to minimize tooling costs by tracking tool lives and scheduling tool changes, and similar supervisory goals. In NC, supervisory control takes the form of *direct numerical control* (Section 6.3), now more commonly referred to as *distributed numerical control*.

It is tempting to conceptualize a supervisory control system as being completely automated, that is, implemented so that the system operates with no human interference or assistance. But in virtually all cases, supervisory control systems are designed to allow for 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.

4.4.6 Distributed Control Systems and Personal Computers

Development of the microprocessor has had a significant impact on the design of control systems. In this section, we consider two related aspects of this impact: (1) distributed control systems and (2) the use of personal computers in control systems. Before discussing these topics, let us provide a brief background of the microprocessor and its uses.

Microprocessors. A *microprocessor* is an integrated circuit chip containing the digital logic elements needed to perform arithmetic calculations, execute instructions stored in memory, and carry out other data processing tasks. The digital logic elements and their interconnections in the circuit form a built-in set of instructions that determines the function of the microprocessor. A very common function is to serve as the central processing unit (CPU) of a microcomputer. By definition, a *microcomputer* is simply a small digital computer whose CPU is a microprocessor and which performs the basic functions of a computer. These basic functions consist of data manipulation and computation, carried out according to software stored in memory to accomplish user applications. The most familiar and widely used example of a microcomputer is the *personal computer* (PC), usually programmed with software for business and personal applications.

Microprocessors are also widely used as controllers in industrial control systems. An important distinction between a PC and a controller is that the controller must be capable of interacting with the process being controlled, as discussed in Section 4.3.1. It must be able to accept data from sensors connected to the process, and it must be able to send command signals to actuators attached to the process. These transactions are made possible by providing the controller with an extensive input/output (I/O) capability and by designing its microprocessor so that it can make use of this I/O capability. The number and type of I/O ports are important specifications of a microprocessor-based controller. By type of I/O ports, we are referring to whether the type of data and signals communicated between the controller and the process are continuous or discrete. We discuss I/O techniques in Chapter 5. In contrast, PCs are usually specified on the basis of memory size and execution speed, and the microprocessors used in them are designed with this in mind.

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 [13]:

- 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 is accomplished.
- *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.

These components are illustrated in a typical configuration of a distributed process control system presented in Figure 4.12. 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 [2]. 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

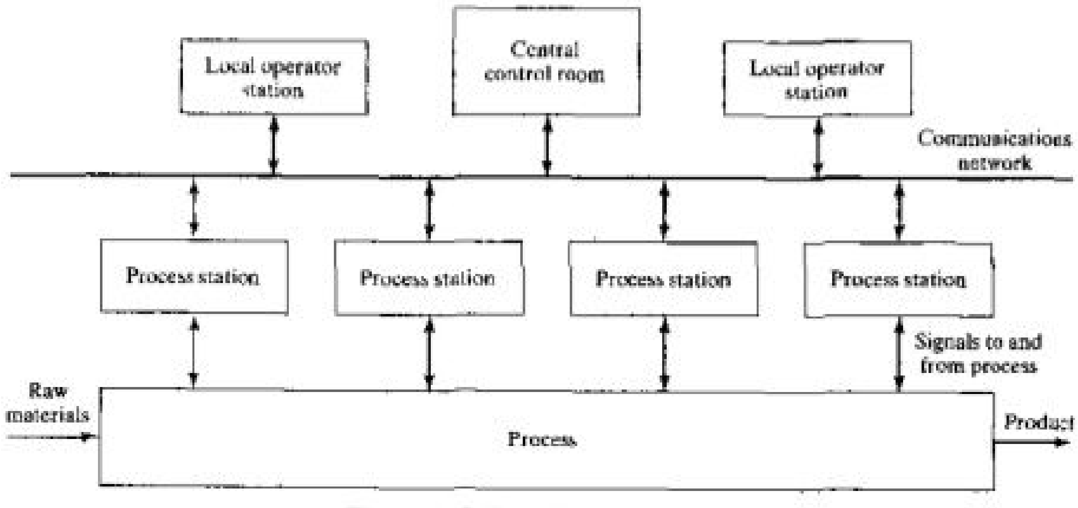


Figure 4.12 Distributed control system.

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 applications 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 DCSs.

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. Advantages of using a PC as only the operator interface include: (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.

Direct control 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 can be identified that have enabled this trend:

- widespread *familiarity with PCs*
- availability of *high-performance PCs*
- trend toward *open architecture* philosophy in control systems design
- Microsoft's *Windows NT*™ (the latest version is Windows 2000™) as the operating system of choice.

The PC is widely known to the general population in the United States and other industrialized nations. A large and growing number of individuals own them. Many others who do not personally own them use them at work. User-friendly software for the home and business has certainly contributed to the popularity of PCs. There is a growing expectation by workers that they be provided with a computer in their workplace, even if that workplace is in the factory.

High-performance CPUs are available in the latest PCs, and the next generation of PCs will be even more powerful. For the last 20 years, it has been observed that processor speed doubles every 12–18 months. This trend, called *Moore's Law*, is expected to continue for at least another 15 years. At the same time, processor costs have decreased by several orders of magnitude, and this trend is expected to continue as well. The projected results are seen in Table 4.5, in which performance is measured in millions of instructions per second (mips), and cost is measured in dollars per mips. In the early-to-mid 1990s, PC performance surpassed that of most digital signal processors and other components used in proprietary controllers [16]. New generations of PCs are currently being introduced more rapidly than PLCs are, allowing cycle speeds of PCs to exceed those of the latest PLCs.

Another important factor in the use of PCs for control applications is the availability of control products designed with an *open architecture* philosophy, 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 control 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 system, including the PCs used in the system.

For process control applications, the PC's operating system must facilitate real-time control and networking. At time of writing, Microsoft's *Windows NT*™ (now Windows 2000™) is being adopted increasingly as the operating system of choice for control and networking applications. Windows NT provides a multitasking environment with sufficient security, reliability, and fault tolerance for many if not most process control applications. At the same time, it provides the user friendliness of the desktop PC and most of the power of an engineering workstation. Installed in the factory, a PC equipped with Windows NT can perform multiple functions simultaneously, such as data logging, trend analysis, tool life monitoring, 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.

Not all control engineers agree that Windows NT can be used for critical process control tasks. For applications requiring microsecond response times, such as real-time motion control for machine tools, many control engineers are reluctant to rely on Windows NT. A common solution to this dilemma is to install a dedicated coprocessor in the PC. The motion servo loops are controlled in real time using the coprocessor motion control card, but the overall operating system is Windows NT.

TABLE 4.5 Trends in Processor Performance and Cost: Moore's Law

Year	*Mips**	*Cost per Mips ($)*
1978	125	9,600.00
1998	333	8.00
2011	100,000	.02

*Mips = millions of instructions per second.

Source: Studebaker [18].

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.

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 4.13. This is a trend that 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. Windows 2000™ provides 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 Excel spreadsheets). 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.

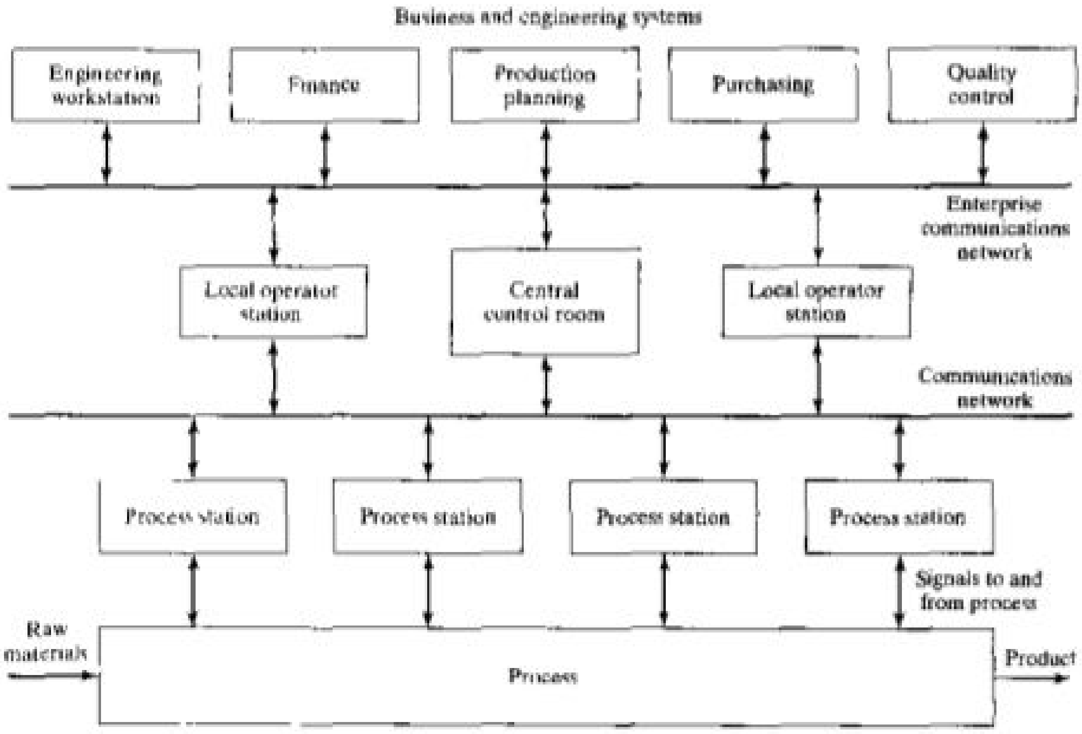


Figure 4.13 Enterprise-wide PC-based DCS.

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] American National Standard, *ANSI/ISA-S88.01, Batch Control, Part 1: Models and Terminology*, Instrument Society of America, Research Triangle Park, North Carolina, 1995.

[2] Astrom, K. J., and Wittenmark, B., *Computer-Controlled Systems-Theory and Design*, Third Edition, Prentice Hall, Upper Saddle River, New Jersey, 1997.

[3] Bateson, R. N., *Introduction to Control System Technology*, Sixth Edition, Prentice Hall, Upper Saddle River, New Jersey, 1999.

[4] Boucher, T. O., *Computer Automation in Manufacturing*, Chapman & Hall, London, 1996.

[5] Cawlfield, D., "PC-based direct control flattens control hierarchy, opens information flow," *Instrumentation & Control Systems*, September 1997, pp 61–67.

[6] Clark, D. C., "A case for Windows NT in distributed process control applications," *Instrumentation & Control Systems*, September 1997, pp 41–50.

[7] Cleveland, P., "Windows NT for real-time control: Which way to go?" *Instrumentation and Control Systems*, November 1997, pp. 63–68.

[8] Groover, M. P., "Industrial Control Systems," *Maynard's Industrial Engineering Handbook*, Fifth Edition, Kjell Zandin, Editor, McGraw-Hill Book Company, New York, (forthcoming, 2001).

[9] Hirsh, D., "Acquiring and sharing data seamlessly," *Instrumentation and Control Systems*, October 1997, pp. 25–35.

[10] Jones, C. T., and L. A. Bryan, *Programmable Controllers-Concepts and Applications*, International Programmable Controls, Inc., Atlanta, Georgia, 1983.

[11] *MicroMentor-Understanding and Applying Micro Programmable Controllers*, Allen-Bradley Company, Inc., 1995.

[12] Olsson, G., and G. Piani, *Computer Systems for Automation and Control*, Prentice Hall, London, 1992.

[13] Platt, G., *Process Control: A Primer for the Nonspecialist and the Newcomer*, Second Edition, Instrument Society of America, Research Triangle Park, North Carolina, 1998.

[14] Polischuk, T., "PCs are in control, but not everywhere," *Instrumentation and Control Systems*, March 1998, pp. 25–36.

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

[16] Sprow, E., "PC-based Controls: Part 1, The Opportunities," *Machine Shop Guide*, October 1998, pp. 28–36.

[17] Stoever, K., "21 st Century Manufacturing: Control Systems and the Internet," *Term Report*, MSE 427, Manufacturing Systems Engineering Program, Lehigh University, April 1998.

[18] Studebaker, P., "The Top 10 Trends in Process Control," *Control*, April 1998.