CBCS SCHEME

USN | | | | | | | | | |

18ME732

Seventh Semester B.E. Degree Examination, Jan./Feb. 2023
Automation and Robotics

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. What is automation? Explain basic elements of an automated system. **(10 Marks)**

b. Briefly explain advanced automation functions. **(10 Marks)**

OR

2 a. Explain with a neat sketch, feed forward control. **(06 Marks)**

b. What is ADC? Explain three phases in ADC. **(06 Marks)**

c. Discuss the input/output devices for discrete data. **(08 Marks)**

Module-2

3 a. What is an automated production line? Explain general configuration of an automated production line and its system configuration. **(10 Marks)**

b. Explain storage buffer in automated production line. **(04 Marks)**

c. A 20 station transfer line has an ideal cycle time $T_c = 1.2$ mins. The probability of station breakdown/cycle is equal for all stations and P = 0.05. Down time $T_d = 0.8$ mins. For each of the upper bound and lower bound, determine

(i) Frequency of line stops / cycle.

(ii) Average actual production rate.

(iii) Line efficiency. **(06 Marks)**

OR

4 a. Discuss the problem areas in analysis and design of automated production lines. **(08 Marks)**

b. Write short notes on the following:

(i) Bar code technology.

(ii) RFID technology. **(12 Marks)**

Module-3

5 a. Define a robot. Explain with neat sketches any two robot configurations. **(10 Marks)**

b. With suitable examples, explain industrial applications of robots. **(10 Marks)**

OR

6 a. Write a note on generations of robots. **(08 Marks)**

b. Write short notes on,

(i) End effectors.

(ii) Robot sensors.

(iii) Robot accuracy and repeatability **(12 Marks)**

Module-4

7 a. What are actuators? Explain with sketches, hydraulic and pneumatic actuators. **(10 Marks)**

b. With neat sketch, explain the working of,

(i) Velocity sensor.

(ii) Touch and tactile sensor. **(10 Marks)**

OR

8	a.	Write an explanatory note on actuator space and joint space.	**(08 Marks)**
	b.	Derive the direct kinematic equation for PUMA 560 robot.	**(12 Marks)**

Module-5

9	a.	Explain the levels of robot programming.	**(10 Marks)**
	b.	List and explain the requirements of robot programming language.	**(10 Marks)**

OR

10 Write short notes on :

a. Offline programming system.
b. Problems in robot programming languages.
c. Issues in OLP systems.
d. Sub tasks in OLP systems. **(20 Marks)**

* * * * *

1. a *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.

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.

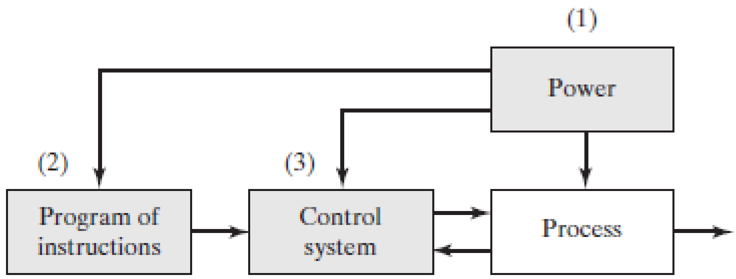


Figure 4.2 Elements of an automated system: (1) power, (2) program of instructions, and (3) control systems.

Fig.1 Basic elements of an automated system

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 non-automated processes.

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 4.1 are accomplished on discrete parts. These parts must be moved into the proper position 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 be removed. If the process is completely automated, then some form of mechanized power is used. If the process is manually operated or semi automated, 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.

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, perform the control calculations, and execute the instructions by transmitting the proper commands to 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* 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, for some processes, it is a legal requirement that records be kept of process performance and/or product quality. These data acquisition and record-keeping functions require power, although in modest amounts.

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, each part or product requires one or more processing steps that are unique to that part or product. These processing steps are performed during a work cycle. A new part is completed at the end of each work cycle (in some manufacturing operations, more than one part is produced during the work cycle: for example, 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, called *part programs* in numerical control.

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 is to perform some manufacturing operation.

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.

1.b

Advanced automation functions

Advanced automation functions include the following:

(1) Safety monitoring,

(2) Maintenance and repair diagnostics, and

(3) Error detection and recovery.

Safety Monitoring

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 include one or more of the following:

(1) Completely stopping the automated system,

(2) Sounding an alarm,

(3) Reducing the operating speed of the process, and

(4) Taking corrective actions to recover from the safety violation

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 work holding 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 in the work cell.
- ✓ Temperature sensors to indicate that a metal work part is hot enough to proceed with a hot forging operation. If the work part is not sufficiently heated, then the metal's ductility might 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 in the work cell.
- ✓ Machine vision systems to perform surveillance of the automated system and its surroundings.

Maintenance and Repair diagnostics

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 its cause can be identified.

3. *Recommendation of repair procedure.* In the third mode of operation, the subsystem recommends to the repair crew 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.

Error detection and recovery

Error Detection: The error detection step uses the automated system's available sensors to determine when a deviation or malfunction has occurred, interpret the sensor signal(s), and classify the error. Design of the error detection subsystem must begin with a systematic enumeration of all 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.

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 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 designed for the detected error, 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 it is detected. However, the designated corrective action must be possible to accomplish while the work cycle is still being executed. If that is not possible, then the process must be stopped.

3. ***Stop the process to invoke corrective action***. In this case, the deviation or malfunction requires that 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 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.

2.a **Feed forward Control.** The strategy in feed forward control is to anticipate the effect of disturbances that will upset the process by sensing them and compensating for them before they affect the process. As shown in Figure 5.3, the feed forward 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 delays and/or imperfections in the feedback measurements, actuator operations, and control algorithms, so feed forward control is usually combined with feedback control, as shown in the figure. Regulatory and feed forward control are more closely associated with the process industries than with discrete product manufacturing.

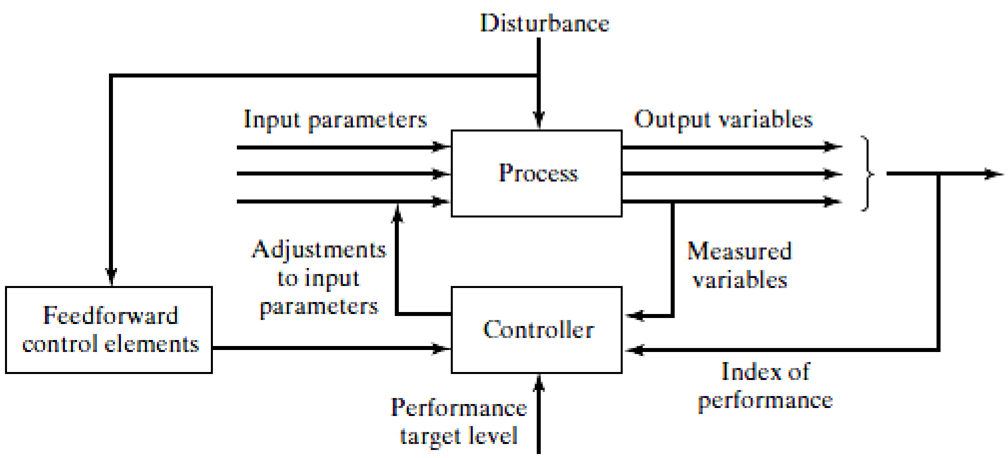


Figure 5.3 Feedforward control, combined with feedback control.

2.b. ***Analog-to-digital converter***. As its name indicates, the function of the ADC is to convert the incoming analog signal into its digital counterpart.

Consider the operation of the ADC, which is the heart of the conversion process. Analog-to-digital conversion occurs in three steps: (1) sampling, (2) quantization, and (3) encoding. Sampling consists of converting the continuous signal into a series of discrete analog signals at periodic intervals, as shown in Figure 6.10. In quantization, each discrete analog signal is assigned to one of a finite number of previously defined amplitude levels. The amplitude levels are discrete values of voltage ranging over the full scale of the ADC

In the encoding step, the discrete amplitude levels obtained during quantization are converted into digital code, representing the amplitude level as a sequence of binary digits. In selecting an analog-to-digital converter for a given application, the following factors are relevant: (1) sampling rate, (2) conversion time, (3) resolution, and (4) conversion method.

The sampling rate is the rate at which the continuous analog signals are sampled or polled. A higher sampling rate means that the continuous waveform of the analog signal can be more closely approximated. When the incoming signals are multiplexed, the maximum possible sampling rate for each signal is the maximum sampling rate of the ADC divided by the number of channels that are processed through the multiplexer. For example, if the maximum sampling rate of the ADC is 1,000 samples/sec, and there are 10 input channels through the multiplexer, then the maximum sampling rate for each input line is 1,000>10 = 100 sample>sec. (This ignores time losses due to multiplexer switching.)

The maximum possible sampling rate of an ADC is limited by the ADC conversion time. Conversion time of an ADC is the time interval between the application of an incoming signal and the determination of the digital value by the quantization and encoding steps of the conversion procedure. Conversion time depends on (1) the type of conversion procedure used by the ADC and (2) the number of bits *n* used to define the converted digital value.

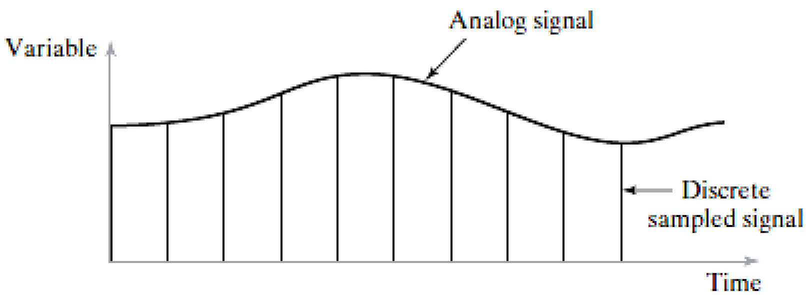


Figure 6.10 Analog signal converted into series of discrete sampled data by analog-to-digital converter.

2.c. Input/Output Devices for Discrete Data

Discrete data can be processed by a digital computer without the kinds of conversion procedures required for continuous analog signals. Discrete data divide into three categories: (a) binary data, (b) discrete data other than binary, and (c) pulse data.

Contact Input/Output Interfaces

Contact interfaces are of two types, input and output. These interfaces read binary data from the process into the computer and send binary signals from the computer to the process, respectively. The terms *input* and *output* are relative to the computer.

A ***contact input interface*** is a device by which binary data are read into the computer from some external source (e.g., a process). It consists of a series of simple contacts that can be either closed or open (on or off) to indicate the status of binary devices connected to the process such as limit switches (contact or no contact), valves (open or closed), or motor pushbuttons (on or off). The computer periodically scans the actual status of the contacts to update the values stored in memory.

The ***contact output interface*** is a device that communicates on/off signals from the computer to the process. The contact positions are set either on or off. These positions are maintained until changed by the computer, perhaps in response to events in the process. In computer process-control applications, hardware controlled by the contact output interface include alarms, indicator lights (on control panels), solenoids, and constant-speed motors. The computer controls the sequence of on/off activities in a work cycle through this contact output interface.

Pulse Counters and Generators

A ***pulse counter*** is a device that converts a series of pulses into a digital value. The value is then entered into the computer through its input channel. The most common type of pulse counter is one that counts electrical pulses. It is constructed using sequential logic gates, called *flip-flops*, which are electronic devices that possess memory capability and that can be used to store the results of the counting procedure.

Pulse counters can be used for both counting and measurement applications. A typical counting application might add up the number of packages moving past a photoelectric sensor along a conveyor in a distribution center. A typical measurement application might indicate the rotational speed of a shaft. One possible method to accomplish the measurement is to connect the shaft to a rotary encoder which generates a certain number of electrical pulses for each rotation. To determine rotational speed, the pulse counter measures the number of pulses received during a certain time period and divides this by the duration of the time period and by the number of pulses in each revolution of the encoder.

A ***pulse generator*** is a device that produces a series of electrical pulses whose total number and frequency are determined and sent by the control computer. The total number of pulses might be used to drive a stepper motor in a positioning system. The frequency of the pulse train, or pulse rate, could be used to control the rotational speed of a stepper motor. A pulse generator operates by repeatedly closing and opening an electrical contact, thus producing a sequence of discrete electrical pulses. The amplitude (voltage level) and frequency are designed to be compatible with the device being controlled.

3.a. **System Configurations**

Although Figure 16.1 shows the flow of work to be in a straight line, the work flow can actually take several different forms: (1) in-line, (2) segmented in-line, and (3) rotary. The in-line configuration consists of a sequence of stations in a straight line arrangement, as in Figure 16.1. This configuration is common for machining big work pieces, such as automotive engine blocks, engine heads, and transmission cases. Because these parts require a large number of operations, a production line with many stations is needed. The in-line configuration can accommodate a large number of stations. In-line systems can also be designed with integrated storage buffers along the flow path

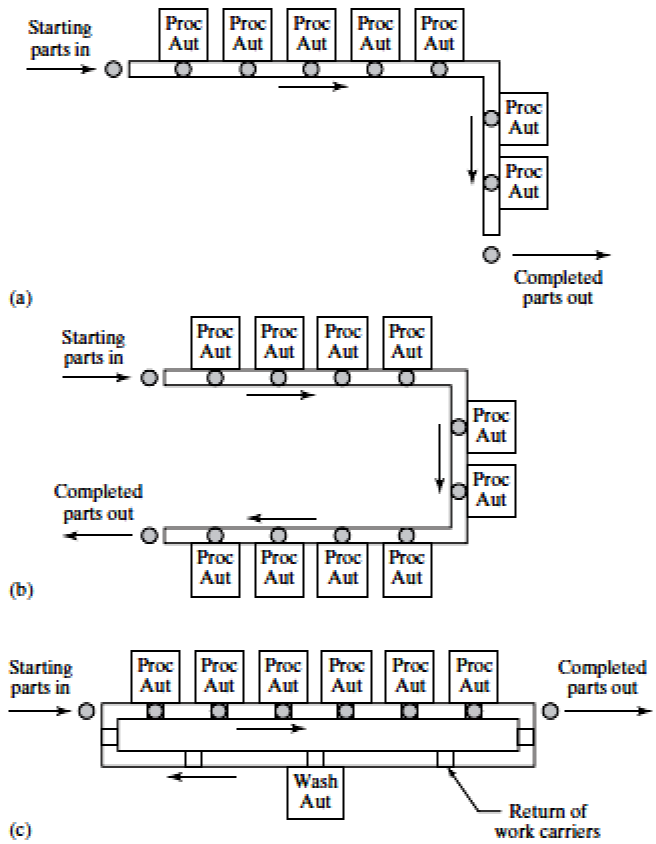


Figure 16.2 Several possible layouts of the segmented in-line configuration of an automated production line: (a) L-shaped, (b) U-shaped, and (c) rectangular. Key: Proc = processing operation, Aut = automated workstation, Wash = work carrier washing station.

The segmented in-line configuration consists of two or more straight-line transfer sections, where the segments are usually perpendicular to each other. Figure 16.2 shows several possible layouts of the segmented in-line category. There are a number of reasons for designing a production line in these configurations rather than in a pure straight line: (1) available floor space may limit the length of the line, (2) a workpiece in a segmented in-line configuration can be reoriented to present different surfaces for machining, and (3) the rectangular layout provides for swift return of work-holding fixtures to the front of the line for reuse.

Figure 16.3 shows two transfer lines that perform metal machining operations on automotive castings. The first line, on the left-hand side, is a segmented in-line configuration in the shape of a rectangle. Pallet fixtures are used in this line to position the starting castings at the workstations for machining. It is a palletized transfer line. The second line, on the right side, is a conventional in-line configuration. When processing on the first line is completed, the parts are manually transferred to the second line, where they are reoriented to present different surfaces for machining. In this line the parts are moved individually by the transfer mechanism, using no pallet fixtures. It is a free transfer line.

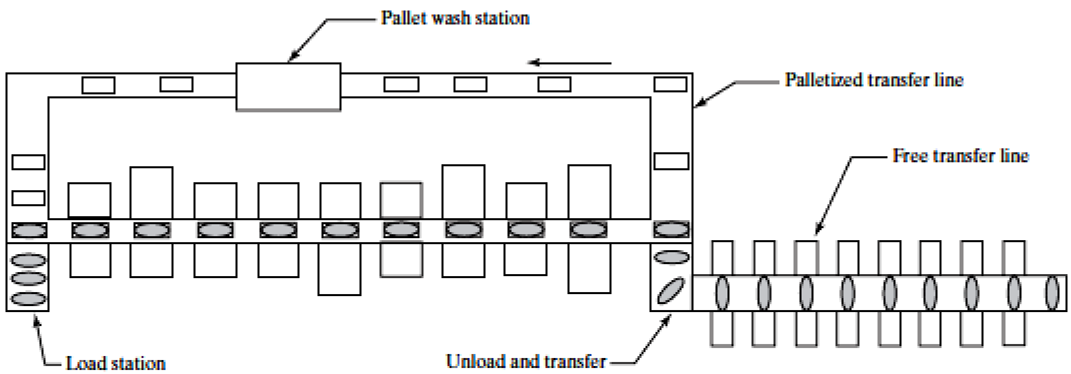


Figure 16.3 Two machining transfer lines. On the left is a segmented in-line configuration that uses pallet fixtures to locate the work parts. The return loop brings the pallets back to the front of the line. On the right, the second transfer line is an in-line configuration. The manual station between the lines is used to reorient the parts, represented as ovals. Pallet fixtures are represented as rectangles.

In the rotary configuration, the work parts are attached to fixtures around the periphery of a circular worktable, and the table is indexed (rotated in fixed angular amounts) to present the parts to workstations for processing. A typical arrangement is illustrated in Figure 16.4. The worktable is often referred to as a dial, and the equipment is called a *dial-indexing machine*.

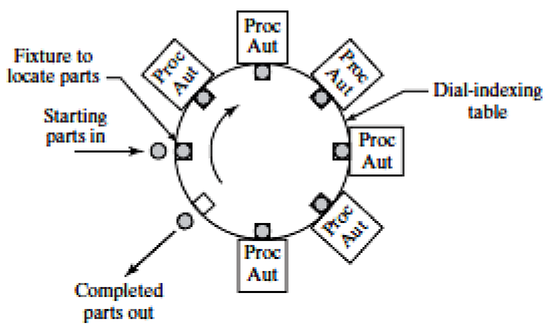


Figure 16.4 Rotary indexing machine (dial-indexing machine).
Key: Proc = processing operation, Aut = automated workstation.

3.b Storage Buffers

Automated production lines can be designed with storage buffers. A storage buffer is a location in the production line where parts can be collected and temporarily stored before proceeding to downstream workstations. The storage buffers can be manually operated or automated. When it is automated, a storage buffer consists of a mechanism to accept parts from the upstream workstation, a place to store the parts, and a mechanism to supply parts to the downstream station. A key parameter of a storage buffer is its storage capacity, that is, the number of work parts it can hold. Storage buffers may be located between every pair of adjacent stations, or between line stages containing multiple stations.

There are several reasons why storage buffers are used on automated production lines:

- To reduce the impact of station breakdowns. Storage buffers between stages on a production line permit one stage to continue operation while the other stage is down for repairs.

- To provide a bank of parts to supply the line. Parts can be collected into a storage unit and automatically fed to a downstream manufacturing system. This permits untended operation of the system between refills.
- To provide a place to put the output of the line.
- To allow for curing time or other process delay. A curing time is required for some processes such as painting or adhesive application. The storage buffer is designed to provide sufficient time for curing to occur before supplying the parts to the downstream station.
- To smooth cycle time variations. Although this is generally not an issue in an automated line, it is relevant in manual production lines, where cycle time variations are an inherent feature of human performance.

2.c

A 20 station transfer line has an ideal cycle time of Tc = 1.2 mins. The probability of station breakdown per cycle is equal for all stations & P = 0.005 breakdowns / cycle. For each of the upper bound & lower bound determine:

a) frequency of line stops per cycle

b) average actual production rate

c) line efficiency

a) For the Upper bound approach

F = 20 (0.005) = 0.10 lines per cycle

20 20

F = 1 - (1 – 0.005) = 1 – (0.995)

= 1 – 0.0946

= 0.0954 line stops per cycle

For the Upper bound approach the production rate,

Rp = 1

20

= 0.500 pc / min

= 30 pc /hr

For the lower bound approach the production time we calculate by using the formula for F

Tp = Tc + F (Td)

= 1.2 + 0.0954 (0.8)

= 1.9631 mins

Production rate = 0.9046

1.9631

= 0.4608 pc / min

= 27.65 pc / hr

The production rate is about 8% lower than that we computed by the upper bound approach.

We should note that:

Rp = 1

0.9631

= 0.5094 cycles / min

= 30.56 cycles / hr

which is slightly higher than in the upper bound case.

c) For the upper bound the line efficiency will be
E = 1.2
2.0
= 0.6
= 60 %
For the lower bound approach we have
E = 1.2
1.9631
= 0.6113
= 61.13 %
Line efficiency is greater with lower bound approach even though production rate is lower.
This is because lower bound approach leaves fewer parts remaining on the line to jam.

4.a **Analysis of Transfer Lines**

In the analysis and design of automated production lines, three problem areas must be considered: (1) line balancing, (2) processing technology, and (3) system reliability.

The line balancing problem is most closely associated with manual assembly lines but it is also an issue on automated production lines. Somehow, the total work content to be accomplished on the automated line must be divided as evenly as possible among the workstations. In a manual assembly line, the total work content can be divided into much smaller work elements, and the elements can then be grouped and assigned to workstations to determine the task that is performed at each station. Each task has a corresponding service time. In an automated production line, the tasks consist of processing steps whose sequence and service times are limited by technological considerations.

For example, in a machining transfer line, certain operations must be performed before others. Drilling must precede tapping to create a threaded hole. Locating surfaces must be machined before the features that will use those locating surfaces are machined.

Process technology refers to the body of knowledge about the particular manufacturing processes used on the production line. For example, in the machining process, process technology includes the metallurgy and machinability of the work material, the proper application of cutting tools, selection of speeds and feeds, chip control, and a host of other problem areas and issues. Many of the problems encountered in machining can be solved by application of machining theory and principles. The same is true of other processes. In each process, a technology has been developed over many years of research and practice. By applying this technology, each individual workstation in the production line can be designed to operate at or near its maximum performance.

The third problem area in the analysis and design of automated production lines is reliability. In a highly complex and integrated system such as an automated production line, failure of any one component can stop the entire system.

Cycle Time Analysis. In the operation of an automated production line, parts are introduced into the first workstation and are processed and transported at regular intervals to succeeding stations.

This interval defines the ideal cycle time *Tc* of the production line. *Tc* is the processing time for the slowest station on the line plus the transfer time, that is,

$$T_c = \text{Max}\{T_{si}\} + T_r$$

where *Tc* = ideal cycle time on the line, min; *Tsi* = the processing time at station *i*, min; and *Tr* = repositioning time, called the transfer time here, min. The Max{*Tsi*} is used in Equation because this longest service time establishes the pace of the production line. The remaining stations with shorter service times must wait for the slowest station. Therefore, these other stations will experience idle time.

In the operation of a transfer line, random breakdowns and planned stoppages cause downtime on the line.

Although the breakdowns and line stoppages occur randomly, their frequency can be measured over the long run. When the line stops, it is down a certain amount of time for each downtime occurrence. Downtime occurrences cause the actual average production cycle time of the line to be longer than the ideal cycle time given by above Equation. The actual average production time T_p can be formulated as follows:

$$T_p = T_c + FT_d$$

where *F* = downtime frequency, line stops/cycle; and T_d = average downtime per line stop, min. The downtime *Td* includes the time for the repair crew to swing into action, diagnose the cause of the failure, fix it, and restart the line. Thus, FT_d = downtime averaged on a per cycle basis.

4.b BAR CODE TECHNOLOGY

Bar codes divide into two basic types: (1) linear, in which the encoded data are read using a linear sweep of the scanner, and (2) two-dimensional, in which the encoded data must be read in both directions.

3 22170 22024 9

(a) (b)

Figure 12.1 Two forms of linear bar codes are (a) width-modulated, exemplified here by the Universal Product Code, and (b) height-modulated, exemplified here by Postnet, used by the U.S. Postal Service.

Linear (One-Dimensional) Bar Codes

Linear bar codes are the most widely used automatic identification and data capture technique. There are actually two forms of linear bar code symbologies, illustrated in Figure 12.1: (a) width-modulated, in which the symbol consists of bars and spaces of varying width; and (b) height-modulated, in which the symbol consists of evenly spaced bars of varying height. The only significant application of the height-modulated bar code symbologies is in the U.S. Postal Service for ZIP code identification, so the discussion here focuses on the width-modulated bar codes, which are used widely in retailing and manufacturing.

In linear width-modulated bar code technology, the symbol consists of a sequence of wide and narrow colored bars separated by wide and narrow spaces (the colored bars are usually black and the spaces are white for high contrast). The pattern of bars and spaces is coded to represent numeric or alphanumeric character.

Bar code readers interpret the code by scanning and decoding the sequence of bars. The reader consists of the scanner and decoder. The scanner emits a beam of light that is swept past the bar code (either manually or automatically) and senses light reflections to distinguish between the bars and spaces. The light reflections are sensed by a photo detector, which converts the spaces into an electrical signal and the bars into absence of an electrical signal. The width of the bars and spaces is indicated by the duration of the corresponding signals.

The Bar Code Symbol. The bar code standard adopted by the automotive industry, the Department of Defense, the General Services Administration, and many other manufacturing industries is Code 39, also known as AIM USD–2 (Automatic Identification Manufacturers Uniform Symbol Description-2). Code 39 uses a series of wide and narrow elements (bars and spaces) to represent alphanumeric and other characters. The wide elements are equivalent to a binary value of one and the narrow elements are equal to zero. The width of the wide bars and spaces is between two and three times the width of the narrow bars and spaces. Whatever the wide-to-narrow ratio, the width must be uniform throughout the code for the reader to be able to consistently interpret the resulting pulse train. Figure 12.4 presents the character structure for USD–2.

The reason for the name Code 39 is that nine elements (bars and spaces) are used in each character and three of the elements are wide. The placement of the wide spaces and bars in the code uniquely designates the character. Each code begins and ends with either a wide or narrow bar. The code is sometimes referred to as code three-of-nine. In addition to the character set in the bar code, there must also be a so-called “quiet-zone” both preceding and following the bar code, in which there is no printing that might confuse the decoder.

Bar Code Readers. Bar code readers come in a variety of configurations; some require a human to operate them and others are stand-alone automatic units. They are usually classified as contact or noncontact readers.

Contact bar code readers are handheld wands or light pens operated by moving the tip of the wand quickly past the bar code on the object or document. The wand tip must be in contact with the bar code surface or in very close proximity during the reading procedure. In a factory data collection application, they are usually part of a keyboard entry terminal. The terminal is sometimes referred to as a stationary terminal in the sense that it is placed in a fixed location in the shop. When a transaction is entered in the factory, the data are usually communicated to the computer system immediately. In addition to their use in factory data collection systems, stationary contact bar code readers are widely used in retail stores to enter the item in a sales transaction.

Noncontact bar code readers focus a light beam on the bar code, and a photo-detector reads the reflected signal to interpret the code. The reader probe is located a certain distance from the bar code (several inches to several feet) during the read procedure. Noncontact readers are classified as fixed beam and moving beam scanners. Fixed beam readers are stationary units that use a fixed beam of light. They are usually mounted beside a conveyor and depend on the movement of the bar code past the light beam for their operation. Applications of fixed beam bar code readers are typically in warehousing and material handling operations where large quantities of materials must be identified as they flow past the scanner on conveyors.

Char.	Bar pattern	9 bits	Char.	Bar pattern	9 bits
1		100100001	K		100000011
2		001100001	L		001000011
3		101100000	M		101000010
4		000110001	N		000010011
5		100110000	O		100010010
6		001110000	P		001010010
7		000100101	Q		000000111
8		100100100	R		100000110
9		001100100	S		001000110
0		000110100	T		000010110
A		100001001	U		110000001
B		001001001	V		011000001
C		101001000	W		111000000
D		000011001	X		010010001
E		100011000	Y		110010000
F		001011000	Z		011010000
G		000001101	-		010000101
H		100001100	.		110000100
I		001001100	space		011000100
J		000011100	*		010010100

*Denotes a start/stop code that must be placed at the beginning and end of every bar code message.

Moving beam scanners use a highly focused beam of light, often a laser, actuated by a rotating mirror to traverse an angular sweep in search of the bar code on the object. A scan is defined as a single sweep of the light beam through the angular path. The high rotational speed of the mirror allows for very high scan rates—up to 1,440 scans/sec. This means that many scans of a single bar code can be made during a typical reading procedure, thus permitting verification of the reading. Moving beam scanners can be either stationary or portable units. Stationary scanners are located in a fixed position to read bar codes on objects as they move past on a conveyor or other material handling equipment. They are used in warehouses and distribution centers to automate the product identification and sortation operations. A typical setup using a stationary scanner is illustrated in Figure 12.5. Portable scanners are handheld devices that the user points at the bar code like a pistol. The vast majority of bar code scanners used in factories and warehouses are of this type.

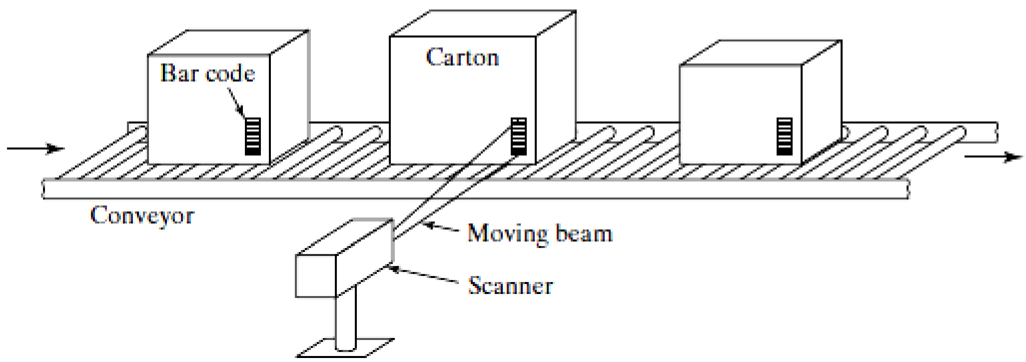


Figure 12.5 Stationary moving beam bar code scanner located along a moving conveyor.

Bar Code Printers. In many bar code applications, the labels are printed in medium-to-large quantities for product packages and the cartons used to ship the packaged products. These preprinted bar codes are usually produced off-site by companies specializing in these operations. The labels are printed in either identical or sequenced symbols. Printing technologies include traditional techniques such as letterpress, offset lithography, and flexographic printing. Bar codes can also be printed on-site by methods in which the process is controlled by microprocessor to achieve individualized printing of the bar coded document or item label. These applications tend to require multiple printers distributed at locations where they are needed. The printing technologies used in these applications include ink-jet, laser printing, and laser etching.

Two-Dimensional Bar Codes

Two-dimensional symbologies divide into two basic types: (1) stacked bar codes and (2) matrix symbologies.

Stacked Bar Codes. The first 2-D bar code to be introduced was a stacked symbology. It was developed in an effort to reduce the area required for a conventional bar code. But its real advantage is that it can contain significantly greater amounts of data. A stacked bar code consists of multiple rows of conventional linear bar codes stacked on top of each other. Several stacking schemes have been devised over the years, nearly all of which allow for multiple rows and variations in the numbers of encoded characters possible.

Matrix Symbologies. A matrix symbology consists of 2-D patterns of data cells that are usually square and are colored dark (usually black) or white. The 2-D matrix symbologies were introduced around 1990. Their advantage over stacked bar codes is their capability to contain more data. They also have the potential for higher data densities—up to 30 times more dense than Code 39. Their disadvantage compared to stacked bar codes is that they are more complicated, which requires more sophisticated printing and reading equipment.

Figure 12.6 A 2-D stacked bar code. Shown is an example of a PDF417 symbol.

Figure 12.7 A 2-D matrix bar code. Shown is an example of the Data Matrix symbol.

RADIO FREQUENCY IDENTIFICATION

Radio frequency identification technology (RFID) represents the biggest challenge to the dominance of bar codes. Companies including Walmart, Target, and Metro AG (in Germany), as well as the U.S. Department of Defense, have mandated that their suppliers use RFID on incoming materials. In fact the Department of Defense requires a combination of Data Matrix and RFID on all "mission-critical" parts, assemblies, and equipment. These requirements have provided a significant impetus for the implementation of RFID in industry. According to a study of Walmart cited in [17], “RFID stores are 63 percent more effective in replenishing out-of-stock items than traditional stores.”

In radio frequency identification, an identification tag or label containing electronically encoded data is attached to the subject item, which can be a part, product, or container (e.g., carton, tote pan, pallet). The identification tag consists of an integrated circuit chip and a small antenna, as pictured in Figure 12.8. These components are usually enclosed in a protective plastic container or are imbedded in an adhesive-backed label that is attached to item. The tag is designed to satisfy the Electronic Product Code (EPC) standard, which is the RFID counterpart to the Universal Product Code (UPC) used in bar codes. The tag communicates the encoded data by RF to a reader or interrogator as the item is brought into the reader’s proximity. The reader can be portable or stationary. It decodes and confirms the RF signal before transmitting the associated data to a collection computer.

Although the RF signals are similar to those used in wireless radio and television transmission, there are differences in how RF technology is used in product identification. One difference is that the communication is in two directions rather than in one direction as in commercial radio and TV. The identification tag is a ***transponder***, a device that emits a signal of its own when it receives a signal from an external source. To activate it, the reader transmits a low-level RF magnetic field that serves as the power source for the transponder when they are near each other. Another difference between RFID and commercial radio and TV is that the signal power is substantially lower in RFID applications (milliwatts to several watts), and the communication distances usually range between several millimeters and several meters. Finally, there are differences in the allowable frequencies that can be used for RFID applications versus radio, TV, and other commercial and military users.

RF identification tags are available in two general types: (1) passive and (2) active. Passive tags have no internal power source; they derive their electrical power for transmitting a signal from radio waves generated by the reader when in close proximity. Active tags include their own battery power packs. Passive tags are smaller, less expensive, longer lasting, and have a shorter radio communication range. Active tags generally possess a larger memory capacity and a longer communication range (typically 10 m and more). Applications of active tags tend to be associated with higher value items due to the higher cost per tag.

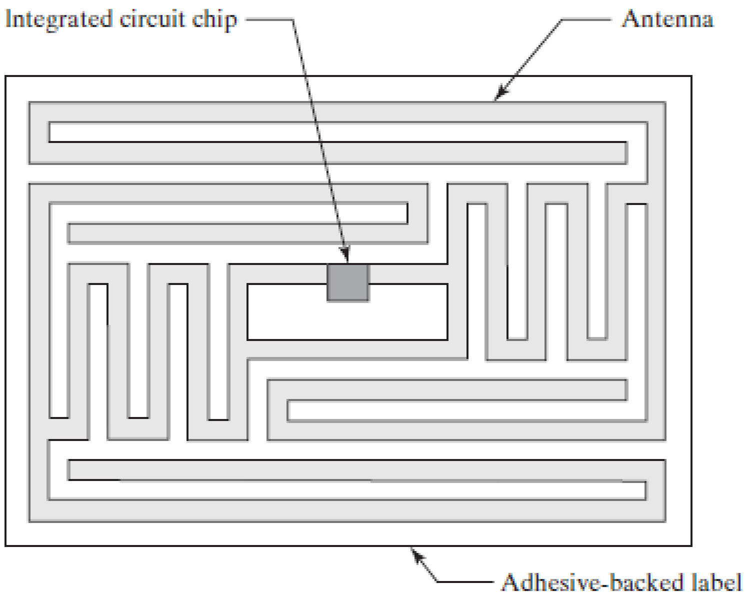


Figure 12.8 RFID label. Approximate size is 20 mm by 30 mm (0.8 in by 1.2 in).

5.a. A robot is a reprogrammable multi functional manipulator designed to move material, parts, tools or specialized devices through variable programmed motions for the performance of variety of tasks.

1. Polar configuration/Spherical configuration

Notation: [**LTR**]: Linear, Twisting and Rotational joint

This configuration also called as Polar coordinate configuration. It goes by the name "spherical coordinate" also because the workspace within which it can move its arm is a partial sphere as shown in figure. The robot has a rotary base and a pivot that can be used to raise and lower a telescoping arm.

i) Operate within a **spherical** work volume
ii) Has 1 prismatic and 2 revolute axes.
iii) First motion is a base rotation, Second motion correspond to an elbow rotation and Third motion is radial or in-out motion
iv) Elbow rotation and arm reach limit the design of full spherical motion.
v) Rarely used in industries but common in automated cranes.

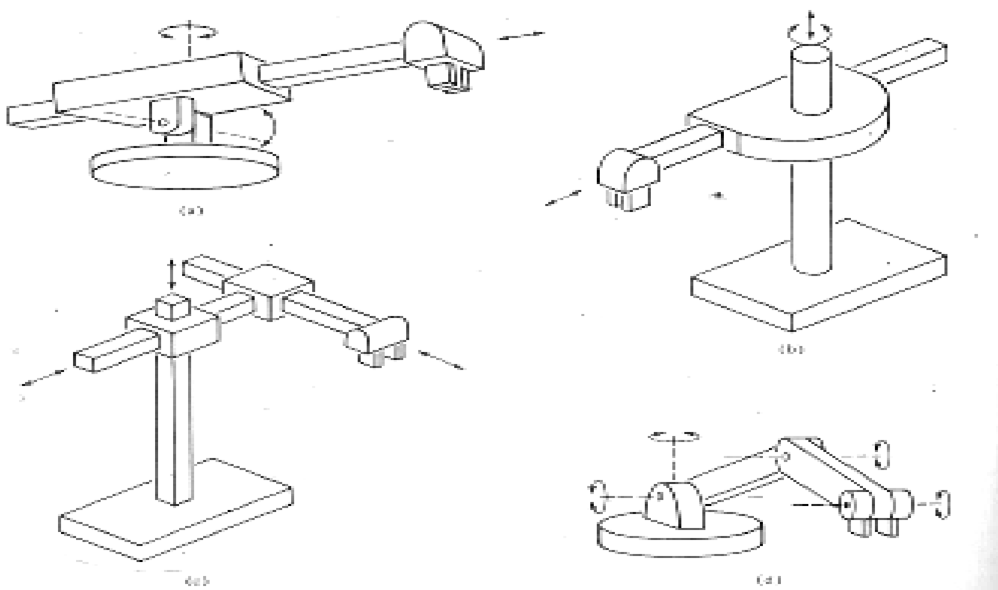


Figure 2-1 The four basic robot anatomies: (a) polar, (b) cylindrical, (c) cartesian, and (d) jointed-arm. (Reprinted from Reference [7].)

2. Cylindrical Configuration

Notation: [**TLL**]: Twisting, Linear and Linear.

This also has 3 degrees of freedom, 2 prismatic and 1 revolute joints. It moves linearly along X and Y axes and rotaion about at its base i.e. Z- axis. The robot body is a vertical column that swivels about a vertical axis. The arm consists of several orthogonal slides which allow the arm to be moved up or down and in and out with respect to the body. This is illustrated schematically in figure.

Features:

i) Operate within a **cylindrical** work volume
ii) 2 prismatic and 1 revolute joints.
iii) Position is specified by Y value (height) extension of arm X axis and angle of rotation of Z axis (θ)
iv) Recommended for pick and place operation such as machine loading and unloading.
v) Lower repeatability and accuracy
vi) Require more sophisticated control
vii) Rigid structure & high lift-carrying capacity

5.b **Applications of Industrial Robots**

Robots are used in a wide field of applications in industry. Most of the current applications are in manufacturing. The applications can usually be classified into one of the following categories: (1) material handling, (2) processing operations, and (3) assembly and inspection.

Material Handling Applications

In material handling applications, the robot moves materials or parts from one place to another. To accomplish the transfer, the robot is equipped with a gripper that must be designed to handle the specific part or parts to be moved. Included within this application category are (1) material transfer and (2) machine loading and/or unloading. In many material handling applications, the parts must be presented to the robot in a known position and orientation. This requires some form of material handling device to deliver the parts **Material Transfer.** A more complex example of material transfer is ***palletizing***, in which the robot retrieves parts, cartons, or other objects from one location and deposits them onto a pallet or other container at multiple positions on the pallet. Other applications similar to palletizing include ***depalletizing***, which consists of removing parts from an ordered arrangement in a pallet and placing them at another location (e.g., onto a moving conveyor); ***stacking*** operations, which involve placing flat parts on top of each other, such that the vertical location of the drop-off position is continuously changing with each cycle; and ***insertion*** operations, in which the robot inserts parts into the compartments of a divided carton.

Machine Loading and/or Unloading. In machine loading and/or unloading applications, the robot transfers parts into and/or from a production machine. The three possible cases are (1) machine loading, in which the robot loads parts into the production machine, but the parts are unloaded from the machine by some other means; (2) machine unloading, in which the raw materials are fed into the machine without using the robot, and for a robot palletizing operation into the work cell in this position and orientation. the robot unloads the finished parts; and (3) machine loading and unloading, which involves both loading of the raw work part and unloading of the finished part by the robot. Industrial robot applications of machine loading and/or unloading include the following processes:

• *Die casting.* The robot unloads parts from the die casting machine. Peripheral operations sometimes performed by the robot include dipping the parts into a water bath for cooling.

• *Plastic molding.* Plastic molding is similar to die casting. The robot unloads molded parts from the injection molding machine.

• *Metal machining operations.* The robot loads raw blanks into the machine tool and unloads finished parts from the machine. The change in shape and size of the part before and after machining often presents a problem in end effector design, and dual grippers (Section 8.3.1) are often used to deal with this issue.

• *Forging.* The robot typically loads the raw hot billet into the die, holds it during the forging strikes, and removes it from the forge hammer. The hammering action and the risk of damage to the die or end effector are significant technical problems.

• *Pressworking.* Human operators work at considerable risk in sheetmetal pressworking operations because of the action of the press. Robots are used to substitute for the workers to reduce the danger. In these applications, the robot loads the blank into the press, then the stamping operation is performed, and the part falls out of the machine into a container.

• *Heat-treating.* These are often relatively simple operations in which the robot loads and/or unloads parts from a furnace.

Processing Operations
In processing applications, the robot performs some operation on a work part, such as grinding or spray painting.

Spot Welding. Spot welding is a metal joining process in which two sheet metal parts are fused together at localized points of contact. Two electrodes squeeze the metal parts together and then a large electrical current is applied across the contact point to cause fusion to occur. The use of industrial robots in this application has dramatically improved the consistency of the welds.

Robots used for spot welding are usually large, with sufficient payload capacity to wield the heavy welding gun. Five or six axes are generally required to achieve the required position and orientation of the welding gun. Playback robots with point-to-point control are used. Jointed-arm robots are the most common type in automobile spot-welding lines, which may consist of several dozen robots.

Arc Welding. Arc welding is used to provide continuous welds rather than individual spot welds at specific contact points. The resulting arc-welded joint is substantially stronger than in spot welding. The robot used in arc welding must be capable of continuous path control. Jointed arm robots consisting of six joints are frequently used. Some robot vendors provide manipulators that have hollow upper arms, so that the cables connected to the welding torch can be contained in the arm for protection, rather than attached to the exterior. Also, programming improvements for arc welding based on CAD/CAM have made it much easier and faster to implement a robot welding cell.

Spray Coating. Spray coating directs a spray gun at the object to be coated. Fluid (e.g., paint) flows through the nozzle of the spray gun to be dispersed and applied over the surface of the object. Spray painting is the most common application in the category, but spray coating refers to a broader range of applications that includes painting.

Assembly and Inspection
In some respects, assembly and inspection are hybrids of the previous two categories: material handling and processing. Assembly and inspection can involve either the handling of materials or the manipulation of a tool.

Assembly
Assembly involves the combining of two or more parts to form a new entity, called a subassembly or assembly. The new entity is made secure by fastening the parts together using mechanical fastening techniques (e.g., screws, bolts and nuts, rivets) or joining processes (e.g., welding, brazing, soldering, or adhesive bonding). Industrial robots used for the types of assembly operations described here are typically small, with light load capacities. The most common configurations are jointed arm, SCARA, and Cartesian coordinate.

Inspection.
There is often a need in automated production to inspect the work that is done. Inspections accomplish the following functions: (1) making sure that a given process has been completed, (2) ensuring that parts have been assembled as specified, and (3) identifying flaws in raw materials and finished parts.

6.a **Generation of Robots**

Engineers and scientists have analyzed the evolution of robots, marking progress according to robot generations.

First generation

A first-generation robot is a simple mechanical arm. These machines have the ability to make precise motions at high speed, many times, for a long time. Such robots find widespread industrial use today. First-generation robots can work in groups, such as in an automated integrated manufacturing system (AIMS), if their actions are synchronized. The operation of these machines must be constantly supervised, because if they get out of alignment and are allowed to keep working, the result can be a series of bad production units.

Second generation

A second-generation robot has rudimentary machine intelligence. Such a robot is equipped with sensors that tell it things about the outside world. These devices include pressure sensors, proximity sensors, tactile sensors, radar, sonar, ladar, and vision systems. A controller processes the data from these sensors and adjusts the operation of the robot accordingly. These devices came into common use around 1980. Second-generation robots can stay synchronized with each other, without having to be overseen constantly by a human operator. Of course, periodic checking is needed with any machine, because things can always go wrong; the more complex the system, the more ways it can malfunction.

Third generation

The concept of a third-generation robot encompasses two major avenues of evolving smart robot technology: the autonomous robot and the insect robot. An autonomous robot can work on its own. It contains a controller, and it can do things largely without supervision, either by an outside computer or by a human being. A good example of this type of thirdgeneration robot is the personal robot about which some people dream. There are some situations in which autonomous robots do not perform efficiently. In these cases, a fleet of simple insect robots, all under the control of one central computer, can be used. These machines work like ants in an anthill, or like bees in a hive.While the individual machines lack artificial intelligence (AI), the group as a whole is intelligent.

Fourth generation and beyond

Any robot of a sort yet to be seriously put into operation is a fourthgeneration robot. Examples of these might be robots that reproduce and evolve, or that incorporate biological as well as mechanical components. Past that, we might say that a fifth-generation robot is something no one has yet designed or conceived.

6.b. **ROBOT END EFFECTORS**

An end effector is a device that attaches to the wrist of the robot arm and enables the general-purpose robot to perform a specific task. It is sometimes referred to as the robot‘s “hand.”

Types of end effectors
The End effectors can be divided into two major categories:

1 . Grippers
2. Tools

Grippers are end effectors used to grasp and manipulate objects during the work cycle. The objects are usually work parts that are moved from one location to another in the cell. Machine loading and unloading applications fall into this category. Owing to the variety of part shapes, sizes, and weights, most grippers must be custom designed. Types of grippers used in industrial robot applications include the following:

• *Mechanical grippers*, consisting of two or more fingers that can be actuated by the robot controller to open and close on the work part (Figure 8.10 shows a two-finger gripper)
• *Vacuum grippers*, in which suction cups are used to hold flat objects
• *Magnetized devices*, for holding ferrous parts
• *Adhesive devices*, which use an adhesive substance to hold a flexible material such as a fabric
• *Simple mechanical devices*, such as hooks and scoops.

Mechanical grippers are the most common gripper type. Some of the innovations and advances in mechanical gripper technology include:

• *Dual grippers*, consisting of two gripper devices in one end effector for machine loading and unloading. With a single gripper, the robot must reach into the production machine twice, once to unload the finished part and position it in a location external to the machine, and the second time to pick up the next part and load it into the machine. With a dual gripper, the robot picks up the next work part while the machine is still processing the previous part. When the machine cycle is finished, the robot reaches into the machine only once: to remove the finished part and load the next part. This reduces the cycle time per part.

ii) **Sensors in Robotics**

The sensors used in robotics include a wide range of devices which can be divided into the following general categories:

a. Tactile sensors
b. Proximity and range sensors
c. Miscellaneous sensors and sensor based systems
d. Machine vision systems

Tactile Sensors

Tactile sensors are devices which indicate contact between themselves and some other solid object. Tactile sensing devices can be classified into two classes: Touch Sensors and Force sensors.

Touch sensors provide a binary output signal which indicates whether or not contact has been made with the object.

Force sensors (Stress Sensors) indicate not only the contact has been made with the object but also the magnitude of the contact force between the two objects.

Touch sensors

Touch sensors are used to indicate that contact has been made between two objects without regard to the magnitude of the contacting force. Included within this category are simple devices such as limit switches, micro-switches, and the like. The simpler devices are frequently used in the design of interlock systems in robotics. For example, they can be used to indicate the presence or absence of parts in a fixture or at the pick-up point along a conveyor.

Another use for a touch sensing device would be as part of an inspection probe which is manipulated by the robot to measure dimensions on a work part. A robot with 6 degrees of freedom would be capable of accessing surfaces on the part that would be difficult for a three-axis coordinate measuring machine, the inspection system normally considered for such an inspection task. Unfortunately, the robot's accuracy would be a limiting factor in contact inspection work.

Force Sensors

The capacity to measure the forces permits the robot to perform a number of tasks. These include the capability to grasp parts of different sizes in material handling, machine loading and assembly work, applying the appropriate level of force for the given part.

Force sensing in robotics can be accomplished in several ways. A commonly used technique is a "force sensing wrist". This consists of a special load-cell mounted between the gripper and the wrist. Another technique is to measure the torque being exerted by each joint. A third technique is to form an array of force-sensing elements so that the shape and other information about the contact surface can be determined.

Tactile array sensors

A tactile array sensor is a special type of force sensor composed of a matrix of force-sensing elements. The force data provided by this type of device may be combined with pattern recognition techniques to describe a number of characteristics about the impression contacting the array sensor surface. Among these characteristics are (1) the presence of an object, (2) the object's contact area, shape, location, and orientation, (3) the pressure and pressure distribution, and (4) force magnitude and location. Tactile array sensors can be mounted in the fingers of the robot gripper or attached to a work table as a flat touch surface. Figures 6-3 and 6-4 illustrate these two possible mountings for the sensor device.

iii) **Accuracy**

Accuracy refers to a robot's ability to position its wrist end at a desired target point within the work volume. The accuracy of a robot can be defined in terms of spatial resolution because the ability to achieve a given target-point depends on how closely the robot can define the control increments for each of its joint motions. In the worst case, the desired point would lie in the middle between two adjacent control increments.

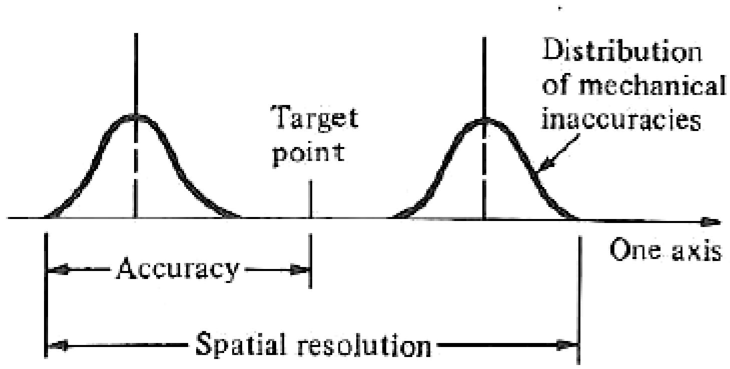


Fig: Illustration of accuracy and spatial resolution in which mechanical inaccuracies are represented by a spatial resolution.

First, the accuracy varies within the work volume, tending to be worse when the arm is in the outer range of its work volume and better when the arm is closer to its base. The reason for this is that the mechanical inaccuracies are magnified with the robot's arm fully extended. The term error map is used to characterize the level of accuracy possessed by the robot as a function of location in the work volume.

Second, the accuracy is improved if the motion cycle is restricted to a limited work range. The mechanical errors will tend to be reduced when the robot is exercised through a restricted range of motions. The robot's ability to reach a particular reference point within the limited work space is sometimes called its local accuracy. When the accuracy is assessed within the robot's full work volume, the term global accuracy is used.

A third factor influencing accuracy is the load being carried by the robot. Heavier workloads cause greater deflection of the mechanical links of the robot, resulting in lower accuracy.

Repeatability

Repeatability is concerned with the robot's ability to position its wrist or an end effector attached to its wrist at a point in space that had previously been taught to the robot. Repeatability and accuracy has to two different aspects of the robot's precision.

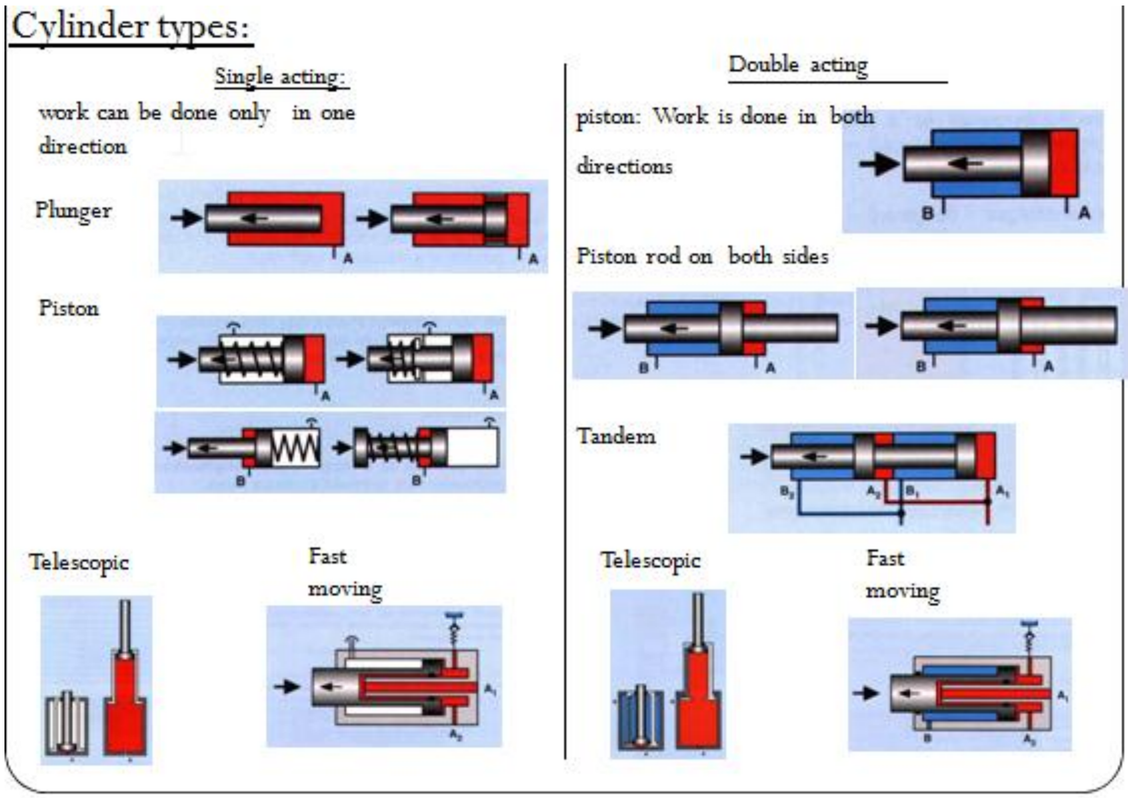
Accuracy relates to the robot's capacity to be programmed to achieve a given target point. The actual programmed point will probably be different from the target point due to limitations of control resolution. Repeatability refers to the robot's ability to return to the programmed point when commanded to do so.

7.a. . Hydraulic actuators

Hydraulic actuators allow a robot to move by the use of fluids moving under pressure through a series of valves by the use of pumps. The hydraulic fluids normally consist of oils which are reasonably non-compressible.

Hydraulic and Pneumatic actuators are classified as

- Linear Actuators (Cylinders)
- Rotary Actuators (Motors)



- A Hydraulic motor is a mechanical actuator that converts hydraulic pressure and flow into torque and angular displacement (rotation).
- Several types of hydraulic motors:
    - Gear Motors
    - Vane Motors
    - Gerotor Motors
    - Radial Piston Motors

7.b.
i)Velocity Sensors Tachometers

A tachometer (also called a revolution-counter, rev-counter, or RPM gauge) is an instrument that measures the rotation speed of a shaft or disk, as in a motor or other machine. The device usually displays the revolutions perminute (RPM) on a calibrated analogue dial, but digital displays are increasingly common.
Tachometers can be divided in to
1. DC (Digital)tachometer
2. AC (analog) tachometer

DCTachometer (Angular Velocity)

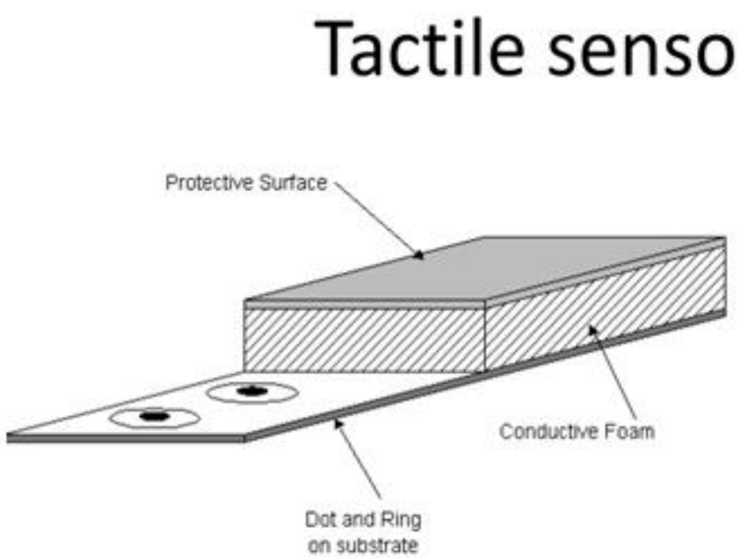
• A tachometer is essentially DC generator providing an output voltage proportional to the angular velocity of the armature
• The rotor is directly connected to the rotating object.
• The output signal that is induced at the rotating coil is picked up using a commutator

device (consists of low resistance carbon brushes)
Permanent Magnet AC Tachometer

• When the rotor is stationary or moving in a quasi-static manner the output voltage will be constant
• As the rotor moves, an additional voltage, proportional to the speed of the rotor will be induced
• The output is an amplitude modulated signal proportional to the rotor speed and demodulation is necessary
• Direction is obtained from the phase angle
• Due to commutator in DC tachometers a slight ripples will appear in output voltage which can not be filtered out, this can overcome by using AC tachometer

ii. Tactile sensor

- A tactile sensor can measure any physical interaction with its environment.
- This sensor is capable to measure the parameters related to the contact of the sensor with an object.
- These sensors are designed according to the biological sense of cutaneous touch, which is able to detect the stimuli from mechanical stimulation, temperature, and pain.
- A tactile sensor will receive and respond to a signal and this signal could be a force or a physical contact.
- Basically a tactile sensor is a touch sensor that can provide information about the object which it.is in contact with it.
- The information about the object could be the shape, size, and type of material.



8.a. Joint space is defined by **a vector whose components are the translational and angular displacements of each joint of a robotic link**. The common linear PID does not include any component of the robot dynamics into its control law whenever it is used in joint space or task-space. Joint space is the parameterisation given by the set of joint variables. For example for a SCARA robot with a single degree of freedom in the wrist (θ1, θ2, d3, θ4).

Actuator space is associated with the mechanism used to actuate a joint. Thus, in certain situations a linear actuator (say a hydraulic cylinder) is used to actuate a revolute joint. The actuator space has each of its coordinate axes defined by one of the actuator variables.

8.b.

In inverse kinematic problem, we find a solution θ given a set of position and orientation of the end-effector. For PUMA 560 robot manipulator, we are given the DH model T^0_6 which contains the orientation (rotation matrix) and the position (displacement matrix) of the end-effector. The angles at each joint is determined by decoupling the PUMA 560 robot manipulator. We initial solve the position (P_{5x}, P_{5y}, P_{5z}) of the frame 5 and used these coordinate values to determine the angles (θ_1, θ_2, θ_3) at the first three joint.

We start by determining the position of the frame 5 as shown in Figure 1. That is

$$P_5 = \begin{bmatrix} P_x - d_6 a_x \\ P_y - d_6 a_y \\ P_z - d_6 a_z \end{bmatrix} . \qquad (1)$$

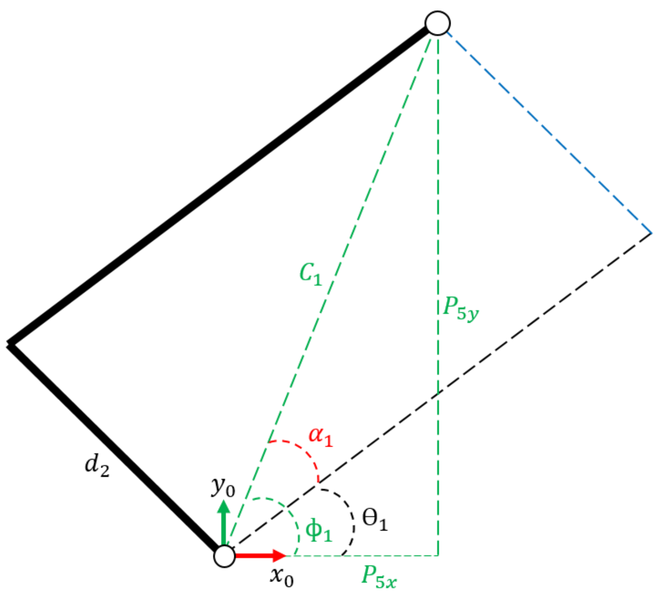


Figure 2: $x_0 - y_0$ plane (Top View)

Next, we solve the angles at the first three joints by viewing PUMA 560 robot manipulator movement across $x_0 - y_0$ and $x_0 - z_0$ planes. In Figure 2, the joint angle θ1 is defined by the difference between angles φ_1 and α_1. The angle α_1 is defined as

$$\alpha_1 = asin\left(\frac{d_2}{C_1}\right) \qquad (2)$$

where

$$C_1 = \sqrt{P_{5x}^2 + P_{5y}^2}$$

Equation (2) can be expressed in terms of atan2 function. We define

$$D_1 = \frac{d_2}{\sqrt{P_{5x}^2 + P_{5y}^2}}$$

and yield

$$\alpha_1 = atan2\left(D_1, \sqrt{1 - D_1^2}\right) \qquad (3)$$

The angle φ_1 is defined as

$$\phi_1 = atan2(P_{5y}, P_{5x}) \qquad (4)$$

The joint angle θ_1 is derived as

$$\theta_1 = atan2(P_{5y}, P_{5x}) - atan2(D_1, \sqrt{1 - D_1^2}) \qquad (5)$$

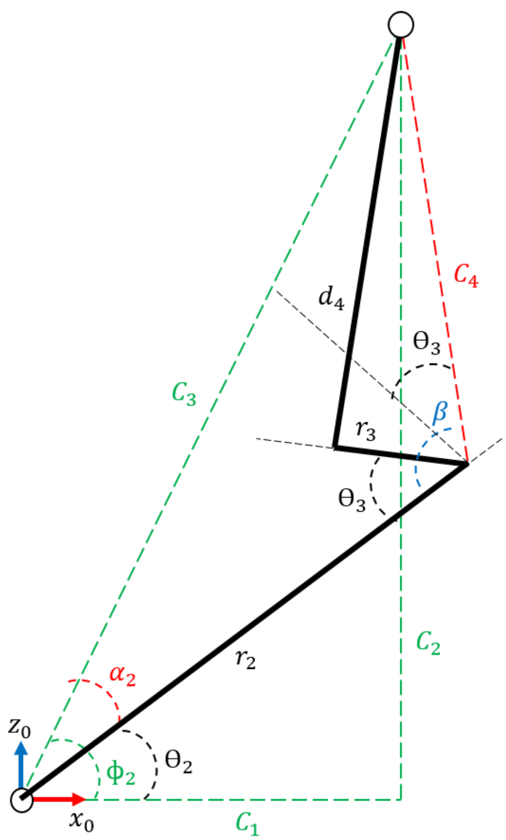


Figure 3: $x_0 - z_0$ plane (Side View)

In Figure 3, we derived the equations to solve for the joint angles θ_1 and θ_2. We define the distances

$$C_2 = P_{5x} - d_1 \quad (6)$$

$$C_3 = \sqrt{C_1^2 + C_2^2} \quad (7)$$

$$C_4 = \sqrt{r_3^2 + d_4^2} \quad (8)$$

and apply cosine law to determine angles, α2 and β. We get the equations

$$\cos \alpha_2 = \frac{C_3^2 + r_2^2 - C_4^2}{2r_2C_3} \quad (9)$$

and

$$\cos \beta = \frac{C_4^2 + r_2^2 - C_3^2}{2r_2C_4} \quad (10)$$

that gives the cosine value of α and β. We set

$D_2 = \frac{C_3^2+r_2^2-C_4^2}{2r_2C_3}$

and

$D_3 = \frac{C_4^2+r_2^2-C_3^2}{2r_2C_4}$

to define

$$\alpha_2 = atan2\left(\sqrt{1 - D_2^2}, D_2\right) \quad (11)$$

and

$$\beta = atan2\left(\sqrt{1 - D_3^2}, D_3\right). \quad (12)$$

The angle φ_2 is determined by

$$\phi_2 = atan2(C_2, C_1). \quad (13)$$

The joint angle θ_2 is the difference between angles φ_2 and α_2 as shown in Figure 3. That is

$$\theta_2 = atan2(C_2, C_1) - atan2\left(\sqrt{1 - D_2^2}, D_2\right). \quad (14)$$

For the joint angle θ_3, we have

$$\theta_3 = atan2\left(\sqrt{1 - D_3^2}, D_3\right) - 90. \quad (15)$$

After we obtain the angles at the first three joints, we used the angles to solve for the forward kinematics for frame 0 to frame 3. We solve the homogeneous transformation matrix T^3_6 for frame 3 to frame 6 by taking the dot product between T^0_6 and the inverse of T^0_3. In general, DH model for 6 degrees of freedom manipulator is defined as

$$T_6^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5. \quad (16)$$

We can simplify the model to $T^0_6 = T^0_3 T^3_6$ and define

$$T_6^3 = T_3^0 \cdot [T_6^3]^{-1}. \quad (17)$$

The angles at joint 4 to 6 is determine by deriving a set of equations from the homogeneous transformation matrix T^3_6. The rotation matrix R^3_6 holds the orientation of the end-effector in reference to frame 3. That is

$$R_6^3 = \begin{bmatrix} c\theta_4 c\theta_5 c\theta_6 - s\theta_4 s\theta_6 & c\theta_4 c\theta_5 s\theta_6 - c\theta_6 s\theta_4 & c\theta_4 s\theta_5 \\ c\theta_4 s\theta_6 + c\theta_5 c\theta_6 s\theta_4 & c\theta_4 c\theta_6 + c\theta_5 s\theta_4 s\theta_6 & s\theta_4 s\theta_5 \\ -c\theta_6 s\theta_5 & s\theta_5 s\theta_6 & c\theta_5 \end{bmatrix} \quad (18)$$

where the notation, sθ = sin θ and cθ = cos θ, is adapted. The R^3_6 is a component in the homogeneous transformation matrix T^3_6. We derived the joint angles θ_4, θ_5 and θ_6 by equating the numerical values of T^3_6 in equation (26).
We obtain the joint angle θ_4 by

$$\frac{\sin\theta_4 \sin\theta_5}{\cos\theta_4 \sin\theta_5} = \frac{T_6^3(2,3)}{T_6^3(1,3)}. \quad (19)$$

The joint angle θ_4 is defined as

$$\theta_4 = atan2\Big(T_6^3(2,3), T_6^3(1,3)\Big). \quad (20)$$

For joint angle θ_5, we have $\cos\theta_5 = T^3_6(3, 3)$ and used this equation to find θ_5 in term of atan2 function. The joint θ_5 is defined as

$$\theta_5 = atan2\Big(\sqrt{1 - T_6^3(3,3)^2}, T_6^3(3,3)\Big). \quad (21)$$

Lastly, the joint angle θ_6 is derived by

$$\frac{\sin\theta_5 \sin\theta_6}{-\cos\theta_6 \sin\theta_5} = \frac{T_6^3(3,2)}{T_6^3(3,1)}. \quad (22)$$

We have the joint angle θ_6 as

$$\theta_6 = atan2\Big(T_6^3(3,2), -T_6^3(3,1)\Big). \quad (23)$$

9.a Robot Programming is a art of teaching a robot online or offline a series of tasks to be performed with required accuracy, velocity and repeatability.
Three levels of Programming

1. **Teach by showing**
2. **Explicit robot programming languages**
- Specialized manipulation languages
- Robot library for an existing computer language
- Robot library for a new general-purpose language

3. **Task-level programming languages**

1. **Teach by showing**

- Early robots were all programmed by a method that we will call **teach by showing,** which involved moving the robot to a desired goal point and recording its position in a memory that the sequencer would read during playback.
- During the teach phase, the user would guide the robot either by hand or through interaction with a **teach pendant.**
- Teach pendants are handheld button boxes that allow control of each manipulator joint or of each Cartesian degree of freedom.
- Some such controllers allow testing and branching, so that simple programs involving logic can be entered.
- Some teach pendants have alphanumeric displays and are approaching hand-held terminals in complexity.

2. Explicit robot programming languages

Ever since the arrival of inexpensive and powerful computers, the trend has been increasingly toward programming robots via programs written in computer programming languages.

- Usually, these computer programming languages have special features that apply to the problems of programming manipulators and so are called **robot programming languages** (RPLs).
- Most of the systems that come equipped with a robot programming language have nonetheless retained a teach-pendant-style interface also.
- Robot programming languages have likewise taken on many forms. We will split them into three categories:
- Specialized manipulation languages
- Robot library for an existing computer language
- Robot library for a new general-purpose language

3. **Task-level programming languages**

- The third level of robot programming methodology is embodied in **task-level programming languages.**
- These languages allow the user to command desired subgoals of the task directly, rather than specify the details of every action the robot is to take.
- In such a system, the user is able to include instructions in the application program at a significantly higher level than in an explicit robot programming language.
- A task-level robot programming system must have the ability to perform many planning tasks automatically.
- For example, if an instruction to "grasp the bolt" is issued, the system must plan a path of the manipulator that avoids collision with any surrounding obstacles, must automatically choose a good grasp location on the bolt, and must grasp it. In contrast, in an explicit robot programming language, all these choices must be made by the programmer.

9.b. **REQUIREMENTS OF A ROBOT PROGRAMMING LANGUAGE**

1. World modeling

- Manipulation programs must, by definition, involve moving objects in three-dimensional space, so it is clear that any robot programming language needs a means of describing such actions.

- The most common element of robot programming languages is the existence of special **geometric types.**
- Given a robot programming environment that supports geometric types, the robot and other machines, parts, and fixtures can be modeled by defining named variables associated with each object of interest.

2. Motion specification

- A very basic function of a robot programming language is to allow the description of desired motions of the robot.
- Through the use of motion statements in the language, the user interfaces to path planners and generators of the style.
- Motion statements allow the user to specify via points, the goal point, and whether to use joint-interpolated motion or Cartesian straight-line motion.

3. Flow of execution

- As in more conventional computer programming languages, a robot programming system allows the user to specify the flow of execution—that is, concepts such as testing and branching, looping, calls to subroutines, and even interrupts are generally found in robot programming languages.
- More so than in many computer applications, parallel processing is generally important in automated workcell applications.

4. Programming environment

- As with any computer languages, a good programming environment fosters pro- grammer productivity.
- Manipulator programming is difficult and tends to be very interactive, with a lot of trial and error.
- If the user were forced to continually repeat the "edit-compile-run" cycle of compiled languages, productivity would be low.

5. Sensor integration

- An extremely important part of robot programming has to do with interaction with sensors.
- Integration with a vision system allows the vision system to send the manipulator system the coordinates of an object of interest.
- The interface to force-control capabilities, comes through special language statements that allow the user to specify force strategies.

10.a. Offline Programming (or Off-Line Programming) means programming robots outside the production environment. Offline Programming eliminates production downtime caused by shopfloor programming. Simulation and Offline Programming allows studying multiple scenarios of a robot work cell before setting up the production cell. Mistakes commonly made in designing a work cell can be predicted in time.

Offline Programming is the best way to maximize return on investment for robot systems and it requires appropriate simulation tools. The time for the adoption of new programs can be cut from weeks to a single day, enabling the robotization of short-run production.

Offline programming is one of the programming methods commonly used with industrial robots. Teach pendants have been the most popular robotic programming method as they are a key component of any industrial robot and allow for on the go programming. However, offline programming is becoming more common and gaining ground on teach pendants..

Robotic offline programming (OLP) involves using software on an external computer to create a robotic application program outside of the production environment. Unlike teach pendants which use online programming, OLP designs articulated robot programs before involving the actual robot. Robotic offline programming is often confused and interchanged with offline robot simulation software, however, these two are actually not the same. Offline simulation involves testing a robot program outside of the production environment and may be used in conjunction with offline programming.

When to Use Offline Programming

Offline programming can be used for most robotic systems but there are some scenarios where it is more beneficial than other robotic programming methods. OLP is ideal for any complex applications that involve several steps. Opting for offline programming can save significant time since complex applications can involve an extensive amount of manual programming when using a teach pendant. It is also ideal for manufacturing processes involving large workpieces or those with high part mixes. It can also be useful for programming robots operating in low volume productions. The only time offline programming is not recommended is for simple applications with minimal steps as a teach pendant is still best for those.

Advantages of Offline Programming

The advantages of offline programming robots over other programming methods is the reason behind its sudden growth. Programming robots outside of the production environment eliminates downtime associated with programming. It is estimated downtime can be reduced from forty hours to just four hours or by one-tenth. Complex applications can go from taking several weeks to implement to just one day, increasing robot uptime and productivity.

b. PROBLEMS PECULIAR TO ROBOT PROGRAMMING LANGUAGES

Robot programming shares all the problems of conventional computer programming, plus some additional difficulties caused by effects of the physical world.

1. Internal world model versus external reality
2. Context sensitivity
3. Error recovery

Internal world model versus external reality

• A central feature of a robot programming system is the world model that is maintained internally in the computer.

• Even when this model is quite simple, there are ample difficulties in assuring that it matches the physical reality that it attempts to model.

• Discrepancies between internal model and external reality result in poor or failed grasping of objects, collisions, and a host of more subtle problems.

• This correspondence between internal model and the external world must be established for the program's initial state and must be maintained throughout its execution.

• During initial programming or debugging, it is generally up to the user to suffer the burden of ensuring that the state represented in the program corresponds to the physical state of the workcell.

Context sensitivity

• Bottom-up programming is a standard approach to writing a large computer program in which one develops small, low-level pieces of a program and then puts them together into larger pieces, eventually attaining a completed program.
• For this method to work, it is essential that the small pieces be relatively insensitive to the language statements that precede them and that there be no assumptions concerning the context in which these program pieces execute.

• Manipulator programs can be highly sensitive to initial conditions—for exam- ple, the initial manipulator position.

• In motion trajectories, the starting position will influence the trajectory that will be used for the motion.
• The initial manipulator position might also influence the velocity with which the arm will be moving during some critical part of the motion.

Error recovery

• Another direct consequence of working with the physical world is that objects might not be exactly where they should be and, hence, motions that deal with them could fail.
• Part of manipulator programming involves attempting to take this into account and making assembly operations as robust as possible, but, even so, errors are likely, and an important aspect of manipulator programming is how to recover from these errors.
• Almost any motion statement in the user's program can fail, sometimes for a variety of reasons. Some of the more common causes are objects shifting or dropping out of the hand, an object missing from where it should be, jamming during an insertion, and not being able to locate a hole.
• The first problem that arises for error recovery is identifying that an error has indeed occurred.
• Because robots generally have quite limited sensing and reasoning capabilities, error detection is often difficult.

c. CENTRAL ISSUES IN OLP SYSTEMS

1 User interface

• A major motivation for developing an OLP system is to create an environment that makes programming manipulators easier.
• However, another major motivation is to remove reliance on use of the physical equipment during programming.
• Manufacturers of industrial robots have learned that the RPLs they provide with their robots cannot be utilized successfully by a large percentage of manufacturing personnel.
• For this reasons, many industrial robots are provided with a two-level interface, one for programmers and one for nonprogrammers.
• Nonprogrammers utilize a teach pendant and interact directly with the robot to develop robot programs.
• Programmers write code in the RPL and interact with the robot in order to teach robot work points and to debug program flow.
• For use as an RPL, interactive languages are much more productive than compiled languages.
• A well-designed user interface should enable nonprogrammers to accomplish many applications from start to finish.

2. 3-D modeling

• A central element in OLP systems is the use of graphic depictions of the simulated robot and its work cell.
• This requires the robot and all fixtures, parts, and tools in the work cell to be modeled as three-dimensional objects
• To speed up program development, it is desirable to use any CAD models of parts or tooling that are directly available from the CAD system on which the original design was done.
• If an OLP system is to be a stand-alone system, it must have appropriate interfaces to transfer models to and from external CAD systems.
• An important use of the three-dimensional geometry of the object models is in automatic collision detection, example in assembly

3. Kinematic emulation
• A central component in maintaining the validity of the simulated world is the faithful emulation of the geometrical aspects of each simulated manipulator.
• With regard to inverse kinematics, the OLP system can interface to the robot controller in two distinct ways.
• First, the OLP system could replace the inverse kinematics of the robot controller and always communicate robot positions in mechanism joint space.
• The second choice is to communicate Cartesian locations to the robot controller and let the controller use the inverse kinematics supplied by the manufacturer to solve for robot configurations.
• The simulator must use the same algorithm as the controller in order to avoid potentially catastrophic errors in simulating the actual manipulator
4 Path-planning emulation
• In addition to kinematic emulation for static positioning of the manipulator, an OLP system should accurately emulate the path taken by the manipulator in moving through space.
• Again, the central problem is that the OLP system needs to simulate the algorithms in the employed robot controller, and such path- planning and -execution algorithms vary considerably from one robot manufacturer to another.
• When a robot is operating in a moving environment (e.g., near another robot), accurate simulation of the temporal attributes of motion is necessary to predict collisions accurately

5. Dynamic emulation
• Simulated motion of manipulators can neglect dynamic attributes if the OLP system does a good job of emulating the trajectory-planning algorithm of the controller and if the actual robot follows desired trajectories with negligible errors.

• However, at high speed or under heavy loading conditions, trajectory- tracking errors can become important.

• Simulation of these tracking errors necessitates both modeling the dynamics of the manipulator and of the objects that it moves and emulating the control algorithm used in the manipulator controller.
• Currently, practical problems exist in obtaining sufficient information from the robot vendors to make this kind of dynamic simulation of practical value, but, in some cases, dynamic simulation can be pursued fruitfully.

10.d AUTOMATING SUBTASKS IN OLP SYSTEMS

- Automatic robot placement
- Collision avoidance and path optimization
- Automatic planning of coordinated motion
- Force-control simulation
- Automatic scheduling
- Automatic assessment of errors and tolerances

Automatic robot placement

- One of the most basic tasks is the determination of the work cell layout so that the manipulator(s) can reach all of the required work points.
- Determining correct robot or work piece placement is by trial and error and is more quickly completed in a simulated world than in the physical cell.

- Automatic placement can be computed by direct search or by heuristic- guided search techniques

- Most robots are mounted flat on the floor (or ceiling) and have the first rotary joint perpendicular to the floor.
- Feasibility can be defined as collision-free ability to reach all work points.
- The result of such an automatic placement is a cell in which the robot can reach all of its work points in well-conditioned configurations.

Collision avoidance and path optimization

- Research on the planning of collision-free paths and the planning of time optimal paths
- Some related problems that have a smaller scope and a smaller search space are also of interest.
- For example, consider the problem of using a six-degree-of-freedom robot for an arc-welding task whose geometry specifies only five degrees of freedom.
- Automatic planning of the redundant degree of freedom can be used to avoid collisions and singularities of the robot.

Automatic planning of coordinated motion

- In many arc-welding situations, details of the process require that a certain relationship between the work piece and the gravity vector be maintained during the weld.
- This results in a two- or three-degree-of-freedom-orienting system on which the part is mounted, operating simultaneously with the robot and in a coordinated fashion.
- In such a system, there could be nine or more degrees of freedom to coordinate.
- Such systems are generally programmed today by using teaching-pendant techniques.
- A planning system that could automatically synthesize the coordinated motions for such a system might be quite valuable.

Force-control simulation

• In a simulated world in which objects are represented by their surfaces, it is possible to investigate the simulation of manipulator force-control strategies.
• This task involves the difficult problem of modeling some surface properties and expanding the dynamic simulator to deal with the constraints imposed by various contacting situations.

• In such an environment, it might be possible to assess various force- controlled assembly operations for feasibility

Automatic scheduling

• Along with the geometric problems found in robot programming, there are often difficult scheduling and communication problems.
• This is particularly the case if we expand the simulation beyond a single work cell to a group of work cells.
• Some discrete-time simulation systems offer abstract simulation of such systems, but few offer planning algorithms.
• Planning schedules for interacting processes is a difficult problem and an area of research.
• An OLP system would serve as an ideal test bed for such research and would be immediately enhanced by any useful algorithms in this area.

Automatic assessment of errors and tolerances

• An OLP system might be given some of the capabilities discussed in recent work in modeling positioning-error sources and the effect of data from imperfect sensors
• The world model could be made to include various error bounds and tolerance information, and the system could assess the likelihood of success of various positioning or assembly tasks.
• The system might suggest the use and placement of sensors so as to correct potential problems.
• Off-line programming systems are useful in present-day industrial applications
and can serve as a basis for continuing robotics research and development.
• A large motivation in developing OLP systems is to fill the gap between the explicitly programmed systems available today and the task-level systems of tomorrow.