**Module 3**

**Machine Vision & Artificial Intelligence:** Introduction to machine vision, The sensing and digitizing function in machine vision, image processing and analysis: image data reduction, segmentation, feature extraction, object recognition, training the vision system, robotic applications.

**Artificial Intelligence (AI):** Goals of AI in research, AI techniques: knowledge representation, problem representation and problem solving and search techniques in problem solving. [Textbook- 1]

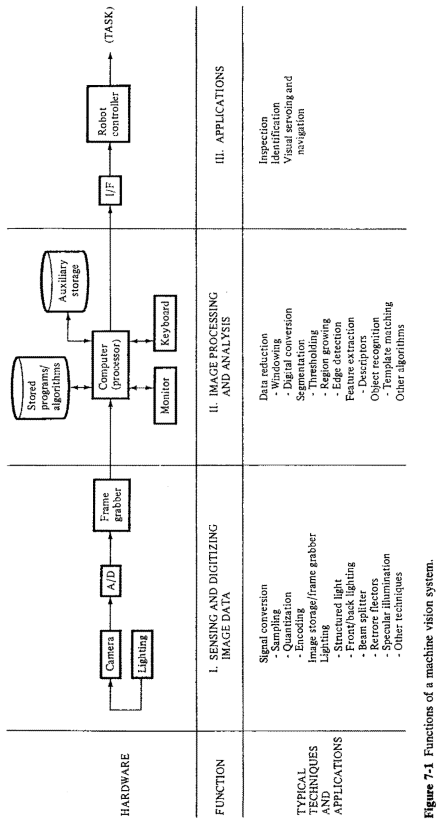
Machine vision (other names include computer vision and artificial vision) is an important sensor technology with potential applications in many industrial operations. Many of the current applications of machine vision are in inspection; however, it is anticipated that vision technology will play an increasingly significant role in the future of robotics. Vision systems designed to be utilized with robot or manufacturing systems must meet two important criteria which currently limit the influx of vision systems to the manufacturing community. The first of these criteria is the need for a relatively low-cost vision system, typically under $30,000. The second criterion is the need for relatively rapid response time needed for robot or manufacturing applications, typically a fraction of a second.

**3a.1 INTRODUCTION TO MACHINE VISION**

Machine vision is concerned with the sensing of vision data and its interpretation by a computer. The typical vision system consists of the camera and digitizing hardware, a digital computer, and hardware and software necessary to interface them. This interface hardware and software is often referred to as a preprocessor. The operation of the vision system consists of three functions:

1. Sensing and digitizing image data 2. Image processing and analysis 3. Application

The relationships between the three functions are illustrated in the diagram of Fig. 7-1 The sensing and digitizing functions involve the input of vision data by means of a camera focused on the scene of interest. Special lighting techniques are frequently used to obtain an image of sufficient contrast for later processing. The image viewed by the camera is typically digitized and stored in computer memory. The digital image is called a frame of vision data, and is frequently captured by a hardware device called a frame grabber. These devices are capable of digitizing images at the rate of 30 frames per second. The frames consist of a matrix of data representing projections of the scene sensed by the camera. The elements of the matrix are called picture elements, or pixels. The number of pixels are determined by a sampling process performed on each image frame. A single pixel is the projection of a small portion of the scene which reduces that portion to a single value. The value is a measure of the light intensity for that element of the scene. Each pixel intensity is converted into a digital value. (We are ignoring the additional complexities involved in the operation of a color video camera).



The digitized image matrix for each frame is stored and then subjected to image processing and analysis functions for data reduction and interpretation of the image. These steps are required in order to permit the real-time application of vision analysis required in robotic applications. Typically an image frame will be thresholded to produce a binary image, and then various feature measurements will further reduce the data representation of the image. This data reduction can change the representation of a frame from several hundred thousand bytes of raw image data to several hundred bytes of feature value data. The resultant feature data can be analyzed in the available time for action by the robot system.

Various techniques to compute the feature values can be programmed into the computer to obtain feature descriptors of the image which are matched against previously computed values stored in the computer. These descriptors include shape and size characteristics that can be readily calculated from the thresholded image matrix.

To accomplish image processing and analysis, the vision system frequently must be trained. In training, information is obtained on prototype objects and stored as computer models. The information gathered during training consists of features such as the area of the object, its perimeter length, major and minor diameters, and similar features. During subsequent operation of the system, feature values computed on unknown objects viewed by the camera are compared with the computer models to determine if a match has occurred.

The third function of a machine vision system is the applications function. The current applications of machine vision in robotics include inspection, part identification, location, and orientation. Research is ongoing in advanced applications of machine vision for use in complex inspection, guidance, and navigation.

Vision systems can be classified in a number of ways. One obvious classification is whether the system deals with a two-dimensional or three dimensional model of the scene. Some vision applications require only a two-dimensional analysis. Examples of two-dimensional vision problems include checking the dimensions of a part or verifying the presence of components on a subassembly.

Many two-dimensional vision systems can operate on a binary image which is the result of a simple thresholding technique. This is based on an assumed high contrast between the object (s) and the background. The desired contrast can often be accomplished by using a controlled lighting system.

Three-dimensional vision systems may require special lighting techniques and more sophisticated image processing algorithms to analyze the image. Some systems require two cameras in order to achieve a stereoscopic view of the scene, while other three-dimensional systems rely on the use of structured light and optical triangulation techniques with a single camera. An example of a structured light system is one that projects a controlled band of light across the object. The light band is distorted according to the three-dimensional shape of the object. The vision system sees the distorted band and utilizes triangulation to deduce the shape.

Another way of classifying vision systems is according to the number of gray levels (light intensity levels) used to characterize the image. In a binary image the gray level values are divided into either of two categories, black or white. Other systems permit the classification of each pixel's gray level into various levels, the range of which is called a gray scale.

**3a.2 THE SENSING AND DIGITIZING FUNCTION IN MACHINE VISION**

Our description of the typical machine vision system in the preceding section identified three functions: sensing and digitizing, image processing and analysis, and application.

Image sensing requires some type of image formation device such as a camera and a digitizer which stores a video frame in the computer memory. We divide the sensing and digitizing functions into several steps. The initial step involves capturing the image of the scene with the vision camera. The image consists of relative light intensities corresponding to the various portions of the scene. These light intensities are continuous analog values which must be sampled and converted into digital form.

The second step, digitizing, is achieved by an analog-to-digital (A/D) converter. The AID converter is either a part of a digital video camera or the front end of a frame grabber. The choice is dependent on the type of hardware in the system. The frame grabber, representing the third step, is an image storage and computation device which stores a given pixel array. The frame grabber can vary in capability from one which simply stores an image to significant computation capability. In the more powerful frame grabbers, thresholding, windowing, and histogram modification calculations can be carried out under computer control. The stored image is then subsequently processed and analyzed by the combination of the frame grabber and the vision controller.

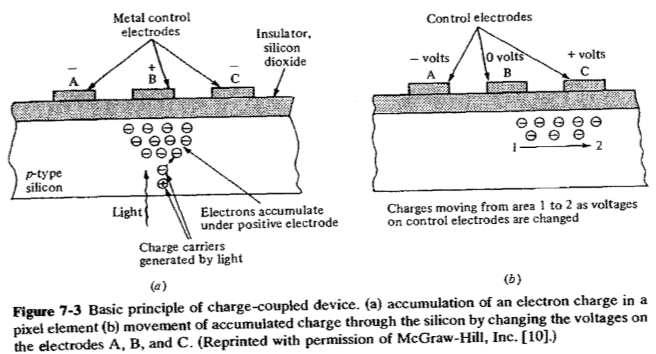
**Imaging Devices**

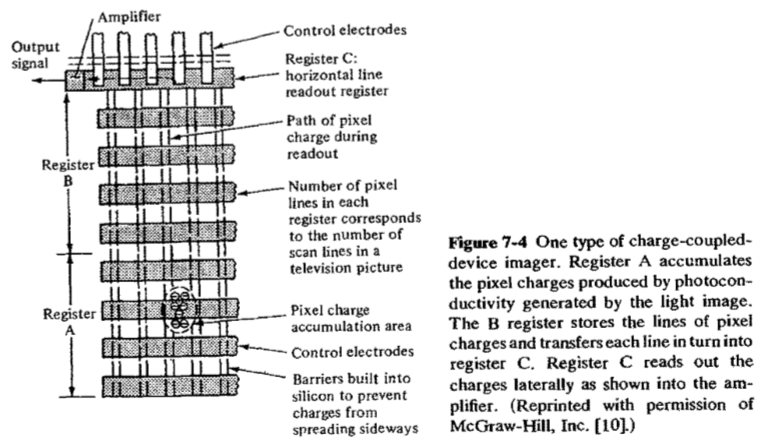
There are a variety of commercial imaging devices available. Camera technologies available include the older black-and-white vidicon camera, and the newer, second-generation, solid state cameras. Solid state cameras used for robot vision include charge-coupled devices (CCD), charge injection devices (CID), and silicon bipolar sensor cameras. For our purposes, we review two such devices in this subsection, the vidicon camera and the charge-coupled device (CCD).

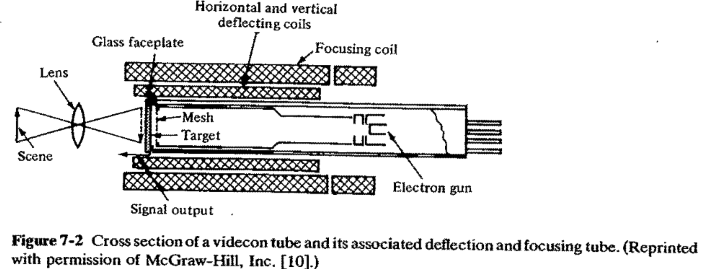
Figure 7-2 illustrates the vidicon camera. In the operation of this system, the lens forms an image on the glass faceplate of the camera. The faceplate has an inner surface which is coated with two layers of material. The first layer consists of a transparent signal electrode film deposited on the faceplate of the inner surface. The second layer is a thin photosensitive material deposited over the conducting film. The photosensitive layer consists of a high density of small areas. These areas are similar to the pixels mentioned previously. Each area generates a decreasing electrical resistance in response to increasing illumination. A charge is created in each small area upon illumination. An electrical charge pattern is thus generated corresponding to the image formed on the faceplate. The charge accumulated for an area is a function of the intensity of impinging light over a specified time.

Once a light sensitive charge is built up, this charge is read out to produce a video signal. This is accomplished by scanning the photosensitive layer by an electron beam. The scanning is controlled by a deflection coil mounted along the length of the tube. For an accumulated positive charge the electron beam deposits enough electrons to neutralize the charge. An equal number of electrons flow to cause a current at the video signal electrode. The magnitude of the signal is proportional to the light intensity and the amount of time with which an area is scanned. The current is then directed through a load resistor which develops a signal voltage which is further amplified and analyzed. Raster scanning eliminates the need to consider the time at each area by making the scan time the same for all areas. Only the intensity of the impinging light is considered. In the United States, the entire faceplate is scanned approximately 30 frames per second. The European standard is 25 frames per second. Raster scanning is typically done by scanning the electron from left to right and top to bottom. The process is such that the system is designed to start the integration with zero accumulated charge. For the fixed scan time the charge accumulated is proportional to the intensity of that portion of the image being considered. The output of the camera is a continuous voltage signal for each line scanned. The voltage signal for each scan line is subsequently sampled and quantized resulting in a series of sampled voltages being stored in digital memory. This analog-to-digital conversion process for the complete screen (horizontal and vertical) results in a two-dimensional array of picture elements (pixels). Typically, a single pixel is quantized to between 6 and 8 bits by the A/D converter.

Another approach to obtaining a digitized image is by use of the charge coupled device (CCD). In this technology, the image is projected by a video camera onto the CCD which detects, stores, and reads out the accumulated charge generated by the light on each portion of the image. Light detection occurs through the absorption of light on a photoconductive substrate (e.g., s1hcon). Charges accumulate under positive control electrodes in isolated wells due to voltages applied to the central electrodes. Each isolated well represents one pixel and can be transferred to output storage registers by varying the voltages on the metal control electrodes. This is illustrated in Figs. 7-3(a) and (b). Figure 7-4 indicates one type of CCD imager. Charges are accumulated for the time it takes to complete a single image after which they are transferred line by line into a storage register. For example, register A in Fig. 7-4 accumulates the pixel charge produced by the light image. Once accumulated for a single picture, the charges are transferred line by line to register 8, The pixel charges are read out line by line through a horizontal register C to an output amplifier. During readout, register A is accumulating new pixel elements. The complete cycle is repeated approximately every th of a second.







**3a.3 IMAGE PROCESSING AND ANALYSIS**

The discussion in the preceding section described how images are obtained, digitized, and stored in a computer. For use of the stored image in industrial applications, the computer must be programmed to operate on the digitally stored image. This is a substantial task considering the large amount of data that must be analyzed. Consider an industrial vision system having a pixel density of 350 pixels per line and 280 lines (a total of 98,000 picture elements), and a 6-bit register for each picture element to represent various gray levels; this would require a total of 98,000 x 6 588,000 bits of data for each -Jos. This is a formidable amount of data to be processed in a short period of time and has led to various techniques to reduce the magnitude of the image processing problem. These techniques include:

1. Image data reduction 2. Segmentation 3. Feature extraction 4. Object recognition

We will discuss these techniques of image data analysis in the following subsections.

**Image Data Reduction**

In image data reduction, the objective is to reduce the volume of data. As a preliminary step in the data analysis, the following two schemes have found common usage for data reduction:

1. Digital conversion 2. Windowing

The function of both schemes is to eliminate the bottleneck that can occur from the large volume of data in image processing. Digital conversion reduces the number of gray levels used by the machine vision system. For example, an 8-bit register used for each pixel would have 28 = 256 gray levels. Depending on the requirements of the application, digital conversion can be used to reduce the number of gray levels by using fewer bits to represent the pixel light intensity. Four bits would reduce the number of gray levels to 16. This kind of conversion would significantly reduce the magnitude of the image-processing problem.

**Segmentation**

Segmentation is a general term which applies to various methods of data reduction. In segmentation, the objective is to group areas of an image having similar characteristics or features into distinct entities representing parts of the image. For example, boundaries (edges) or regions (areas) represent two natural segments of an image. There are many ways to segment an image.

Three important techniques that we will discuss are:

l. Thresholding 2. Region growing 3. Edge detection

In its simplest form, thresholding is a binary conversion technique in which each pixel is converted into a binary value, either black or white. This is accomplished by utilizing a frequency histogram of the image and establishing what intensity (gray level) is to be the border between black and white. This is illustrated for an image of an object as shown in Fig. 7-6. Figure 7-6(a) shows a regular image with each pixel having a specific gray tone out of 256 possible gray levels. The histogram of Fig. 7-6(b) plots the frequency (number of pixels) versus the gray level for the image. For histograms that are bimodal in shape, each peak of the histogram represents either the object itself or the background upon which the object rests. Since we are trying to differentiate between the object and background, the procedure is to establish a threshold (typically between the two peaks) and assign, for example, a binary bit 1 for the object and 0 for the background. The outcome of this thresholding technique is illustrated in the binary-digitized image of Fig. 7-6(c). To improve the ability to differentiate, special lighting techniques must often be applied to generate a high contrast.

It should be pointed out that the above method of using a histogram to determine a threshold is only one of a large number of ways to threshold an image. It is however the method used by many of the commercially available robot vision systems today. Such a method is said to use a global threshold for the entire image. In some cases this is not possible and a local thresholding method as described below may be employed.

When it is not possible to find a single threshold for an entire image (for example, if many different objects occupy the same scene, each having different levels of intensity), one approach is to partition the total image into smaller rectangular areas and determine the threshold for each window being analyzed.

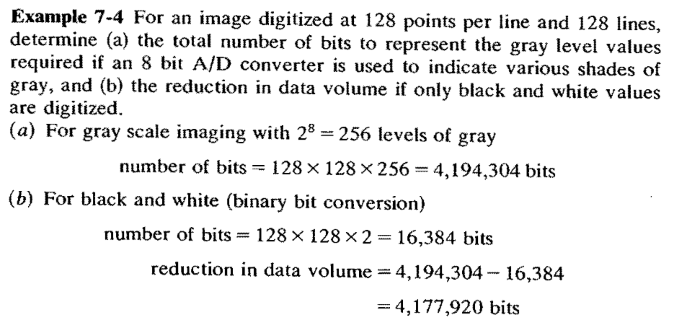
Thresholding is the most widely used technique for segmentation in industrial vision applications. The reasons are that it is fast and easily implemented and that the lighting is usually controllable in an industrial setting. Once thresholding is established for a particular image, the next step is to identify particular areas associated with objects within the image. Such regions usually possess uniform pixel properties computed over the area. The pixel properties may be multidimensional; that is, there may be more than a single attribute that can be used to characterize the pixel (e.g., color and light intensity). We will avoid this complication and confine our discussion to single pixel attributes (light intensity) of a region.

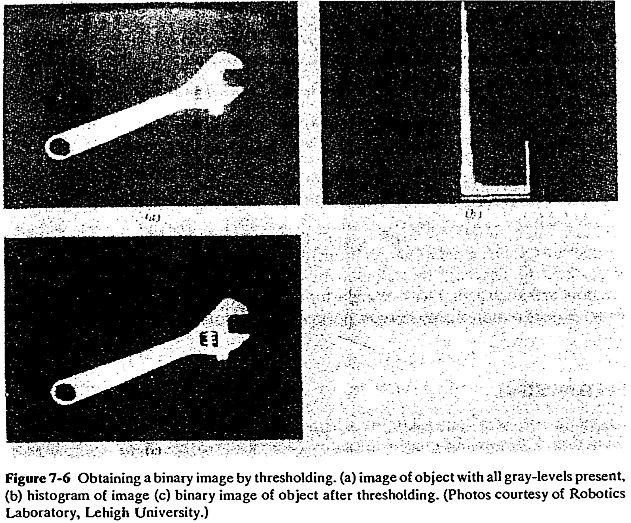
***Region growing*** is a collection of segmentation techniques in which pixels are grouped in regions called grid elements based on attribute similarities. Defined regions can then be examined as to whether they are independent or can be merged to other regions by means of an analysis of the difference in their average properties and spatial connectiveness. For instance, consider an image as depicted in Fig. 7-7(a). To differentiate between the objects and the background, assign l for any grid element occupied by an object and 0 for background elements. It is common practice to use a square sampling grid with pixels spaced equally along each side of the grid. For the two-dimensional image of a key as shown, this would give the pattern indicated in Fig. 7-7(b). This technique of creating "runs" of ls and 0s is often used as a first-pass analysis to partition the image into identifiable segments or "blobs." Note that this simple procedure did not identify the hole in the key of Fig. 7-7(a). This could be resolved by decreasing the distance between grid points and increasing the accuracy with which the original image is represented.

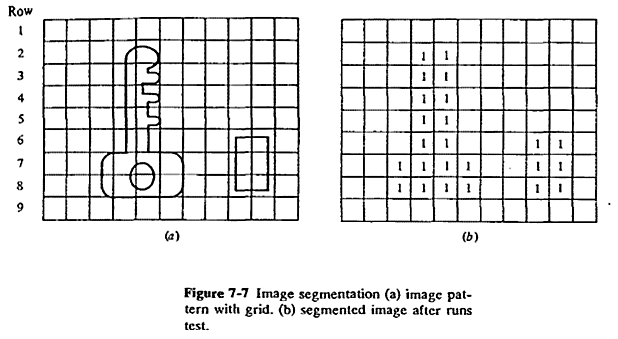
For a simple image such as a dark blob on a light background, a runs technique can provide useful information. For more complex images, this technique may not provide an adequate partition of an image into a set of meaningful regions. Such regions might contain pixels that are connected to each other and have similar attributes, for example, gray level. A typical region-growing technique for complex images could have the following procedure:

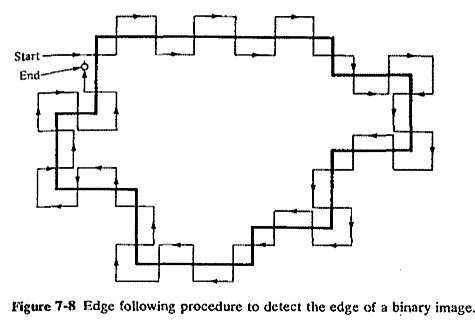
1. Select a pixel that meets a criterion for inclusion in a region. In the simplest case this could mean select white pixel and assign a value of l.
2. Compare the pixel selected with all adjacent pixels. Assign an equivalent value to adjacent pixels if an attribute match occurs.
3. Go to an equivalent adjacent pixel and repeat process until no equivalent pixels can be added to the region.

This simple procedure of "growing" regions around a pixel would be repeated until no new regions can be added for the image. The region growing segmentation technique described here is applicable when images are not distinguishable from each other by straight thresholding or edge detection techniques. This sometimes occurs when lighting of the scene cannot be adequately controlled. In industrial robot vision systems, it is common practice to consider only edge detection or simple thresholding. This is due to the fact that lighting can be a controllable factor in an industrial setting and hardware/computational implementation is simpler. Edge detection considers the intensity change that occurs in the pixels at the boundary or edges of a part. Given that a region of similar attributes has been found but the boundary shape is unknown, the boundary can be determined by a simple edge following procedure. This can be illustrated by the schematic of a binary image as shown in Fig. 7-8. For the binary image, the procedure is to scan the image until a pixel within the region is encountered. For a pixel within the region, turn left and step, otherwise, turn right and step. The procedure is stopped when the boundary is traversed and the path has returned to the starting pixel. The contour-following procedure described can be extended to gray level images.



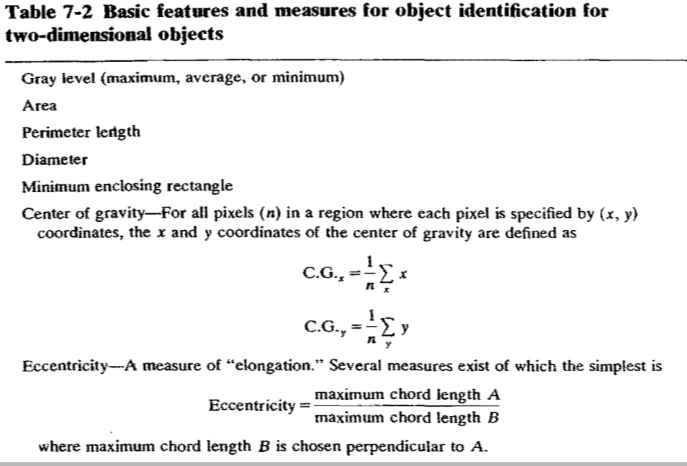


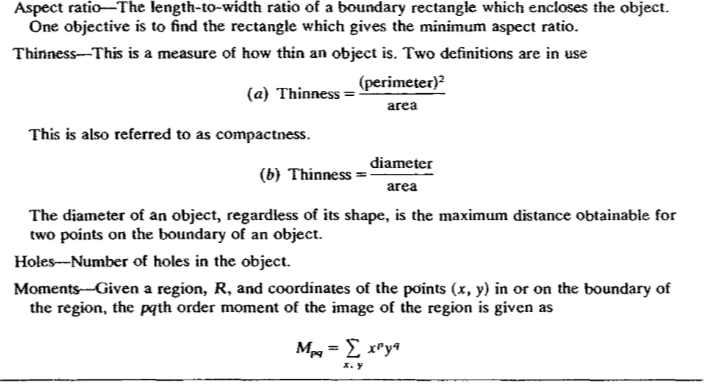


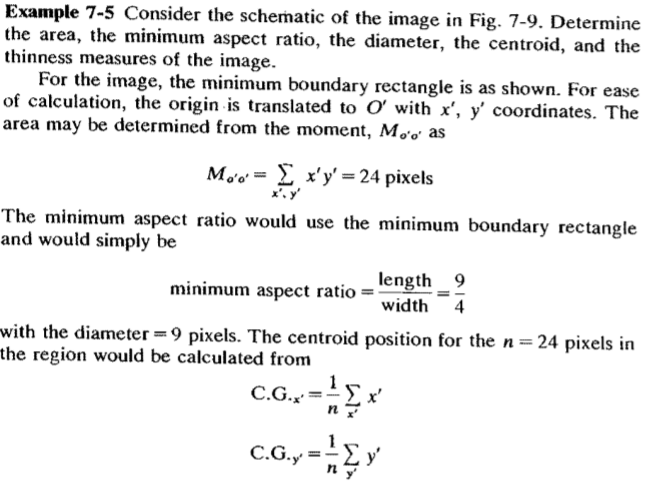


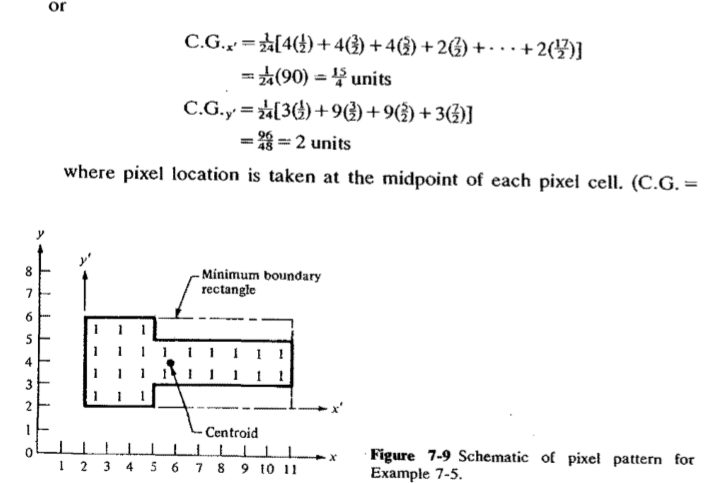
**Feature Extraction**

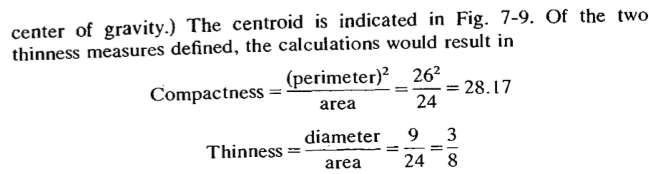
In machine vision applications, it is often necessary to distinguish one object from another. This is usually accomplished by means of features that uniquely characterize the object. Some features of objects that can be used in machine vision include area, diameter, and perimeter. A feature, in the context of vision systems, is a single parameter that permits ease of comparison and identification. A list of some of the features commonly used in vision applications is given in Table 7-2. The techniques available to extract feature values for two-dimensional cases can be roughly categorized as those that deal with boundary features and those that deal with area features. The various features can be used to identify the object or part and determine the part location and/or orientation. The region-growing procedures described before can be used to determine the area of an object's image. The perimeter or boundary that encloses a specific area can be determined by noting the difference in pixel intensity at the boundary and simply counting all the pixels in the segmented region that are adjacent to pixels not in the region; that is, on the other side of the boundary. An important objective in selecting these features is that the features should not depend on position or orientation. The vision system should not be dependent on the object being presented in a known and fixed relationship to the camera. The preceding measures provide some basic methods to analyze images in a two-dimensional plane. Various other measures exist for the three-dimensional case as well. To illustrate some of the two-dimensional definitions and measures, the following example is provided.











**Object Recognition**

The next step in image data processing is to identify the object the image represents. This identification problem is accomplished using the extracted feature information described in the previous subsection. The recognition algorithm must be powerful enough to uniquely identify the object. Object recognition techniques used in industry today may be classified into two major categories:

1. Template-matching techniques
2. Structural techniques

Template-matching techniques are a subset of the more general statistical pattern recognition techniques that serve to classify objects in an image into predetermined categories. The basic problem in template matching is to match the object with a stored pattern feature set defined as a model template. The model template is obtained during the training procedure in which the vision system is programmed for known prototype objects. These techniques are applicable if there is not a requirement for a large number of model templates. The procedure is based on the use of a sufficient number of features to minimize the frequency of errors in the classification process. The features of the object in the image (e.g., its area, diameter, aspect ratio, etc.) are compared to the corresponding stored values. These values constitute the stored template. When a match is found, allowing for certain statistical variations in the comparison process, then the object has been properly classified.

Structural techniques of pattern recognition consider relationships between features or edges of an object. For example, if the image of an object can be subdivided into four straight lines (the lines are called primitives) connected at their end points, and the connected lines are at right angles, then the object is a rectangle. This kind of technique, known as syntactic pattern recognition, is the most widely used structural technique. Structural techniques differ from decision-theoretic techniques in that the latter deals with a pattern on a quantitative basis and ignore for the most part interrelationships among object primitives. A detailed discussion of pattern recognition techniques is the subject of complete books and is beyond the scope of this text.

It can be computationally time consuming for complete pattern recognition. Accordingly, it is often more appropriate to search for simpler regions or edges within an image. These simpler regions can then be used to extract the required features. The majority of commercial robot vision systems make use of this approach to the recognition of two-dimensional objects. The recognition algorithms are used to identify each segmented object in an image and assign it to a classification (e.g., nut, bolt, flange, etc.)

**3a.4 TRAINING THE VISION SYSTEM**

The purpose of vision system training is to program the vision system with known objects. The system stores these objects in the form of extracted feature values which can be subsequently compared against the corresponding feature values from images of unknown objects. Typical features used in machine vision were discussed in the previous section and summarized in Table 7-2.

Training of the vision system should be carried out under conditions as close to operating conditions as possible. Physical parameters such as camera placement, aperture setting, part position, and lighting are the critical conditions that should be simulated as closely as possible during the training session.

Vision system manufacturers have developed application software for each individual system marketed. The software is typically based on a high-level programming language. For example, Object Recognition Systems Inc. uses the "C" language, and Automatix lnc. uses their internally developed language called RAIL (for Robot Automatic lncorporated Language). There are two versions of RAIL, one for automated vision systems and the other for robot programming.

**3a.5 ROBOTIC APPLICATIONS**

Many of the current applications of machine vision are inspection tasks that do not involve the use of an industrial robot. A typical application is where the machine vision system is installed on a high-speed production line to accept or reject parts made on the line. Unacceptable parts are ejected from the line by some mechanical device that is communicating with the vision system. Machine vision applications can be considered to have three levels of difficulty. These levels depend on whether the object to be viewed is controlled in position and/or appearance. Controlling the position of an object in a manufacturing environment usually requires precise fixturing. Controlling the appearance of an object is accomplished by lighting techniques. Also, appearance is influenced by the object's surface texture or coloration. The three levels of difficulty used to categorize machine vision applications in an

* 1. The object can be controlled in both position and appearance.
  2. Either position or appearance of the object can be controlled but not both.
  3. Neither position nor appearance of the object can be controlled.

The third level of difficulty requires advanced vision capabilities. The objective in engineering the vision application is to lower the level of difficulty involved, thereby reducing the level of sophistication of the vision system required in the application. For example, one problem that occurs in object recognition is that the recognition process is facilitated if the object is in a known position and orientation. Parts in a factory are typically not positioned and oriented in this manner. This problem can be reduced from a third level to a first level of difficulty by fixturing the parts and using techniques such as structured lighting to control the appearance. In this section we will emphasize the use of machine vision m robotic applications. Robotic applications of machine vision fall into the three broad categories listed below:

1. Inspection
2. Identification
3. Visual servoing and navigation

The first category is one in which the primary function is the ***inspection*** process. This is carried out by the machine vision system, and the robot is used in a secondary role to support the application. The Objectives of machine vision inspection include checking for gross surface defects, discovery of flaws in labeling (during final inspection of the product package), verification of the presence of components in assembly, measuring for dimensional accuracy, and checking for the presence of holes and other features m a part. When these kinds of inspection operations are performed manually, there is a tendency for human error. Also, the time required in most manual inspection operations requires that the procedures be accomplished on a sampling basis. With machine vision, these procedures are carried out automatically, using 100 percent inspection, and usually in much less time.

The second category, ***identification***, is concerned with applications m which the purpose of the machine vision system is to recognize and classify an object rather than to inspect it. Inspection implies that- the part must be either accepted or rejected. Identification involves a recognation process m which the part itself, or its position and/or orientation, is determined. This is usually followed by a subsequent decision and action taken by the robot. Ident1ficat1on applications of machine vision include part sorting, palletizing and depalletizing, and picking parts that are randomly oriented from a conveyer or bin.

In the third application category, ***visual servoing and navigational control***, the purpose of the vision system is to direct the actions of the robot (and other devices in the robot cell) based on its visual input. The generic example of robot visual servoing is where the machine vision system is used to control the trajectory of the robot's end effector toward an object in the workspace. Industrial examples of this application include part positioning, retrieving parts moving along a conveyor, retrieving and reorienting parts moving along a conveyor, assembly, bin picking, and seam tracking in continuous arc welding. An example of navigational control would be in automatic robot path planning and collision avoidance using visual data. Clearly the visual data are just an important input in this type of task and a great deal of intelligence is required in the controller to use the data for navigation and collision avoidance. This and the visual servoing tasks remain important research topics and are not now viable applications of robot vision systems.

The bin-picking application is an interesting and complex application of machine vision in robotics which involves both identification and servoing. Bin picking involves the use of a robot to grasp and retrieve randomly oriented parts out of a bin or similar container. The application is complex because parts will be overlapping each other. The vision system must first recognize a target part and its orientation in the container, and then it must direct the end effector to a position to permit grasping and pickup. The difficulty is in the fact that the target part is jumbled together with many other parts (probably identical parts), and the conditions of contrast between the target and its surroundings are far from ideal for part recognition.

Solution of the bin-picking problem owes much to the pioneering work in vision research at the University of Rhode Island. At the time of writing, there are two commercially available bin-picking systems, one offered by Object Recognition Systems, Inc. (ORS) called the i-bot l system, and the other by General Electric Co. called BinVision. We will describe the ORS product to illustrate the operation of these systems. Figure 7-10 shows the i-bot system. It utilizes a Unimation PUMA robot to pick parts from the bin. The system is capable of identifying the position and orientation of three parts in a 5-s image-processing cycle. However, the parts must have a relatively large aspect ratio (length-to-width ratio) for the system to operate effectively. Also, the objects in the bin must be the same in terms of features such as size, shape, and texture.

Seam tracking in continuous arc welding is another example of visual servoing and navigation in robotic vision systems. Because the vision system must operate in the presence of an arc-welding torch, special problems in the interpretation of the vision image must be solved. A typical situation uses a form of structured light to project a known pattern on the parts to be welded and uses the observed geometric distortion to determine the path and other parameters required for a successful welding operation.

**3b) ARTIFICIAL INTELLIGENCE**

**3b.1 INTRODUCTION**

Artificial intelligence efforts aim at developing systems that appear to behave intelligently. Often this is described as developing machines that "think." Typically this work is carried out as a branch of computer science, although it also contains elements of psychology, linguistics, and mathematics. In the introduction to The Handbook of Artificial Intelligence the editors state:

“***Artificial intelligence is the part of computer science concerned with the characteristics we associate* with *intelligence in human behavior-understanding language, learning, reasoning, solving problems, and so on***”

Because of the difficulty in defining intelligence, AI research is often best described by discussing its goals, as we do in the next section.

**3b.2 GOALS OF AI RESEARCH**

While the general objective of AI work is the development of machines which exhibit intelligent behavior, this statement is in itself too broad and ambiguous to be meaningful. The following paragraphs will **describe some of the areas of AI which are presently being pursued as distinct areas of research**.

1. **Problem solving**. Examples of problem-solving systems are the chessplaying programs. Given a set of rules and strategies these systems are capable of playing chess on a proficient level. Problem solving does not necessarily restrict itself to games. Consider, in the future, telling your household robot to fetch the paper. On the way to solving the problem of retrieving the paper the robot must identify and solve a number of smaller problems, hopefully in the most efficient sequence. First the robot must identify a possible path to reach the paper, then it must deal with obstacles along that path, such as opening the door. Finally it must deal with grasping the paper and returning it. Almost all tasks with which we are confronted daily involve problem solving. Design of a system requires the breaking down of the problem into successively smaller problems until the solutions begin to become evident. At this point we must "show" robots all of the motions required to perform an assembly task. A robot endowed with a form of "problem solver" may be able to develop the assembly strategy itself. We will discuss this example in greater detail in a later section.

The techniques of problem solving are the same whether developing a robotic assembly strategy solver or some other problem solver.

2. **Natural language**. In spite of the proliferation of computers into department stores, many people are still uneasy or uncomfortable dealing with computers. This is due in part to the need to talk to a computer in its language rather than the user's "natural" language. The problems facing natural language researchers are numerous. The computer must not only be able to understand the meanings of words, but how those meanings may differ in context with other words. The system must also be able to understand the syntax of the language so that the relationships of the words is understood. Imagine the possible meanings for "Time flew out the window" if taken purely from a grammatical standpoint.

3. **Expert systems**. This area of research is concerned with developing systems that appear to behave as human experts in specific fields. Through a dialog with a human operator an expert system can recommend tests to be performed and ask appropriate questions until it arrives at a conclusion. At present, expert systems are being used to configure computer systems, design circuits, and perform medical diagnosis. Some of the issues involved with the design of expert systems include the problems of dealing with vast amounts of data, explaining the system's "reasoning" on the way to reaching a conclusion, representing the data collected from the human experts, and improving the "knowledge base" with experience.

4. **Learning**. One of the attributes of intelligence is the ability to learn from experience. If machines were capable of learning then the task of endowing them with knowledge, as in the case of expert systems, would be greatly simplified. Some systems have been developed which have shown the ability to learn from experience, but to this date limited progress has been made.

5. **Vision**. Most of the basic concepts employed in commercial vision systems are the result of AI research. One of the more interesting goals of AI vision research is to permit the systems to perform scene analysis. That is, present the vision system with a scene and allow the system to identify objects within the scene. Some of the other areas of AI research include: automatic programming, hardware development, and deductive reasoning.

**3b.3 AI TECHNIQUES**

AI is concerned with the use of data or knowledge. Therefore techniques must be developed for two basic tasks: data representation and data manipulation. In this section we will look at some of the approaches for representing data and using that data in some specific manner. We will not discuss actual programs for doing this, only the general techniques which might be employed by AI programs.

**Knowledge Representation**

When we discuss knowledge representation we are not concerned with the physical operation of the computer, that is, we are not discussing the storage of words as a series of ls and 0s. Rather we are discussing the relationships of facts with respect to each othe1, for example, the statement, "Some birds have wings." Before discussing the various representations of knowledge we must first describe the various types of knowledge which may require representation.

1. **Objects**. More specifically facts about objects, such as "robotics students drink heavily" or "birds have wings."

2. **Events**. Not only the event itself, such as "The robotics student broke his arm," but perhaps the time or cause-effect relation of the event. "The robotics student broke his arm yesterday and the nasty instructor made him pay for it."

3. **Performance**. If the AI system is one which is designed to control a robot then it must have data on the performance of the arm, that is, its kinematics, dynamics, what bits to manipulate in the hardware, and so on.

4. **Metaknowledge**. This is the knowledge about our knowledge. This includes our knowledge of the origin of the information, its relative importance, its rehab1hty, and so on. For example, one would give little weight to the following information "The study of robotics is painless" if that information came from a history student.

At this point let us look at some of the various techniques for representing knowledge.

1.  **Logic**: Formal logic was developed by mathematicians and philosophers to provide a method for making inferences from facts. Formal logic allows facts to be represented in a specific syntax, and, by applying defined rules of inference to these facts, allows conclusions to be drawn. We are probably familiar with the following type of example: Given the two statements

a. All roboticists play games.

b. Jack is a roboticist.

We can conclude by using a rule of inference that

c. Jack plays games.

Conceivably, if a computer were endowed with all of the possible facts about a subject, and also all of the applicable rules of inference, then it should be able to develop any new facts about the subject that may be inferred from the original set. We will discuss later in the chapter the concept of "combinatorial explosion" which may limit the usefulness of logic-based (or any other) representation schemes. As the number of facts becomes larger, the number of combinations of facts with the rules of inference which may apply also increases. This results m generating a problem too large for the computer to solve m a reasonable amount of time.

2. **Procedural representations**: In the previous discussion we examined a method for storing facts. It is also necessary to store information on how to use facts. In a logic-based system as above, every rule of inference must be applied to prove a new point. If it were possible to encode the information along with a way to use that information, it is possible that a more efficient system could be developed. For instance, if we wished to find out whether or not Jack plays games, the database of the computer could contain the following facts:

a. Jack is a roboticist. b. All roboticists play games.

Additionally the system could have stored the following procedure about Jack

*If need-prove plays (x)*

*show is-roboticist (x)*

which states that if we need to prove some fact about **x**, we can do so by showing that **x** is a roboticist. With this information the system would be able to avoid any other information about Jack and jump instantly to the correct solution. This technique of storing information as a procedure is commonly used in computer programs in if-then statements. It offers the advantage of directing the system to find the solution more directly. On the other hand, It takes away from the generality of the system and makes changes difficult.

3. **Semantic networks**: Semantic networks are representations of information which consist of nodes and arcs. Nodes typically represent objects, concepts, or situations and the arcs represent the relationships between nodes. The nodes and links are labeled with simple language descriptions. Figure 10-1 illustrates the following information as a semantic network:

*Jack is a student.*

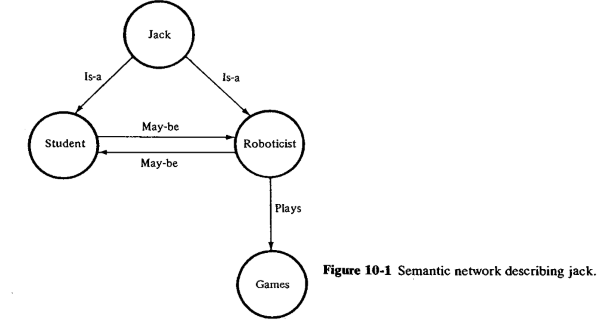
*Jack is a roboticist.*

*Roboticists play games.*

*Roboticists may be students.*

*Students may be roboticists.*

Many inferences may be drawn about Jack, students, and roboticists by investigating the network. Networks are used extensively in AI research as a means of knowledge representation. Unfortunately they also have their drawbacks. They may be too simplistic; how does a network deal with ideas or large amounts of knowledge? As with all representation techniques it cannot be pushed to an extreme.



4.  **Production systems**: Production systems store information in the form of items called "productions." Productions all have the form of IF some expression is true THEN some action. For example, the information we have about Jack and roboticists would be presented as:

a. IF the person is a roboticist THEN he/she plays games.

b. IF the person is reading this book THEN he/she is a roboticist.

If we were able to catch Jack reading this book then we would know that he is, in fact, a game player. Production systems are used to break the problem down into successively more manageable tasks. They are typical of the construction of "expert systems." They provide for uniformly designed systems using the IF-THEN construction as well as a modularity, in that each rule has no direct effect on other rules. Unfortunately, among other problems, these systems may become very inefficient as they become larger.

5. **Frames:** Another representation technique is the use of frames. Frames can be considered as predefined structures with empty slots which are filled to represent the specific facts. For example, the general frame for a student might look like the following:

STUDENT Frame

Height: in inches

Weight: in pounds

Studies: Robotics or History

and for Jack the frame would look like

JACK Frame

Height: 70 inches

Weight: 150 pounds

Studies: Robotics

The slots could be filled by references to other frames or by procedures defining how the slots are to be filled.

**6. Other representation techniques**: In many cases choosing the correct representation technique can greatly simplify the problem. As an example let us investigate a video view of an object. One way to represent the object would be as a semantic net with each pixel representing a node and each arc representing the "is-connected-to" property attached to each adjoining node. A simpler approach is to represent the picture as an array of values corresponding to the brightness of each individual pixel. The point is that the choice of a representation technique should be made only after giving consideration to the type of problem being solved. The above mentioned techniques are applied to different circumstances and in some cases different representation schemes should be found.

**Problem Representation and Problem Solving**

Before discussing problem-solving techniques it is useful to explain what is meant by "problem solving." Problem solving is the task of reaching some specified goal. Examples of these goals may be:

Finding the proof to a mathematical theorem.

Solving a puzzle, such as Rubik's cube.

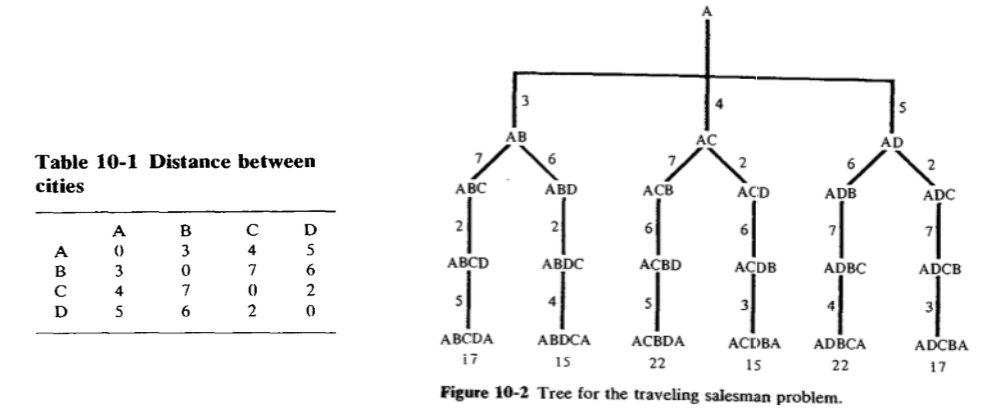
Determining a sequence of assembly steps.

Choosing the next move in a chess game.

In order to solve these problems it is necessary to represent the problem in some way that is amenable to discussion and solution. Two possible schemes for problem representation are:

1. The state-space representation 2. The problem-reduction representation

**1. State-space representation:** In this method we can visualize the problem as all of the possible states found in developing the solution configured as a tree. The tree is made of nodes, which represent the states of the system after certain actions have been taken. The actions are represented by the arcs that connect the nodes. This can be illustrated by using the "traveling salesman" problem. A traveling salesman has to travel to four cities A, B, C, and D. The salesman wishes to travel to all four cities using the shortest possible path and by going to each city only once. He wishes to begin and end his trip at city A The distances between the cities is given in Table 10-1. The state-space representation of the trip is shown in Fig. 10-2. By looking at the state-space representation, we can see that the problem becomes one of choosing the branch of the tree with the shortest sum of arc lengths. There are two paths which provide a solution of 15: ABDCA and ACOBA. Later in this section we will see how to search the state-space tree for this solution. This form of problem solving is called "forward reasoning," because we worked our way forward through all of the states until we found the solution.



2. Problem-reduction representation. In this case we see an example of backward reasoning. In the problem-reduction representation we present the goal as the primary data item and then reduce the problem until we have a set of primitive problems; that is, simpler problems for which we have the data available. This simplification may involve breaking the problem down into a set or sets of smaller problems which must all be solved or into alternative problems, any one of which may be solved. The scheme is graphically represented as an "and-or" graph. In an and-or graph arcs which are connected by a horizontal bar are "anded" and arcs which are not connected are "ored." Figure 10-3 illustrates a simple and-or graph which states:

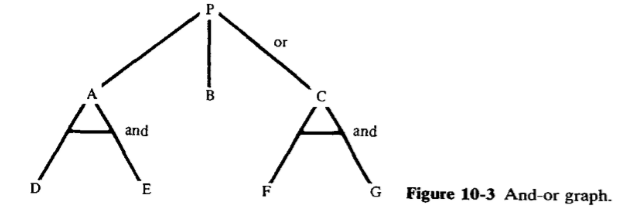
a. P may be solved by solving A or B or C.

b. A may be solved by solving D and E.

c. B may be solved directly (B is a primitive).

d. C may be solved by solving F and G. e. D, E, F, and G are primitives.

In this case, then, the simplest scheme for solving A is to solve B. In the next subsection we discuss the different techniques for searching for the solution.



**Search Techniques in Problem Solving**

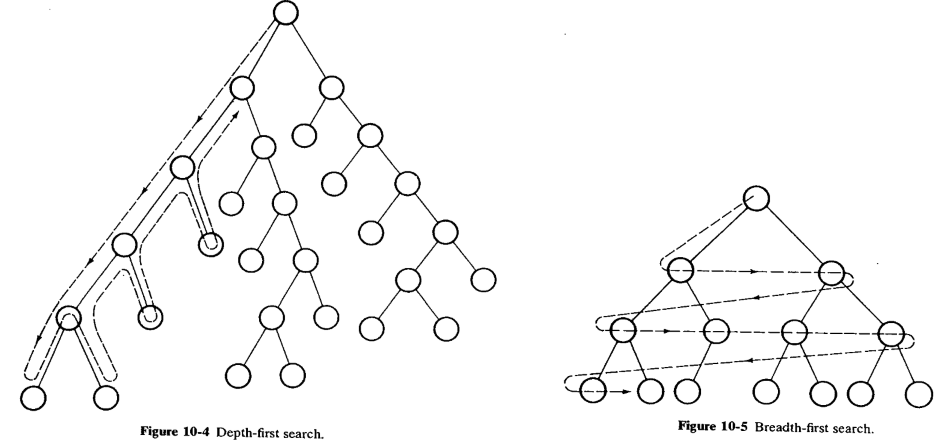
Up to this point we have considered techniques for representing knowledge and problems. If we employ the proper representation technique, then it is often only a matter of searching for a solution to reach the desired goal. In many cases it is possible that the necessary search is too large to be successful in a reasonable amount of time. For example, consider the game of chess. Given that there is a known starting state and that the goal is to capture the opponent's king, it is possible to develop all of the possible move combinations in a state-space representation and then search for the best winning solution and follow that path. Unfortunately the number of move combinations in a game of chess may be on the order of 10120. Because of this type of "combinatorial explosion," techniques that seek to minimize the search effort have been developed. This does not mean that search is the only available technique. Just as the method of knowledge and problem representation must be carefully considered for each task, so should the solution technique. In any case some of the developed search techniques follow:

**1. Depth-first search**: As the name implies, this search technique searches as deeply as possible in the tree network in an effort to find a solution. Figure 10-4 illustrates this principle. The objective is to find the node labeled "S" (solution). Using the depth-first search technique, the system searches through the network as deeply as possible until it finds a terminal node. If that node is not the desired solution, then it backtracks until it finds the first available branch point and goes down the next branch. In the example illustrated the system will eventually find the solution node. In some systems it is possible that the depth of the tree or certain branches of the tree is so deep that the solution is never found.

**2. Breadth-first search:** Figure 10-5 illustrates the concept of breadth-first search. The system evaluates all nodes at the same level in the tree before moving on to evaluate the next level. This system is more conservative than depth-first searches and is not as easily trapped. In cases where the tree is relatively deep on all branches it may be less efficient than a depth-first search.

**3. Hill climbing**: Hill-climbing techniques provide a variation on depth first searches. Rather than moving in an arbitrary decision at each branch point, the hill-climbing algorithm attempts to make the best choice among the possible branches from the present node. This choice is based upon some selection technique. For example, in the traveling salesman problem, the next node chosen may be based on the total distance up to the next possible node. The risk here is that we are only looking at the next node, that is, at local information. While we may be making the best local choices, we may be missmg a far better overall solution that had one bad arc.

**4. Best-first search:** This is a variation on hill climbing. In this case, rather than choosing the best next branch from a node, the system selects the best next node regardless of its position in the system. This generally provides the optimum solution, but does not guarantee it.



**5 Branch and bound:** This is similar to the best-first search. The system evaluates all of the partial paths to the first level and expands the most promising path to the next level(s). As the path ceases to be the optimum (it becomes too long) the system expands the new most promising path. In this way it is always investigating the optimum path to the deepest level necessary. In order to ensure that the optimum path is found all partial paths must be extended until they become longer than the solution path, or until a shorter solution is found.

**6. Constraints:** In some cases it is not necessary to consider all of the possible options in developing the tree. Very often in the real would, constraints are placed upon information by its context. We know, for instance, that mice and cats are not likely to live in the same cage. Applying these constraints as we are traveling through the tree may lead to a significant reduction in the possible number of nodes to be explored.

There are many other search techniques beyond those explained here but they are beyond the scope of this text. At this point it is worthwhile to review what has been covered so far.

Knowledge can be represented in a number of ways. Not only are the data important, but also the relation of data to other information and the rules for manipulating data are important. Problems may also be represented m a number of ways. Problems may be solved by forward or backward reasoning. By representing solutions as trees, optimal solutions may be found using search techniques.

**Module -4**

**Robot cell design and control, Material Transfer, Machine Loading/Unloading:**

Robot cell layouts, multiple robots and machine interference, other considerations in work -cell design, work-cell control, interlocks, error detection and recovery, work -cell controller, robot cycle time analysis.

**Material Transfer, Machine Loading/Unloading:** General considerations in robot material handling, material transfer applications, machine loading and unloading. [Textbook-1]

**4a.1 ROBOT CELL LAYOUTS**

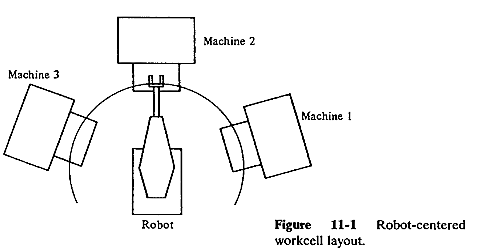
Robot workcells can be organized into various arrangements or layouts. These layouts can be classified into three basic types:

1. Robot-centered cell 2. In-line robot cell 3. Mobile robot cell

The following subsections describe these workcell configurations.

**4a.1.1 Robot-Centered Workcell**

In the robot-centered cell, illustrated in Fig. 11-1, the robot is located at the approximate center of the cell and the equipment is arranged in a partial circle around it. The most elementary case is where one robot performs a single operation, either servicing a single production machine, or performing a single production operation. Initial installations of industrial robots in the 1960s were illustrative of this case. Die casting, one of the very first applications for a robot, required the robot to unload the part from the die after each casting cycle and dip it into a quenching bath. Other production machine applications required the robot to both load and unload the workpart. For some of these applications, the cycle times of the machine were relatively long compared to the part-handling time of the robot. Metal-machining operations are examples of this imbalance condition. This required the robot to be idle for a high proportion of the cycle, causing low utilization of the robot. To increase the robot utilization, the workcell concept was developed in which one robot serviced several machines as pictured in Fig. 11-1. An application of the robot-centered cell in which the robot performs the process is arc welding. In this case, the robot accomplishes the production operation itself, rather than servicing a production machine tool. With these robot-centered cell arrangements, a method for delivering the workparts into and/or out of the cell must be provided. Conveyors, parts feeders with delivery chutes, and pallets are the means for accomplishing this function. Machining, die casting, plastic molding, and other similar production operations for discrete part production are examples of this case. These devices are used to present the parts to the robot in a known location and orientation for proper pickup. In arc welding, human operators are often used to accomplish the parts loading and unloading function for the robot.



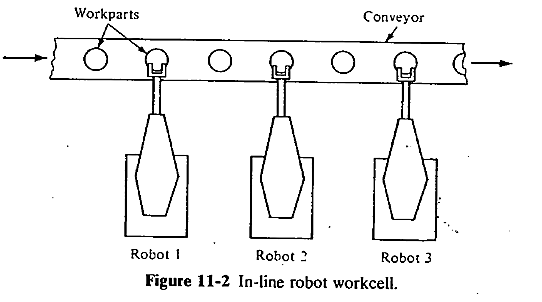
**4a.1.2 In-Line Robot Cell**

With the in-line cell arrangement, pictured in Fig. 11-2, the robot is located along a moving conveyor or other handling system and performs a task on the product as it travels past on the conveyor. Many of the in-line cell layouts involve more than a single robot placed along the moving line. A common example of this cell type is found in car body assembly plants in the automobile industry. Robots are positioned along the assembly line to spot weld the car body frames and panels. The three categories of transfer systems that can be used with the in-line cell configuration 3 are:

1. Intermittent transfer 2. Continuous transfer 3. Nonsynchronous transfer

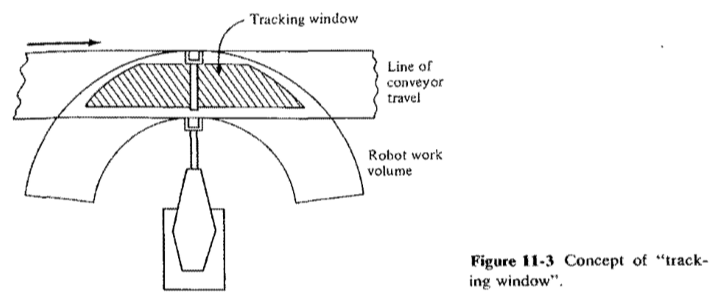
An intermittent transfer system moves the parts with a start-and-stop motion from one workstation along the line to the next. It is sometimes called a synchronous transfer system because all of the parts are moved simultaneously and then registered at their next respective stations. In a robot layout using intermittent transfer, the robot is in a stationary location and constitutes one position along the line at which a part or product stops for processing. The advantage possessed by the intermittent transfer system in robot applications is that the part can be registered in a fixed location and orientation with respect to the robot during the robot's work cycle. This registration of the part relative to the robot becomes a problem when the continuous transfer system is used to move parts in the cell. With this type of transfer system, the workparts are moved continuously along the line at constant speed. This means that the position and orientation of the part is continuously changing with respect to any fixed location along the line. The problem can be solved by using either of two means2 :

1. A moving baseline tracking system
2. A stationary baseline tracking system



The moving baseline tracking system involves the use of some sort of transport system to move the robot along a path parallel to the line of travel of the workpart while the operation is performed on the part. In this way, the relative position of the part and the robot remain constant during the work cycle. The problem with this arrangement is that an additional degree of freedom must be provided for the robot to move along the conveyor. This additional degree of freedom is usually accomplished by mounting the robot on a cart which can be moved along a track or rail parallel to the conveyor. This solution involves considerable capital expense to construct the system to maintain accurate registration between the robot and the part. One of the operational problems that must be taken into account in the design of a production line with several robots is the potential interference and collision problem between robots at adjacent stations along the line. The easiest way to solve the problem is to space the robots sufficiently to avoid the possibility of interference. However, this requires a significant amount of floor space. An alternative solution is to provide the workcell with enough intelligence that it knows where each robot is at any moment, and can control the sequence so as to avoid collisions.

In the stationary baseline tracking system, the robot is located in a stationary position along the line but its manipulator is capable of tracking the moving workpart. "Tracking" in this context means that the robot is able to maintain the positions of the programmed points, including the orientation of the end effector and the motion velocities, in relation to the workpart even though the part is moving along a conveyor. The engineering problems that must be solved to implement a stationary baseline tracking system are considerable although different from those encountered in the moving baseline system. First, the robot must have sufficient computational and control capability to accomplish tracking. This requires that the regular motion pattern of the manipulator be continuously translated in space in a direction parallel to the conveyor and at a speed equal to that of the conveyor. This allows the relative positions of the end effector and the part to be maintained during the cycle. A second problem, related to the first is concerned with the robot's "tracking window." The tracking window can be thought of as the intersection of the robot's work volume with the line of travel of the workpart along the conveyor. This concept is illustrated in Fig. 11-3. Allowances must be factored into this definition to account for manipulation of the wrist an end effector and for access to portions of the workpart that might be difficult to reach. For a robot with tracking capability, the total motion cycle in a particular application must be consistent with the tracking window for that application. A third problem involves the sensing of the part on the conveyor.



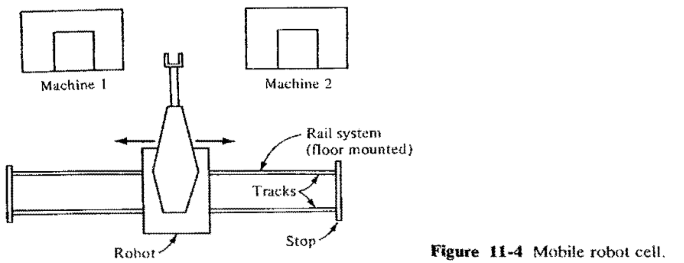
For cells in which different product models will be processed, a sensor system must be used to identify which model is being delivered to the robot. A sensor is also required to determine that the part has entered the tracking window and that the robot can commence its work cycle. Other sensors are needed to track the position and velocity of the part during the cycle so as to coordinate with the robot tracking system. It is risky to presume that there will be no variations in the location and speed of the part as it is being processed.

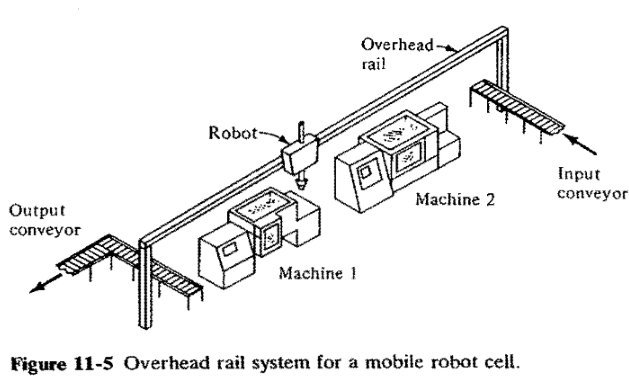
The third type of transport system is nonsynchronous transfer. It is also referred to by the name "power-and-free" system. In this materials-handling system, each part moves independently along the conveyor in a stop-and-go fashion. When a particular workstation has completed its processing of a part, that part proceeds to move toward the next workstation in the line. Hence, at any given moment, some workparts are being processed while others are located between stations. The design and operation of this type of transfer system is more complicated than the other two because each part must be provided with its own independently operated, moving cart. However, the problem of designing and controlling the robot system used in conjunction with the power-and-free method is less complicated than for the continuous transfer method. For the irregular timing of arrivals on the nonsynchronous transfer system, sensors must be provided to indicate to the robot when to begin its work cycle. The more complex problems of registration between the robot and the part that must be solved in the continuously moving conveyor systems are not encountered on either the intermittent transfer or the nonsynchronous transfer.

**4a.1.3 Mobile Robot Cells**

The third category of robot cell design is one in which the robot is capable of moving to the various pieces of equipment within the cell. This is typically accomplished by mounting the robot on a mobile base which can be transported on a rail system. The rail systems used in robot cells are either tracks fastened to the floor of the plant or overhead rail systems. Figure 11-4 illustrates the concept of the track-on-floor system, while the overhead rail system is shown in Fig. 11-5. The advantage of the overhead rail system compared to the floor-mounted track system is that less floor space is required. The disadvantage is the increased cost of constructing the overhead system.

A mobile robot cell would be appropriate when the robot is servicing several machine tools with long processing cycles. In this situation, the robot would be able to share its time among the machines without significant idle time for either itself or the machines it is servicing. If a separate robot were to service each of the machines, the utilization of the robots would be low because most of their time would be spent waiting for the machine cycles to complete. Accordingly, one of the problems in the design of a mobile robot cell is to find the optimum number of machines for the robot to service. The objective in this problem is to maximize the number of machines in the cell without causing idle time on any of the machines.





**4a.2 MULTIPLE ROBOTS AND MACHINE INTERFERENCE**

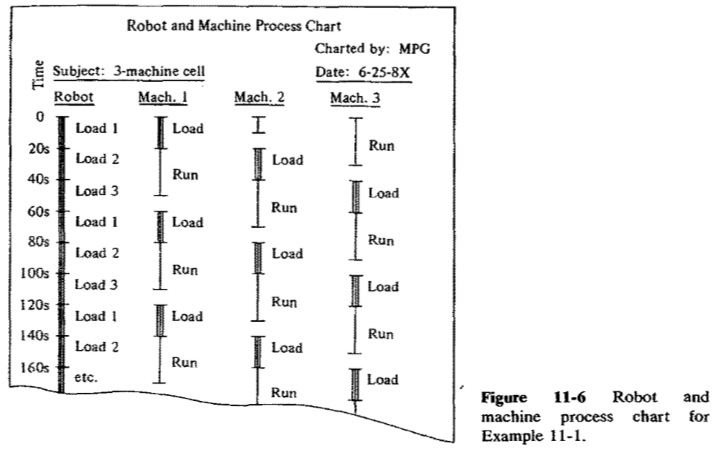
In some robot cells, there will be more than one robot required to perform the application. The in-line robot cell is a common example of this situation. In other cases, one robot will work with more than one machine in the cell. Either the robot-centered cell or the mobile robot cells are illustrative of this possibility. In either of these situations, care must be taken to ensure that the different pieces of equipment do not interfere with one another. There are two ways in which this interference can occur.

The first case involves physical interference of the robots, where the work volumes of two robots in the cell overlap each other. In this situation, the danger of collision exists between the robot arms. This is most easily prevented by separating the robots by an adequate distance to avoid the problem. However, there are some applications in which it is desirable for two robots to share the same space. An example would be where one robot places a workpart at a certain location, and the second robot picks the part up. The location must be in the work envelopes of both robots. Accordingly, an alternative approach is to coordinate the programmed motion cycles of the two robots so that the arms are never close enough to risk a collision.

The second type of interference is when there are two or more machines being serviced by one robot, and the machine cycles are timed in such a way that idle time is experienced by one or more machines while another machine is being serviced by the robot. This is called machine interference and it is a common problem encountered when a human worker is assigned to service multiple machines. The difference between machine interference with a human worker and machine interference with a robot is that the amount of interference in the human cell is affected by variations in worker cycle times and by the worker's level of effort. With greater variation in the cycle time and a lower effort level, the machine interference will tend to be greater. In a robot cell, the robot's cycle time will not be affected by effort level and the amount of cycle time variation will be significantly less than for human work.

Machine interference can be measured as the total idle time of all the machines in the cell as compared to the operator (or robot) cycle time. The measure is most commonly expressed as a percent. To illustrate the problem, we will use an example of a three-machine cell in which a robot is used to load and unload the machines.

**Example 11-1** Each of the three machines in the cell are identical and they have identical cycle times of 50 s. This cycle time is divided between run time (30 s) and service time (load/unload) by the robot (20 s). The organization of the cycle time is shown in the robot and machine process chart of Fig. 11-6. It can be seen that each machine has idle time during its cycle of 10 s while the robot is fully occupied throughout its work cycle. Total machine idle time of all three machines is 3 x 10 = 30 s and the cycle time of the robot is 3 X 20 = 60 s. Accordingly, the machine interference is 30 s/60 s = 50%.



In this example the cycles of the three machines are the same. In this case, the question of whether or not machine interference will occur is determined by the relative values of machine cycle time and robot cycle time. The machine cycle time is the sum of service time and run time. The robot cycle time is equal to the number of machines multiplied by the service time. If the robot cycle time is greater than the machine cycle time, there will be resulting machine interference if the machine cycle time is greater than the robot cycle time, there will be no machine interference, but the robot will be idle for part of the cycle.

In the case where the service and run times of the machines are different the above relationships become complicated by the problem of determining the best sequence of servicing times for the machines into the robot cycle time.

**4a.3 OTHER CONSIDERATIONS IN WORKCELL DESIGN**

There are several other issues that must be considered in the design of the workcell. Among these considerations are the following:

1. **Changes to other equipment in the cell**. To implement the workcell and interface the robot to the other equipment in the cell, alterations will often have to be made to the equipment. Special fixtures and control devices must be devised to permit the cell to operate as a single, integrated mechanism. Examples of these fixtures and controls include work-holding nests and conveyor stops to position and orient the parts for the robot, changes in the machines to allow the robot arm to gain access to the equipment, and limit switches and other devices to interface the various components in the cell.
2. **Part position and orientation.** For raw workparts being delivered into the .cell, it is important that the robot have a precise pickup location to get the parts from the conveyor or other work-handling system. At this pickup point, the parts must be in a known orientation to enable the robot to grasp and hold it consistently and accurately. During subsequent processing within the cell, this part orientation should not be lost. A method for achieving these objectives of part positioning and orientation must be designed into the workcell.
3. **Part identification problem.** In cells where more than one type of part is processed or assembled, a method of identifying the particular part type must be determined. This can be done by any of a number of automated means, involving optical techniques or limit switches to sense differences in size or part geometry.
4. **Protection of the robot from its environment.** In certain types of applicaltions (e.g., spray painting, hot metal-working operations), a means of protecting the robot from the adverse effects of its environment must be provided.
5. **Utilities.** Providing the necessary utilities (e.g., electricity, air pressure, gas for furnaces, etc.) must be included among the factors considered in the design of the workcell layout.
6. **Control of the workcell.** The activities of the robot must be coordinated with those of the other equipment in the cell. This subject is referred to by the term workcell control, and we devote several sections to it and its related topics in this chapter.
7. **Safety**. A means of protecting human personnel from harm in and around the robot workcell must be provided. This is generally accomplished by means of fences or other barriers, and by designing a safety monitoring system to interrupt the cell operation if unsafe conditions are encountered.

**4a.4 WORKCELL CONTROL**

In addition to the problem of designing the physical layout of the robot cell, another problem is concerned with coordinating the various activities that occur in the cell. Most of these activities occur sequentially, but simultaneous activities can also occur. There are other factors such as the safety of human personnel which must also be considered. Coordination of these various activities is accomplished by a device called the workcell controller or workstation controller. The functions performed by the workcell controller can be divided into three categories as suggested by Thomas:

1. Sequence control 2. Operator interface 3. Safety monitoring

These functions are accomplished either by the robot controller itself or by a higher-level control device, such as a programmable controller. The robot controller usually has a limited input/output capability to permit interfacing with other pieces of equipment in the cell. If the control requirements to operate the cell become at all complicated, then a higher-level controller is needed. We shall discuss the use of programmable controllers in a subsequent section of this chapter. Also, the implementation of an effective workcell controller is dependent largely on the programming capabilities of the robots used.

**Sequence Control**

Sequence control is the primary function of the workstation controller during regular automatic operation of the workcell. It includes the following kinds of control functions:

* Control of the sequence of activities in the workcell
* Control of simultaneous activities
* Making decisions to proceed with the work cycle based on events that occur in the cell
* Making decisions to stop or delay the work cycle based on events that occur in the cell

**Operator Interface**

The purpose of the operator interface in workstation control is to provide a means for human operators to interact with the operation of the cell. There are several situations where this would be required. Among the most important cases are the following:

1. The human is an integral part of the workcell.
2. Emergency stop conditions.
3. Program editing or data input by operator.

The first situation is where the human worker plays an integral role in the operation of the cell. The human performs a portion of the work cycle and the robot performs a portion of the cycle. In this situation, there is a need to allow for variations in the time required by the operator to perform the manual part of the cycle. In a robotic cell, this can be easily accomplished by means of stop/start controls placed conveniently for the operator. The operator uses these controls to regulate the robot cycle as required by the situation!

**Safety Monitoring**

In addition to the operator's ability to override the regular work cycle in the event of an observed safety hazard, the workcell controller should also be capable of monitoring its own operation for unsafe or potentially unsafe conditions in the cell. This function is called safety monitoring or hazard monitoring.

**4a.5 INTERLOCKS**

An interlock in robotic workcell design is a method of preventing the work cycle sequence from continuing unless a certain condition or set of conditions are satisfied. It is a feature of workcell control which plays an important role in regulating the sequence in which the various elements of the cycle are carried out. Referring back to Examples 11-2 and 11-3, interlocks would be used for the following purposes:

* To make sure that a raw workpart was at the pickup location on the conveyor before the robot tried to grasp the part
* To determine when the machining cycle was completed before the robot attempts to load the part into the fixture
* To indicate that the part has been successfully loaded so that the automatic machining cycle can begin

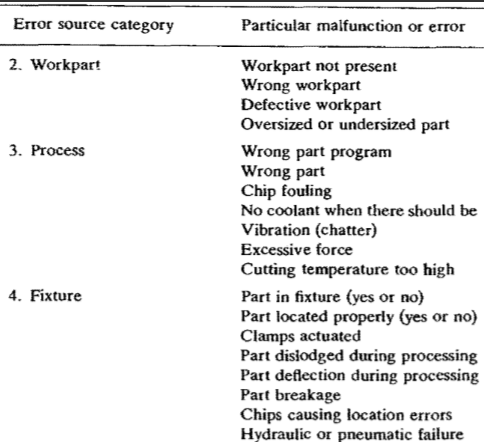
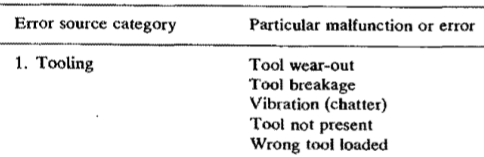
In each of these instances, it is critical that one element of the cycle has been completed before any attempt is made to begin the next element. The method of regulating the sequence of the elements would involve the use of interlocks. Interlocks can be divided into two basic categories: output interlocks and input interlocks. An output interlock involves the use of a signal sent from the workstation controller to one of the machines or other devices in the workcell. It corresponds to the SIGNAL programming statement. For example, an output interlock would be used to signal the machining center in Examples 11-2 and 11-3 to commence the automatic cycle. The output signal originates from the workcell controller and is contingent upon certain conditions being satisfied. In our example, the conditions would be that the workpart has been properly loaded and the robot gripper has been removed to a safe distance. These conditions are usually determined by means of input interlocks. An input interlock makes use of a signal sent from one of the components in the cell to the workstation controller. It corresponds to the WAIT command.

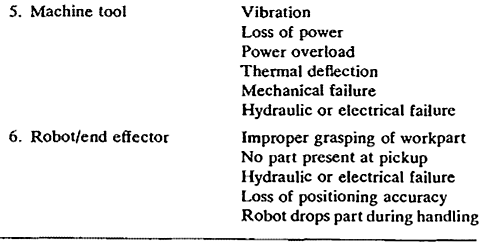
It is employed to indicate that a certain condition or set of conditions have been met and that the programmed work cycle sequence can continue. As an illustration, an input interlock would be used in a machine-loading application to signal the workstation controller that the part has been properly loaded into the fixture on the machine tool table. This condition might readily be sensed by means of a simple limit switch mounted on the fixture which indicates that the part is in place. Interlocks are essential in nearly all robotic workcells consisting of several operating pieces of equipment that must all work in a coordinated fashion. They serve to interface the various components of the workstation. Their use provides a synchronization and pacing of the activities in the cell which could not be accomplished through timing alone in the work cycle. Interlocks allow for variations in the times taken for certain elements in the cycle. They prevent work elements from starting before they should start. And they help to prevent damage of the various components of the cell. In the design of the workcell, consideration must be given not only to the regular sequence of events that will occur during normal operation of the cell, but also to the possible irregularities and malfunctions that might happen. In the regular cycle, the various sequential and simultaneous activities must be identified, together with the conditions that must be satisfied in order for each activity to successfully take place. For the potential malfunctions, the applications engineer must determine a method of identifying that the malfunction has occurred and what action must be taken to respond to the malfunction. Then, for both the regular and irregular events in the cycle, interlocks must be provided to accomplish the required sequence control and hazard monitoring that must occur during the work cycle. In some cases, the interlock signals can be generated by the electronic controllers for the machines and other devices used in the workcell. For example, numerically controlled machine tools would be capable of being interfaced to the workcell controller to signal completion of the automatic machining cycle. In other cases, the applications engineer must design the interlocks using sensors to generate the required signals. Interlocks are often implemented by means of limit switches and other simple devices that serve as sensors. In some cases, the robot cell must make use of more advanced sensors in order to successfully perform the work cycle. Examples of the latter cases would include seam tracking in robotic arc welding, identification of part position and orientation on a moving conveyor, and determining that a component had been properly assembled before proceeding with further assembly work on the product.

**4a.6 ERROR DETECTION AND RECOVERY**

Execution of the work cycle is expected to be repeated over and over for efficient operation of the robot cell. However, malfunctions and errors can occur during the cycle, for which some form of correction is needed to restore the workcell to regular automatic operation. In most robot cells, it is necessary to stop the workcell when errors occur, and to provide human assistance for the corrective action. This generally results in production delays before the maintenance crew arrives to diagnose the problem and make repairs. There is a trend in programmable automation technology to attempt to endow the robot (or other automated equipment) with the capability to sense errors and malfunctions when they occur, and to take the necessary compensating action to restore the system to normal operation. This capability is referred to as error detection and recovery. By its name, error detection and recovery consists of two ingredients: error detection and error recovery. The detection problem is concerned with the use of the appropriate sensors to determine when an error has occurred. It also includes the associated intelligence to interpret the sensor signals so that errors can be properly recognized and classified. In general, errors in manufacturing can be classified as random errors, systematic errors, and illegitimate errors. Random errors are those that result from stochastic phenomena and are usually characterized by their statistical nature. For example, part size in a machining operation would be expected to vary randomly about some mean value. Depending on the amount of the variation, this could cause problems in a subsequent manufacturing process. Systematic errors are not determined by chance but by some bias that exists in the process. For example, an incorrect setting in the production machine or fixture would likely result in a systematic error in the product. The third class is the illegitimate error, typically resulting from an outright mistake, either by the equipment or by a human error. An error in the robot program would reflect this kind of mistake. Although this general classification of errors may be helpful in determining potential sources of malfunctions in a process, it is usually not sufficient to design an error detection and recovery system for a specific application. An example will serve to illustrate how the errors might be classified into more specialized categories for a given process.

**Example 11-5** In the context of an automated machining cell that is tended by a robot, the error categories would include: (1) tooling, (2) workpart, (3) process, (4) fixture, (5) machine tool, and (6) robot/end effector. This does not include the usual safety monitoring system that would probably be employed in the robot cell. In each of the categories, there are particular malfunctions and errors that could occur. The following list illustrates some of the possibilities:





Given that an error has occurred and that the error detection system has correctly sensed and classified the error, then certain corrective procedures can be initiated. The error recovery problem is concerned with defining and implementing the strategies that can be employed by the robot to correct or compensate for the malfunction that has occurred. The classification of errors is required during error detection because a specific recovery strategy must usually be developed to deal with each specific type of error. Recovery strategies can be grouped into the following general categories:

1. **Adjustments at the end of the current cycle**. This recovery strategy would represent a relatively low level of urgency. At the end of the ,current cycle, the robot program would branch to a subroutine to make the required corrections, then branch back to the main program.
2. **Adjustments during current cycle**. The error is sufficiently serious that corrective action must be taken during the current cycle of operation. However, it is not so urgent that the process must be stopped. The corrective action is typically accomplished by calling a special subroutine that has been designed to deal with the particular error.
3. **Stop process and invoke corrective algorithm**. The error in this case requires that the process be stopped, and that a subroutine be called to correct the error. At the end of the correction algorithm, the process can be resumed or restarted.
4. **Stop process -- call for help**. This action is usually taken either because the malfunction is one that cannot be corrected by the robot or because an unclassified error is identified for which no corrective algorithm has been designed. In either case, human assistance is required to restore the system

**4a.7 THE WORKCELL CONTROLLER**

The workcell control system is concerned with the coordination of the robot's activities with those of the other equipment in the cell. A number of options are available to satisfy the requirements of the workcell controller. These options include the use of the robot controller itself, relays, programmable controllers, and small stand-alone computers (minicomputers or microcomputers). The decision of which option to select depends on the complexity of the cell (e.g., the number of separate pieces of equipment, the number of separate control actions that must be controlled, and the number of robots in the cell), and whether the robot controller alone is capable of handling all of the activities. In the subsections below, we compare the various alternatives.

**The Robot Controller**

There are various types of control technology used for robot controllers. These include the simpler limited sequence controllers, electronic controllers, and computer controls. The more sophisticated types usually have a limited input/output capability to interface with other equipment. This input/output interface is provided specifically for the incorporation of interlocks in the workcell. The robot controller has the capability to tie the incoming signals to the work cycle program, so that the proper sequencing of output signals and robot motions can take place. The number of input/output ports might range between 10 and 20 for playback robots. A typical arrangement for the input/output module of the robot controller would be as follows:

Input ports (perhaps 10 to 25 input lines)-These would be used for incoming signals from external pieces of equipment. The signals would be binary (voltage on or off) and could be referenced as logical conditions in the robot program for purposes of interlocking. On newer controllers, the input ports would include the capacity to read in analog signals. Output ports (perhaps 10 to 25 output lines)-These ports would be used for output interlock signals to the external equipment. The signals would be initiated or terminated according to logical conditions in the robot program, thus resulting in some response by the external equipment. Again, some newer robots would have the capacity for analog outputs as well as binary signal outputs. Input port (perhaps five input lines)-These would be reserved for safety interlocks. Upon receipt of a signal from the external safety sensor on one of these lines, the controller would immediately interrupt the program, thus stopping the robot. In some cases, these input ports might be used to simply turn the power off to the manipulator.

This represents a limited input/output capacity for a workcell with any degree of complexity. Also, today's robot controllers are generally limited to sequence control and, as the above list indicate, often do not possess the capability to incorporate any significant safety monitoring or operator interfacing into the workcell control system. With the growing use of computer controls and the need by competing robot manufacturers to increase the control capabilities of their products, it is expected that future robot controllers will be equipped with enhanced input/output capacity and the capability to control intelligent robots.

**Electromechanical Relays**

An electromechanical relay is a control device used to actuate electrical circuits in response to changes in incoming signals. They are commonly used in industrial applications to provide sequence control of electrically operated equipment although they are gradually being displaced by more modern devices such as programmable controllers. Relays can be used to augment the capabilities of a robot controller in the design of a workcell control system. Their use would typically be reserved for simple robot cells, such as pick-and-place applications, and where the robot has very limited input/output capacity. With relays, it would be relatively easy to include a simple safety monitoring scheme in the workcell. Such a scheme might consist of a fence surrounding the work place with a safety gate to gain access to the cell. Using the appropriate sensors (e.g., a limit switch to indicate closure of the safety gate) the relays could be set up to stop the robot, perhaps by interrupting its power source, as soon as a hazardous condition was sensed. The limitations of relay control include the difficulty in interfacing with plant computer systems, their hard-wired configuration which makes it difficult to change over to a new workstation control task, and the fact that they are susceptible to mechanical wear and are less reliable than computer-type controls. The functions of a relay panel can be accomplished by a programmable controller, which avoids the above problems.

**Programmable Controllers**

Programmable controllers were introduced in the late 1960s as a replacement for systems of electromechanical relays. Up until that time, relay panels constituted the standard technique for accomplishing sequence control in industrial operations. The programmable controller was smaller in size, more reliable, more flexible, and its use could be readily learned by shop personnel who were familiar with the logic diagrams used for relay control panels. A programmable controller (PC) can be defined as a digitally operating device with programmable memory that is capable of generating output signals according to logic operations and other functions performed on input signals. The program for a PC determines the sequence of operations and the generation of input and output signals. A PC is programmed by specifying the same kinds of logic diagrams, called ladder diagrams, used for years to set up relay control panels. Other programming methods are also possible on many programmable controllers, including the use of symbolic notation similar to computer programming. The functions that can be accomplished on a programmable controller typically include:

Control relay functions-The generation of an output signal based on logic rules applied to one or more input signals. Timing functions-For example, the generation of an output signal for a specified length of time. Counting functions-An internal counter in the PC is used to sum the number of contact closures and generate an output signal when the sum reaches a certain level. Arithmetic functions-Some PCs can perform the basic arithmetic operations such as addition, subtraction, multiplication, and division. Analog control functions-Another feature which is available on some PCs is the capability to simulate analog functions, such as proportional, integral, and derivative control.

These functions permit the PC to perform as a powerful robot workcell controller. Some PCs have the capacity to accept several hundred input/output connections, significantly more than a typical robot controller. This means that the PC can control a more complex workcell with more activities taking place in the cell. An automobile body spot-welding line, in which many robots perform various welding operations, would use a programmable controller as the overall cell control device. In addition to handling a greater number of input and output signals, the programmable controller also possesses other features that are beyond the capability of most robot controllers. These features include:

Maintenance and diagnostic functions-The CRT terminal used to program the. PC can also be used in some systems to monitor the operation of the workcell. Some PCs have sophisticated diagnostic capabilities to quickly determine the origin of a problem when it occurs. Operator interface-The use of the PC as the robot cell controller allows greater capacity and flexibility to implement the operator interface. Display terminals can be included in complex cells to provide operating performance information about production rates, tool usage, equipment breakdowns, and other data. Printers can be included at the control station to provide hard copy reports about the cell performance. Safety monitoring-More sophisticated hazard monitoring systems can be implemented with programmable controllers. A greater number of safety conditions can be observed while the cell is operating than is possible with the robot controller alone.

**A Computer as the Workcell Controller**

Some robot applications have requirements for which a digital computer is the most appropriate method of workcell control. We are referring to the use of a stand-alone computer (generally a minicomputer or microcomputer) rather than the computer which is used as the r**o**bot control unit. In cases where a computer is the workcell controller, it would be used either in series with a programmable controller or as a substitute for the PC. The computer might perform other functions in the plant, and so it would be implemented to control the robot cell in a time-sharing mode of operation.

Also, the computer would probably form a component in a hierarchical computer network in the factory connected down to the programmable controller(s) and/or robot controller(s) in the cell, and connected up to the next hierarchical level in the plant. Programmable controllers are specialized devices that are designed to be interfaced with industrial processes. They are provided with input/output ports that can be directly wired to the plant equipment. This is an advantage over the digital computer, and special arrangements must be made to interface the computer to the industrial equipment in the cell. However, the PC has certain limitations in data processing and programming languages which give the computer an advantage in applications requiring these capabilities. Some examples of the kind of robot application features that might tend to favor the use of computers for workcell control would include the following:

Cases in which there are several cells whose operations must be coordinated. and significant amounts of data must be communicated between the cells. Cells in which the error detection and recovery problem constitutes a significant portion of the coding that must be programmed into the workcell operation. Where several different products are made on the same robot-automated production line, the operations at the different stations have to be coordinated and sequenced properly. Computers would be well suited to the data processing chores required in this type of application. In cases where the production lines are used for assembly operations, the various sizes and styles of the component parts must be sorted and matched to the particular model being assembled at each respective workstation along the line. Situations in which a high level of production scheduling and inventory control are required in the operation of the cell. Again, this type of data processing function might require the use of a computer in addition to or as a substitute for a programmable controller.

The differences between digital computers and programmable controllers are principally differences in applications rather than differences in basic technology. The PC can, in fact, be considered to be a specialized form of digital computer with dedicated features for input/output control of industrial equipment. The technologies of the two types of control devices are quite similar.

**4a.8 ROBOT CYCLE TIME ANALYSIS**

The amount of time required for the work cycle is an important consideration in the planning of the workcell. The cycle time determines the production rate for the job, which is a significant factor in the economic success of the robot installation. In the case of work performed by a human operator, the -time required to accomplish the cycle would be determined by one of several work measurement techniques. One of these work measurement techniques is called MTM (for Methods Time Measurement). With MTM, the work cycle is divided into its basic motion elements and standard time values are assigned to each of the elements to construct the time for the total cycle. The standard time values have been previously compiled by studying similar elements and analyzing the factors that determine the time required to perform the elements. For example, the time required for a human operator to transport an object from one place to another depends on such factors as the weight of the object, the distance the object is moved, and the precision with which the object is located at the end of the move. An approach similar to MTM has been developed by Nof and Lechtman8 at Purdue University for analyzing the cycle times of robot work. The method, called RTM (for Robot Time and Motion), is useful for estimating the amount of time required to accomplish a certain work cycle before setting up the workstation and programming the robot. This would allow an applications engineer to compare alternative methods of performing a particular robot task. It could even be utilized as an aid in selecting the best robot for a given application by comparing the performance of the different candidates on the given work cycle. The methodology of RTM is similar to MTM. There are 10 general categories of robot work cycle elements as presented in Table 11-1. The I 0 categories can be collected into four major groups:

1. **Motion elements**. These are the manipulator movements, performed either with or without load.
2. **Sensing elements**. These are sensory activities performed by robots equipped with sensing capabilities. Examples include vision sensing, force sensing, and position sensing.
3. **End effector elements**. These elements relate to the action of the gripper or tool attached to the robot wrist as its end effector.
4. **Delay elements**. These are delay times resulting from waiting and processing conditions in the work cycle.

**4b) Material Transfer, Machine Loading/Unloading**

There are many robot applications in which the robot is required to move a workpart or other material from one location to another. The most basic of these applications is where the robot picks the part up from one position and transfers it to another position. In other applications, the robot is used to load and/or unload a production machine of some type. In this book we divide material-handling applications into two specific categories:

1. Material transfer applications 2. Machine loading/unloading applications

There are other robot applications which involve parts handling. These include assembly operations and holding parts during, inspection.

**4b.1 GENERAL CONSIDERATIONS IN ROBOT MATERIAL HANDLING**

In planning an application in which the robot will be used to transfer parts, load a machine, or other similar operation, there are several considerations that must be reviewed.

and we itemize them below as a reference checklist.

1. Part positioning and orientation. In most parts-handling applications the parts must be presented to the robot in a known position and orientation. Robots used in these applications do not generally possess highly sophisticated sensors (e.g., machine vision) that would enable them to seek out a part and identify its orientation before picking it up.
2. Gripper design. Special end effectors must be designed for the robot to grasp and hold the workpart during the handling operation.
3. Minimum distances moved. The material-handling application should be planned so as to minimize the distances that the parts must be moved. This can be accomplished by proper design of the workcell layout (e.g., keeping the equipment in the cell close together), by proper gripper design (e.g., using a double gripper in a machine loading/unloading operation), and by careful study of the robot motion cycle.
4. Robot work volume. The cell layout must be designed with proper consideration given to the robot's capability to reach the required extreme locations in the cell and still allow room to maneuver the gripper.
5. Robot weight capacity. There is an obvious limitation on the materialhandling operation that the load capacity of the robot must not be exceeded. A robot with sufficient weight-carrying capacity must be specified for the application.
6. Accuracy and repeatability. Some applications require the materials to be handled with very high precision. Other applications are less demanding in this respect. The robot must be specified accordingly.
7. Robot configuration, degrees of freedom, and control. Many parts transfer operations are simple enough that they can be accomplished by a robot with two to four joints of motion. Machine-loading applications often require more degrees of freedom. Robot control requirements are unsophisticated for most material-handling operations. Palletizing operations, and picking parts from a moving conveyor are examples where the control requirements are more demanding.
8. Machine utilization problems. It is important for the application to effectively utilize all pieces of equipment in the cell. In a machine loading/unloading operation, it is common for the robot to be idle while the machine is working, and the machine to be idle while the robot is working. In cases where a long machine cycle is involved, the robot is idle a high proportion of the time. To increase the utilization of the robot, consideration should be given to the possibility for the robot to service more than a single machine.

We now proceed to deal with the specific cases of material transfer and machine loading/unloading applications in the following two sections.

**4b.2 MATERIAL TRANSFER APPLICATIONS**

Material transfer applications are defined as operations in which the primary objective is to move a part from one location to another location. They are usually considered to be among the most straightforward of robot applications to implement. The applications usually require a relatively unsophisticated robot, and the interlocking requirements with other equipment are typically uncomplicated. These applications are sometimes called pick-and-place operations because the robot simply picks the part from one location and places it in another location. Some material transfer applications have motion patterns that change from cycle to cycle, thus requiring a more sophisticated robot. Palletizing and depalletizing operations are examples of this more complicated case. In this type of application, the robot must place each part in a different location on the pallet, thus forcing the robot to remember or compute a separate motion cycle until the pallet is fully loaded.

**Pick-and-Place Operations**

As defined above, pick-and-place operations involve tasks in which the robot picks up the part at one location and moves it to another location. In the simplest case, the part is presented to the robot by some mechanical feeding device or conveyor in a known location and orientation. The known location is a stationary location, achieved either by stopping the conveyor at the appropriate position, or by using a mechanical stop to hold the part at the stationary location. An input interlock (commonly based on using a simple limit switch) would be designed to indicate that the part is in position and ready for pickup. The robot would grasp the part, pick it up, move it, and position it at a desired location. The orientation of the part remains unchanged during the move. The desired location is usually at a position where there is the capability to move the part out of the way for the next delivery by the robot. This basic case is illustrated in Fig. 13-1. In this simple case, the robot needs only 2 degrees of freedom. As shown in the figure, l degree of freedom is needed to lift the part from the pickup point and put it down at the drop-off point. The second degree of freedom is required to move the part between these two positions. In some pick-and-place operations, a reorientation of the workpart is accomplished during the move. This part reorientation requires a robot with one or more additional degrees of freedom.

One complication encountered in material transfer operations is when the robot is required to track a moving pickup point. In robotic materials handling, tracking arises when parts are carried along a continuously moving conveyor, typically an overhead hook conveyor, and the robot is required to pick parts from the conveyor. The opposite case is when the robot must put parts onto the moving conveyor. In either case a more sophisticated sensor-interlock system is required to determine the presence and location of the parts in the robot's tracking window.

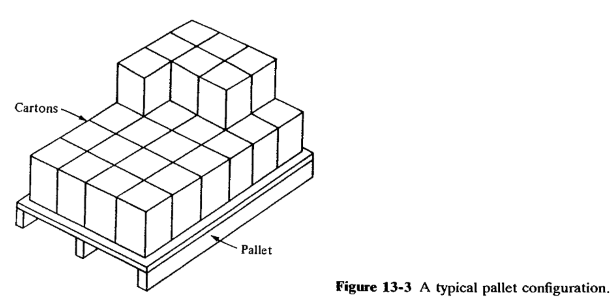
Another complication arises in material transfer operations (as well as palletizing and other operations) when different objects are being handled by the same robot. In material transfer, a single conveyor might be used to move more than one type of part. The robot must be interfaced to some type of sensor system capable of distinguishing between the different parts so that the robot can execute the right program subroutine for the particular part. For example, there may be differences in the way the part must be retrieved from the conveyor due to part configuration, and the placement of the part by the robot may vary for different parts. In other cases where multiple items arc handled, the information system which supports the workcell can be used to keep track of where each item is located in the workcell. The information support system would be used either in lieu of or in addition to sensors. The following example illustrates a material-handling situation which uses both sensors and a hierarchical information system to support the control of the robot cell.



**Palletizing and Related Operations**

The use of pallets for materials handling and storage in industry is widespread. Instead of handling individual cartons or other containers, a large number of these containers are placed on a pallet, and the pallet is then handled. The pallets can be moved mechanically within the plant or warehouse by fork lift trucks or conveyors. Shipments of palletized product to the customer are very common because of the convenience in handling, both at the manufacturer's warehouse and the customer's receiving department.

The only handling of the individual cartons arises when the product is placed onto the pallet (palletizing) or when it is removed from the pallet (depalletizing). We will discuss palletizing as the generic operation. The loading of cartons onto pallets is typically heavy work, performed manually by unskilled labor. It is also repetitive work (picking cartons up at one location and putting them down at another location) except that the locations change from carton to carton. A typical pallet configuration is illustrated in Fig. 13-3, showing how each container must be placed at a different location on the pallet. The variation in carton location is in three dimensions, not simply two dimensions, since the pallets are usually stacked on top of each other in layers. Robots can be programmed to perform this type of work. Because the motion pattern varies in the palletizing operation, a computer-controlled robot using a high-level programming language is convenient. This feature facilitates the mathematical computation of the different pallet locations required during the loading of a given pallet. The kinds of programming capabilities required in palletizing were discussed in our previous chapters on robot programming. A less sophisticated robot limited to leadthrough programming can also be used, but the programming becomes laborious because each individual carton location on the pallet must be individually taught. Another technical problem that must be addressed is that when humans perform palletizing, the cartons are often randomly located prior to loading.



Unless some sensor scheme is used to identify these carton locations, they must be delivered to a known pickup point for the robot.

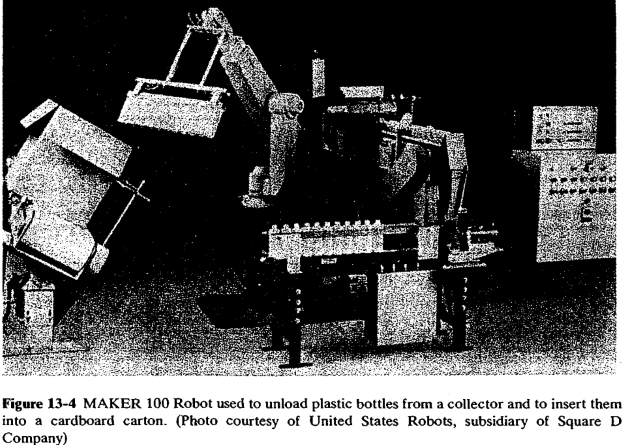
There are a number of variations on the palletizing operation, all of which use a similar work cycle and a robot with the same general features needed for the generic case. These other operations include:

Depalletizing operations, the reverse of palletizing, in which the robot removes cartons from a pallet and places them onto a conveyor or other location.

Inserting parts into cartons from a conveyor. This is similar to palletizing. Figure 13-4 shows an example of this operation.

Removing parts from cartons. This is the reverse of the preceding situation.

Stacking and unstacking operations, in which objects (usually flat objects such as metal sheets or plates) are stacked on top of each other.



In palletizing and related operations, the robot may be called on to load different pallets differently. Reasons for these differences would include the following: The pallets may vary in size; different products may be loaded onto the pallets; and there may be differences in the numbers and combinations of cartons going to different customers. To deal with these variations, methods of identifying the cartons and/or pallets and the way in which they are to be loaded or unloaded must be devised. Bar codes and other optical schemes are sometimes used to solve the identification problem. Differences in the loading or unloading of the pallets must be accomplished by means of program subroutines which can be called by the workcell controller. For depalletizing operations, the optical reader system would identify the pallet and the appropriate unloading subroutine would then be applied for that pallet. For palletizing operations, the systems problems can become more complicated because there may be an infinite number of different situations that could arise. For example, if the robot were used to palletize cartons for different customer orders, it is conceivable that each customer order would be different. A method would have to be devised for delivering the correct combination of cartons to the palletizing workstation, and integrating that process with the robot loading procedure. In the future, robots· may be equipped with sufficient intelligence to figure out how to load the different cartons onto the pallet. At the time of this writing, it is a systems problem of significant proportions.

**4b.3 MACHINE LOADING AND UNLOADING**

These applications are material-handling operations in which the robot is used to service a production machine by transferring parts to and/or from the machine. There are three cases that fit into this application category:

**Machine load/unload**. The robot loads a raw workpart into the process and unloads a finished part. A machining operation is an example of this case.

**Machine loading.** The robot must load the raw work part or materials into the machine but the part is ejected from the machine by some other means. In a press working operation, the robot may be programmed to load sheet metal blanks into the press, but the finished parts are allowed to drop out of the press by gravity.

**Machine unloading**. The machine produces finished parts from raw materials that are loaded directly into the machine without robot assistance. The robot unloads the part from the machine. Examples in this category include die casting and plastic modeling applications.

The application is best typified by a robot-centered workcell which consists of the production machine, the robot, and some form of parts delivery system. To increase the productivity of the cell and the utilization of the robot, the cell may include more than a single production machine. This is desirable when the automatic machine cycle is relatively long, hence causing the robot to be idle a high proportion of the time. Some cells are designed so that each machine performs the same identical operation. Other cells are designed as flexible automated systems in which different parts follow a different sequence of operations at different machines in the cell. In either case, the robot is used to perform the parts handling function for the machines in the cell. Robots have been successfully applied to accomplish the loading and/or unloading function in the following production operations:

* Die casting
* Plastic molding
* Forging and related operations
* Machining operations
* Stamping press operations

**4b.3.1 Die Casting:**

Die casting is a manufacturing process in which molten metal is forced into the cavity of a mold under high pressure. The mold is called a die (hence the name, die casting). The process is used to cast metal parts with sufficient accuracy so that subsequent finishing operations are usually not required. Common metals used for die-casted parts include alloys of zinc, tin, lead, aluminum, magnesium, and copper.

The die consists of two halves that are opened and closed by a die casting machine. During operation the die is closed and molten metal is injected into the cavity by a pump. To ensure that the cavity is filled, enough molten metal is forced into the die that it overflows the cavity and creates "flash" in the space between the die halves. When the metal has solidified, the die is opened and the cast part is ejected, usually by pins which push the part away from the mold cavity. When the part is removed from the machine, it is often quenched (to cool the part) in a water bath. The flash that is created during the casting process must be removed subsequently by a trimming operation which cuts around the periphery of the part. Thus, the typical die-casting production cycle consists of casting, removing the part from the machine, quenching, and trimming.

The production rates in the die-casting process range from about 100 up to 700 openings of the die per hour, depending on type of machine, the metal being cast, and the design of the part. For small parts, the die can be designed with more than one cavity, thus multiplying the number of parts made for each casting cycle. The die-casting machines have traditionally been tended by human operators. The work tends to be hot, repetitive, dirty, and generally unpleasant for humans.

Perhaps because of these conditions, die casting was one of the very first processes to which robots were applied. The first use of a robot in die casting was around 1961. Engel berger cites one instance in which a Unimate robot had been used in a die casting application for over 90,000 hours2

The die-casting process represents a relatively straightforward application for industrial robots. The alterations required of the die-casting machine are minimal, and the interlocking of the robot cycle with the machine cycle can be accomplished by simple limit switches. Few problems are encountered in either the programming of the robot or the design of the gripper to remove the part from the machine when the die is opened. The process requires only that the robot unload the die-casting machine, since the metal is in the molten state before the part is formed. On some die-casting machines (called cold-chamber die-casting machines), the molten metal must be ladled from the melting container into the injection system. This part of the process is more difficult for robots to accomplish.

**4b.3.2 Plastic Molding:**

Plastic molding is a batch-volume or high-volume manufacturing process used to make plastic parts to final shape and size. The term plastic molding covers a number of processes, including compression molding, injection molding, thermoforming, blow molding, and extrusion.

Injection molding is the most important commercially, and is the process in this group for which robots are most often used. The injection-molding operation is quite similar to die casting except for the differences in materials being processed. A thermoplastic material is introduced into the process in the form of small pellets or granules from a storage hopper. It is heated in a heating chamber to 200 to 300°C to transform it into semifluid (plastic) state and injected into the mold cavity under high pressure. The plastic travels from the heating chamber into the part cavity through a sprue-and-runner network that is designed into the mold. If too much plastic is injected into the mold, flash is created where the two halves of the mold come together. If too little material is injected into the cavity, sink holes and other defects are created in the part, rendering it unacceptable. When the plastic material has hardened sufficiently, the mold opens and the part(s) are removed from the mold. Injection molding is accomplished using an injection-molding machine, a highly sophisticated production machine capable of maintaining close control over the important process parameters such as temperature, pressure, and the amount of material injected into the mold cavity. Traditionally, injection molding machines have been operated on a semiautomatic cycle, with human operators used to remove the parts from the mold. Many injection-molding operations can be fully automated so long as a method can be developed for removing the parts from the mold at the end of the molding cycle. If a part sticks in the mold, considerable damage to the mold can occur when it closes at the beginning of the next cycle. Methods of removing the parts from the mold include: gravity to cause the parts to drop out of the mold, directing an air stream to force the parts out of the mold, and the use of robots to reach into the mold and remove the parts. The selection of the method depends largely on the characteristics of the molding job (part size, weight, how many parts to be molded per shot).

Industrial robots are sometimes employed to unload injection-molding machines when other less expensive automatic methods are deemed to be insufficiently reliable. One of the robot application problems in injection molding is that the production times are considerably longer than in die casting, hence causing the robot to be idle for a significant portion of the cycle. When humans tend the molding machines, this time can be utilized to perform such tasks as cutting the parts from the sprue-and-runner system, inspecting the parts, and removing the flash from the parts if that is necessary. However, some of these tasks are difficult for a robot to perform, and methods must be devised to accomplish these activities that do not rely on a human operator performing the unloading function. Cutting the parts from the sprue-and-runner system can be readily accomplished by the robot using a trimming apparatus similar to the setup used in die casting for trimming the flash from the casting. Part inspection and flash removal are not as easily accomplished by the robot.

Another issue arising when long molding cycle times are involved is whether the robot should be used to tend one machine or two. If two molding machines are tended by the robot, there is a significant likelihood that the two molding cycles will be different. This creates machine interference problems, in which one machine must wait for the robot because it is presently engaged in unloading the other machine. This waiting can lead to problems in overheating of the plastic and upsetting of the delicate balance between the various process parameters in injection molding.

**4b.3.3 Forging and Related Operations:**

Forging is a metalworking process in which metal is pressed or hammered into the desired shape. It is one of the oldest processes and derived from the kinds of metalworking operations performed by blacksmiths in ancient times. It is most commonly performed as a hot working process in which the metal is heated to a high temperature prior to forging. It can also be done as a cold working process. Cold forging adds considerable strength to the metal and is used for high-quality products requiring this property such as hand tools (e.g., hammers and wrenches). Even in hot forging, the metal flow induced by the hammering process adds strength to the formed part.

The term forging includes a variety of metalworking operations, some of which are candidates for automation using robots. These operations include die forging and upset forging. Other processes in the forging category include press forging and roll die forging. Generally these processes do not lend themselves to the use of robots for parts loading and unloading of the machines.

Die forging is a process accomplished on a machine tool called a drop hammer in which the raw billet is hit one or more times between the upper and lower portions of a forging die. The die often has several cavities of different shapes which allow the billet to be gradually transformed from its elementary form into the desired final shape. The drop hammer supplies mechanical energy to the operation by means of a heavy ram to which the upper portion of the forging die is attached. The ram is dropped onto the part, sometimes being accelerated by steam or air pressure. Die forging can be carried out either hot or cold.

Upset forging, also called upsetting, is a process in which the size of a portion of the workpart (usually a cylindrical part) is increased by squeezing the material into the shape of a die. The formation of the head on a bolt is usually made by means of an upsetting operation. The process is performed by an upsetting machine, also called a header. The blank (unformed raw workpart) is clamped by the two halves of a die possessing the desired shape of the product. The die is open on one end, and a plunger is forced by the upsetting machine into the blank causing it to take the shape of the die. Upsetting is often used in high-volume production of hardware items such as bolts, nails, and similar items. In these cases, the economics permit the use of fixed automation to produce the parts. In other cases, where the production of parts is in medium-sized batches, automation can sometimes be accomplished using industrial robots.

Forging, especially hot forge operations, is one of the worst industrial jobs for humans. The environment is noisy and hot, with temperatures at the workplace well above l00°F for hot forging. The air in the forge shop is generally filled with dirt, furnace fumes, and lubricant mist. The operation itself is repetitive, often requiring considerable physical strength to move and manipulate the heavy parts during the operation. The human operator experiences the blows from the drop hammer directly through the grasping tongs used to hold the part and in the form of vibration through the floor.

Unfortunately, the process is not easily adapted for robots. Some of the technical and economic problems include:

The forging hammers and upsetting machines used for low- and medium production runs are typically older machines, designed for manual operation, and do not lend themselves to the interfacing required for robotics automation.

Short production runs are typical in many forge shops, thus making it difficult to justify the robot setup and programming effort for any single part.

The parts occasionally stick in the dies. This can be readily detected by a human operator but poses problems for the robot. To minimize the frequency of sticking, the human operator periodically sprays lubricant into the die openings. The robot would have to be equipped and programmed to do this also.

The design of a gripper for forging is a significant engineering problem for several reasons. First, the parts are hot, perhaps 2000°F, and the gripper must be protected against these temperatures. Second, the gripper must be designed to withstand the shock from the hammer blows because the parts must typically be held in position by the robot during the process. Third, the gripper must be designed to accommodate substantial changes in the shape of the parts during successive hits in the forging cycle. Some aspects of the forging process require operator judgment. The part must be heated to a sufficient temperature in order to successfully perform the hot working operation and the human operator often makes this judgment. A cold part would probably ruin the die. The raw workparts are generally placed in the heating furnace at random, and selecting the parts that are ready to be formed is an operator decision. Another problem is that different parts can require a different number of hammer strokes to form the final shape, a judgment that is made by the operator.

Each of these problem areas must be addressed in order for the robot forging application to be a success. Many of the problems are solved by making considerable use of interlocks and sensors. These devices permit the determination of such process variables as part temperature before processing, the presence of the workpart in the gripper, whether the part is stuck in the die, whether the robot arm is clear of the ram before operation of the drop hammer, and other factors.

**4b.3.4 Machining Operations**

Machining is a metalworking process in which the shape of the part is changed by removing excess material with a cutting tool. it is considered to be a secondary process in which the final form and dimensions are given to the part after a process such as casting or forging has provided the basic shape of the part. There are a number of different categories of machining operations. The principal types include turning, drilling, milling, shaping, planning and grinding. Commercially, machining is an important metalworking process and is widely used in many different products, ranging from those that are made in low quantities to those produced in very high numbers. In mid - volume and high-volume production, the operation is very repetitive with the same machining sequence being repeated on part after part.

The machine tools that perform machining operations have achieved a relatively high level of automation after many years of development. In particular, the use of computer control (e.g., computer numerical control and direct numerical control) permits this type of equipment to be interfaced with relative ease to similarly controlled equipment such as robots.

Robots have been successfully utilized to perform the loading and unloading functions in machining operations. The robot is typically used to load a raw workpart (a casting, forging, or other basic form) into the machine tool and to unload the finished part at the completion of the machining cycle. Figure 13-5 illustrates a machine tool loading and unloading operation in which the finished parts are palletized (lower left corner of the figure) after the machining cycle. The following robot features generally contribute to the success of the machine tool load/unload application:

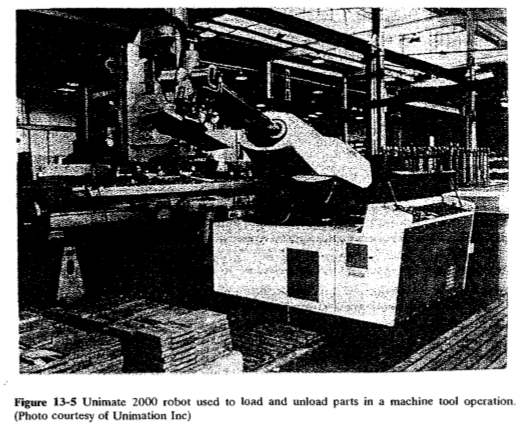
**Dual gripper**. The use of a dual gripper permits the robot to handle the raw workpart and the finished part at the same time. This permits the production cycle time to be reduced.

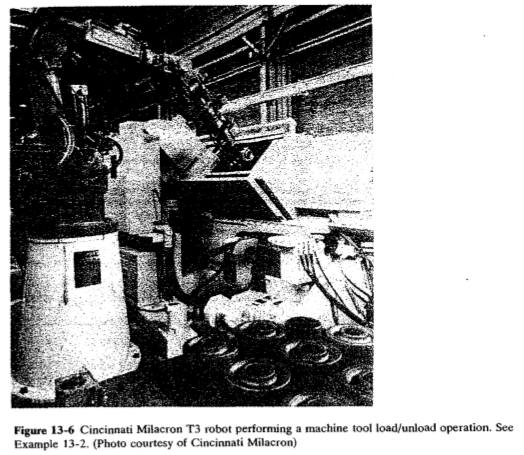
**Up to six joint motions**. A large number of degrees of freedom of the arm and wrist are required to manipulate and position the part in the machine tool.

**Good repeatability**. A relatively high level of precision is required to properly position the part into the chuck or other workholding fixture m the machine tool.

**Palletizing and depalletizing capability**. In midvolume production, the raw parts are sometimes most conveniently presented to the workcell and delivered away from the workcell on pallets. The robot's controller and programming capabilities must be sufficient to accommodate this requirement.

**Programming features**. There are several desirable programming features that facilitate the use of robots in machining applications. In machine cells used for batch production of different parts, there is the need to perform some sort of changeover of the setup between batches. Part of this changeover procedure involves replacing the robot program for the previous batch with the program for the next batch. The robot should be able to accept disk, tape, or other storage medium for ease in changing programs. Another programming feature needed for machining is the capability to handle irregular elements, such as tool changes or pallet changes, in the program.





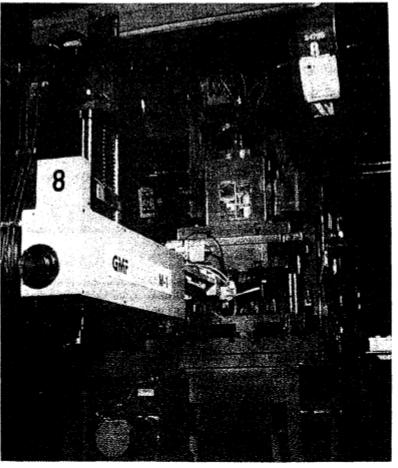


Figure 13-7 GMF robot performing machine loading/unloading task at a stamping press. (Photo courtesy of GMF Robotics)

**4b.3.5 Stamping Press Operations**

Stamping press operations are used to cut and form sheet metal parts. The process is performed by means of a die set held in a machine tool called a press (or stamping press). The sheet metal stock used as the raw material in the process comes in several forms, including coils, sheets, and individual flat blanks. When coil stock is fed into the press, the process can be made to operate in a highly automated manner at very high cycle rates. When the starting material consists of large flat sheets or individual blanks, automation becomes more difficult. These operations have traditionally been performed by human workers, who must expose themselves to considerable jeopardy by placing their hands inside the press in order to load the blanks. During the last decade, the Occupational Safety and Health Act (OSHA) has required certain alterations in the press in order to make its operation safer. The economics of the OSHA requirements have persuaded many manufacturers to consider the use of robots for press loading as alternatives to human operators. Noise is another factor which makes pressworking an unfriendly environment for humans.

Robots are being used for handling parts in pressworking operations, largely as a result of the safety issue. The typical task performed by the robot is to load the flat blanks into the press for the stamping operation. There are variations in the way this can be done. In forming operations, the robot can be used to hold the blank during the cycle so that the formed part is readily removed from the press. In the case of many cutting operations, the robot loads the blank into the press, and the parts fall through the die during the press cycle. Another robot application in pressworking involves the transfer of parts from one press to another to form an integrated pressworking cell. Figure 13-7 shows a TLL configuration robot working in conjunction with a pressworking operation.

One of the limiting factors in using industrial robots for press loading is the cycle time of the press. Cycle times of less than a second are not uncommon in pressworking. These cycle rates are too fast for currently available commercial robots. There is generally a direct relationship between the physical size of the part and the press cycle time required to make the part. Bigger presses are needed to stamp bigger parts and bigger presses are inherently slower. Accordingly, robots are typically used in pressworking for larger parts.

**Module -5**

**Processing Operations, Assembly & Inspection:** Spot welding, continuous arc welding, spray coating, other processing operations using robots. Assembly and robotic assembly automation, parts presentation methods, assembly operations, compliance and remote center compliance (RCC) device, assembly system configurations, designing for robotic assembly, inspection automation. [Textbook-1]

**Autonomous Mobile Robots: Introduction, Planning &Navigation:** Introduction, basic control scheme for mobile robots (only basic understanding of perception, localization, path planning & motion control). [Textbook-3]

In addition to parts-handling applications, there is a large class of applications in which the robot actually performs work on the part. This work almost always requires that the robot's end effector is a tool rather than a gripper. Accordingly, the use of a tool to perform work is a distinguishing characteristic of this group of applications. The type of tool depends on the processing operation that is performed. We divide the processing operations that are performed by a robot into the following categories for purposes of organizing this chapter:

1. Spot welding
2. Continuous arc welding
3. Spray coating
4. Other processing operations

The two welding categories represent important application areas for robots. Spot welding is probably the single most common application for industrial robots in the United States today because they are widely used in automobile body assembly lines to weld the frames and panels together. Arc welding is an application that is expected to grow in use as we develop the technology required for using robots in this process. Spray coating usually means spray painting, an operation that is accompanied by an unhealthy work environment for humans, and therefore represents a good opportunity for robots. We use the term "spray coating" to indicate that there are additional applications beyond painting for a robot to spray a substance onto a surface. The final category in the listing above is a miscellaneous applications area. It includes certain machining operations, polishing, deburring, and other processing operations. These operations are usually, but not always, characterized by the use of a rotating spindle by the robot. We discuss the four categories of operations and how robots are used to accomplish these operations in the following sections.

**5a.1 SPOT WELDING**

As the term suggests, spot welding is a process in which two sheet metal parts are fused together at localized points by passing a large electric current through the parts where the weld is to be made. The fusion is accomplished at relatively low voltage levels by using two copper (or copper alloy) electrodes to squeeze the parts together at the contact points and apply the current to the weld area. The electric current results in sufficient heat in the contact area to fuse the two metal parts, hence producing the weld.

The two electrodes have the general shape of a pincer. With the two halves of the pincer open, the electrodes are positioned at the point where the parts are to be fused. Prior clamping or fixturing of the parts is usually required to hold the pieces together for the process. The two electrodes are squeezed together against the mating parts, and the current is applied to cause heating and welding of the contacting surfaces. Then the electrodes are opened and allowed to cool for the next weld. A water circulation system is often used to accelerate the cooling of the electrodes. The actual welding portion of the sequence typically requires less than a second. Therefore, the rates of production in spot welding are largely dependent on the time required for positioning of the welding electrodes and the parts relative to each other. Another factor that affects production rate is the wear of the electrodes. Because of the heat involved in the process, the tips of the electrodes gradually lose their shape and build up a carbon deposit which affects their electric resistance. Both of these effects reduce the quality of the welds made. Therefore, the electrode tips must periodically be dressed to remove the deposits and restore the desired shape.

Spot welding has traditionally been performed manually by either of two methods. The first method uses a spot-welding machine in which the parts are inserted between the pair of electrodes that are maintained in a fixed position. This method is normally used for relatively small parts that can be easily handled.

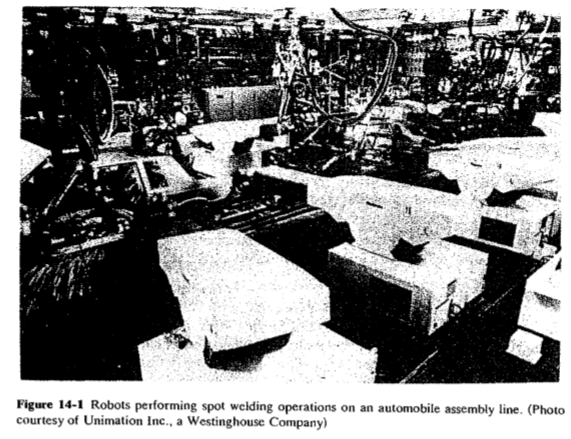
The second method involves manipulating a portable spot-welding gun into position relative to the parts. This would be used for larger work such as automobile bodies. The word "portable" is perhaps an exaggeration. The welding gun consists of the pair of electrodes and a frame to open and close the electrodes. In addition, large electrical cables are used to deliver the current to the electrodes from a control panel located near the workstation. The welding gun with cables attached is quite heavy and can easily exceed 100 lb in weight. To assist the operator in manipulating the gun, the apparatus is suspended from an overhead hoist system. Even with this assistance, the spot-welding gun represents a heavy mass and is difficult to manipulate by a human worker at the high rates of production desired on a car body assembly line. There are often problems with the consistency of the welded products made on such a manual line as a consequence of this difficulty.

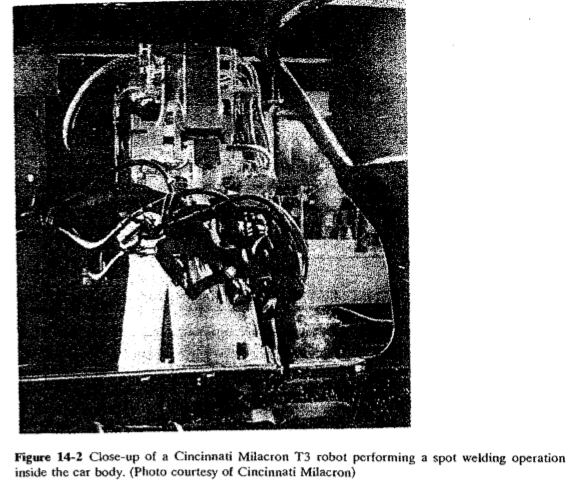
**5a.1.1 Robots in Spot Welding**

As a result of these difficulties, robots have been employed with great success on this type of production line to perform some or all of the spot-welding operations. A welding gun is attached as the end effector to each robot's wrist, and the robot is programmed to perform a sequence of welds on the product as it arrives at the workstation. Some robot spot-welding lines operate with several dozen robots all programmed to perform different welding cycles on the product. Today, the automobile manufacturers make extensive use of robots for spot welding. In 1980 it was reported3 that there were 1200 robots used in this application. Figure 14-1 shows an overview of an automobile body assembly line in which robots are used to perform the spot-welding operations. Figure 14-2 shows a close-up of a spot-welding gun mounted on a Cincinnati Milacron T3 robot performing its task inside the car body.

The robots used in spot welding must possess certain capabilities and features to perform the process. First, the robot must be relatively large. It must have sufficient payload capacity to readily manipulate the welding gun for the application. The work volume must be adequate for the size of the product. The robot must be able to position and orient the welding gun in places on the product that might be difficult to access. This might result in the need for an increased number of degrees of freedom. The controller memory must have enough capacity to accomplish the many positioning steps required for the spot-welding cycle. In some applications, the welding line is designed to produce several different models of the product. Accordingly, the robot must be able to switch from one programmed welding sequence to another as the models change. For welding lines in which there are multiple robots, programmable controllers are used to keep track of the different models at the various welding stations and to download the programs to the robots at individual workstations as needed.

The benefits that result from automation of the spot-welding process by means of robots are improved product quality, operator safety, and better control over the production operation. Improved quality is in the form of more consistent welds and better repeatability in the location of the welds. Even robots with relatively unimpressive repeatability specifications are able to locate the spot welds more accurately than human operators. Improved safety results simply because the human is removed from a work environment where there are hazards from electrical shocks and burns. The use of robots to automate the spot-welding process should also result in improvements in areas such as production scheduling and in-process inventory control. The maintenance of the robots and the welding equipment becomes an important factor in the successful operation of an automated spot-welding production line.

****

****

**5a. 2 CONTINUOUS ARC WELDING**

Arc welding is a continuous welding process as opposed to spot welding which might be called a discontinuous process. Continuous arc welding is used to make long welded joints in which an airtight seal is often required between the two pieces of metal being joined. The process uses an electrode in the form of a rod or wire of metal to supply the high electric current needed for establishing the arc. Currents are typically 100 to 300 A at voltages of 10 to 30 V. The arc between the welding rod and the metal parts to be joined produces temperatures that are sufficiently high to form a pool of molten metal to fuse the two pieces together. The electrode can also be used to contribute to the molten pool, depending on the type of welding process.

Arc welding is usually performed by a skilled human worker who is often assisted by a person called a fitter. The purpose of the fitter is to organize the work and to fixture the parts for the welder. The working conditions of the welder are typically unpleasant and hazardous. The arc from the welding process emits ultraviolet radiation which is injurious to human vision. As a result, welders are required to wear eye protection in the form of a welding helmet with a dark window. This dark window filters out the dangerous radiation, but it is so dark that the welder is virtually blind while wearing the helmet except when the arc is struck. Other aspects of the process are also hazardous. The high temperatures created in arc welding and the resulting molten metals are inherently dangerous. The high electrical current used to create the arc is also unsafe. Sparks and smoke are generated during the process and these are a potential threat to the operator.

There are a variety of arc-welding processes, and the reader is referred to the available manufacturing process texts for more details than we can include here. For robot applications, two types of arc welding seem the most practical: gas metal arc welding (GMAW) and gas tungsten arc welding (GTAW). GMA welding (also called MIG welding for metal inert gas welding) involves the use of a welding wire made of the same or similar metals as the parts being joined. The welding wire serves as the electrode in the arc-welding process. The wire is continuously fed from a coil and contributes to the molten metal pool used in the fusion process. GMA welding is typically used for welding steel. In GTA welding (also called TIG welding for tungsten inert gas welding), a tungsten rod is used as the electrode to establish the arc. The melting point of tungsten is relatively high, and therefore the electrode does not melt during the fusion process. If filler metal must be added to the weld, it must be added separately from the electrode. The GTA process is typically used for welding aluminum, copper, and stainless steel. In both GMA and GTA welding, inert gases such as helium or argon are used to surround the immediate vicinity of the welding arc to protect the fused surfaces from oxidation.

**5a. 2.1 Problems for Robots in Arc Welding**

Because of the hazards for human workers in continuous arc welding, it is logical to consider industrial robots for the process. However, there are significant technical and economic problems encountered in applying robots to arc welding. Continuous arc welding is commonly used in the fabrication industries where products consisting of many components are made in low quantities. It is difficult to justify automation of any form in these circumstances. A related problem is that arc welding is often performed in confined areas that are difficult to access, such as the insides of tanks, pressure vessels, and ship hulls. Humans can position themselves into these areas more readily than robots.

One of the most difficult technical problems for welding robots is the presence of variations in the components that are to be welded. These variations are manifested in two forms. One is the variation in the dimensions of the parts in a batch production job. This type of dimensional variation means that the arc-welding path to be followed will change slightly from part to part. The second variation is in the edges and surfaces to be welded together. Instead of being straight and regular, the edges are typically irregular. This causes variations in the gap between the parts and other problems in the way the pieces mate together prior to the welding process. Human welders are able to compensate for both of these variations by changing certain parameters in the welding process (e.g., adjusting the welding path, changing the speed at which the joint is traversed, depositing more filler metal where the gap is large, etc.). Industrial robots do not possess the sensing capabilities, skills, and judgment of human welders to make these compensations. There are two approaches to compensate for these variations and irregularities in robot welding applications:

1. Correct the upstream production operations so that the variations are reduced to the point where they do not create a problem in the robot welding process.
2. Provide the robot with sensors to monitor the variations in the welding process and the control logic to compensate for part variations and weld gap irregularities.

Correction of the production operations that deliver parts to the arc-welding process is an attractive alternative because it tends to contribute to the overall quality of the product, and because it simplifies the welding robot project. The potential disadvantage of this approach is that it is likely to increase the cost of manufacturing the individual components because their dimensions must be held to closer tolerances. The second approach represents an area of intensive research and development activity in robotics.

**Features of the Welding Robot**

An industrial robot that performs arc welding must possess certain features and capabilities. Some of the technical considerations in arc-welding applications are discussed in the following:

1. **Work volume and degrees of freedom**. The robot's work volume must be large enough for the sizes of the parts to be welded. A sufficient allowance must be made for manipulation of the welding torch. Also, if two part holders are included in the workstation, the robot must have adequate reach to perform the motion cycle at both holders. Five or six degrees of freedom are generally required for arc-welding robots. The number is influenced by the characteristics of the welding job and the motion capabilities of the parts manipulator. If the parts manipulator has 2 degrees of freedom, this tends to reduce the requirement on the number of degrees of freedom possessed by the robot.
2. **Motion control system.** Continuous-path control is required for arc welding. The robot must be capable of a smooth continuous motion in order to maintain uniformity of the welding seam. In addition, the welding cycle requires a dwell at the beginning of the movement in order to establish the welding puddle, and a dwell at the end of the movement to terminate the weld.
3. **Precision of motion.** The accuracy and repeatability of the robot determines to a large extent the quality of the welding job. The precision requirements of welding jobs vary according to size and industry practice, and these requirements should be defined by each individual user before selecting the most appropriate robot.
4. **Interface with other systems**. The robot must be provided with sufficient input/output and control capabilities to work with the other equipment in the cell. These other pieces of equipment are the welding unit and the parts positioners. The cell controller must coordinate the speed and path of the robot with the operation of the parts manipulator and the welding parameters such as wire feed rate and power level.
5. **Programming**. Programming the robot for continuous arc welding must be considered carefully. To facilitate the input of the program for welding paths with irregular shapes, it is convenient to use the walkthrough method in which the robot wrist is physically moved through its motion path. For straight welding paths, the robot should possess the capability for linear interpolation between two points in space. This permits the programmer to define the beginning and end points of the path and the robot is capable of computing the straight line trajectory between the points.

Some welding applications require the robot to follow a weave pattern (back and forth motion across the welding seam) during the operation. Other applications require a series of passes along the same path, but each pass must be slightly offset from the previous one to allow for the welding bead that was laid down in the previous pass. Both of these requirements are generally associated with large welding jobs where the amount of material to be added is greater than what can be applied normally during a single welding pass. Robots intended specifically for arc welding are often provided with features to facilitate the programming of weave patterns and multiple welding passes.

**5a.2.2 Sensors in Robotic Arc Welding**

At present, a wide variety of arc-welding sensors are either commercially available or under development in various research and development laboratories.

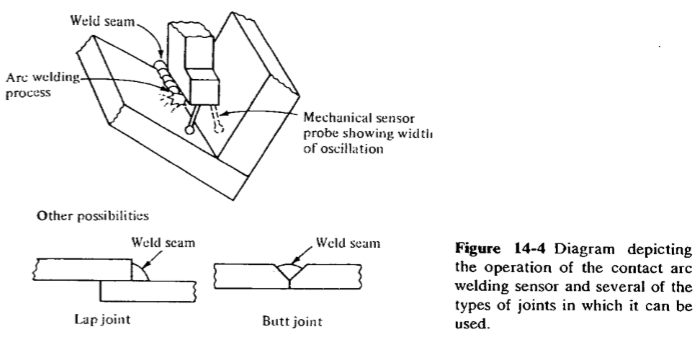
The robotic arc-welding sensor systems considered here are all designed to track the welding seam and provide information to the robot controller to help guide the welding path. The approaches used for this purpose divide into two basic categories: contact and noncontact sensors.

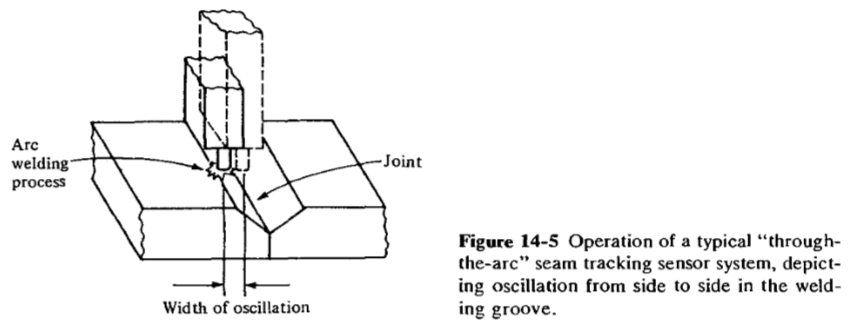
**5a.2.2.1 Contact arc-welding sensors**: Contact arc-welding sensors make use of a mechanical tactile probe (some of the probe systems would better be described as electromechanical) to touch the sides of the groove ahead of the welding torch and to feed back position data so that course corrections can be made by the robot controller. Some systems use a separate control unit designed to interpret the probe sensor measurements and transmit the data to the robot controller. To accomplish the position measurements, the probe must be oscillated from one side of the groove to the other by the sensor system. The nature of the operation of these sensor systems limits their application to certain weld geometries in which the side-to-side motion of the probe permits it to make contact with the edges or surfaces that are to be welded. Some of the weld geometries in this category include butt welds that have grooved joints, lap joints, and fillet welds. Figure 14-4 shows a diagram which depicts the operation of the contact arc welding sensor and several of the types of joints in which it can be used.

Another limitation of the contact arc weld sensor is that the probe must be maintained in the proper position ahead of the welding torch, and this makes these systems most effective on welds that are long and straight. These kinds of arc-welding applications do not make full use of most robots' capabilities for more complex path control

**5a.2.2.2 Noncontact arc-welding sensors:** The second basic type of sensor system used to track the welding seam uses no tactile measurements. A variety of sensor schemes have been explored in this category, but our discussions will concentrate on arc-sensing systems and vision-based systems since these are the approaches used more in today's commercial systems.

Arc-sensing systems (sometimes called "through-the-arc" systems) rely on measurements taken of the arc itself, in the form of either electric current (in constant-voltage welding) or voltage (in constant-current welding). In order to interpret these signals, they must be varied during the arc-welding process. This is accomplished by causing the arc to weave back and forth across the joint as it moves down the path. The side-to-side motion along the joint can be achieved by programming the robot to perform the weave pattern, or by means of a servo system that attaches to the robot wrist and determines the position of the torch, or by other mechanisms. The weaving motion permits the electrical signals to be interpreted in terms of vertical and cross-seam position of the torch. The controller performs an adaptive positioning of the torch as it moves forward along the joint centerline so that the proper path trajectory can be maintained. As irregular edges are encountered along the weld path, the control system compensates by regulating either the arc length (for constant-current systems) or the distance between the torch tip and the work surface (for constant-voltage systems). Operation of the typical through the-arc seam tracking system is illustrated in Fig. 14-5.





Vision-based systems represent a promising technology for tracking the seam in arc-welding operations. These systems utilize a vision camera mounted on the robot near the welding torch to view the weld path. In some cases the camera is an integral component of the welding head. Highly structured light is usually required for the camera sensors to function reliably.

There are two approaches used with vision sensors for arc welding: two-pass systems and single-pass systems. In both types, the robot must be programmed for the welding path before the operation begins.

In the two-pass systems, the vision camera takes a preliminary pass over the seam before the welding operation begins. As indicated above, the robot must be programmed for the particular seam path before either pass is taken. Then the two passes are taken automatically by the robot. In the first pass, light is projected onto the seam and the camera scans the joint at high speed (speeds up to 1 m/s are claimed), checking for deviations from the anticipated seam path. These deviations are analyzed by the controller and remembered for the second pass. During the second pass, in which the welding process is performed, the controller makes adjustments in the seam path to correct for the deviations detected in the first pass. The first pass requires only about l0 percent of the time for the welding pass, and the advantage gained by using two passes is that the vision system can see a clean view of the welding path on the preliminary scanning pass, absent of the smoke and intense brightness encountered during actual welding.

In the single-pass system, the vision camera is aimed at the welding seam just ahead of the torch. Deviations from the programmed seam location are detected and corrections are made in the weld path. The obvious advantage of the single-pass systems, compared to the two-pass systems, is that time is saved by eliminating the need for a second pass along the weld path. Another advantage is that the single-pass systems are able to compensate for thermal distortions in the weld path caused by the welding process.

Examples of commercial vision systems in the single-pass category are the Robovision II from Automatix Inc., and WeldYision from General Electric. In the Automatix system, the camera is focused about 4 cm in front of the weld. The observed image is analyzed to extract the location of the center of the seam, the seam width, and the distance of the seam from the camera. In the General Electric system, the vision sensor is incorporated into the design of the welding torch. The image observed by the camera includes the weld puddle and the seam ahead of it. By analyzing both the weld puddle and the seam, the controller is able to make adjustments in the process to automatically track the seam.

**5a.2.2.3 Advantages and Benefits of Robot Arc Welding**

A robot arc-welding cell for batch production has the potential for achieving a number of advantages over a similar manual operation. These advantages include the following:

1. Higher productivity
2. Improved safety and quality-of-work life
3. Greater quality of product
4. Process rationalization

The productivity of a manual arc-welding operation is characteristically quite low. The productivity is often measured by the "arc-on" time. This gives the proportion of time during the shift that the welding process is occurring, and therefore production is taking place. Typical values of arc-on time range between 10 and 30 percent. The lower value corresponds to one-of-a-kind welding jobs, and the higher value corresponds to batch type production. One of the reasons why the arc-on time is low in manual welding is the fatigue factor. The hand-eye, coordination required and the generally uncomfortable working environment tend to be tiring to the human welder and frequent rest periods must be taken. With robot welding cells for batch production, a 50 to 70 percent arc-on time can be realized. There are several factors that contribute to the increased arc-on time when robots are used in batch production. Certainly one factor is the elimination of the fatigue factor. Robots do not experience fatigue in the sense that human workers do. A robot can continue to operate during the entire shift without the need for periodic rest breaks. Another contributing factor is the presence of two parts positioners in the cell. The robot can be performing the welding operation at one positioner while the human operator is unloading the previous assembly and loading new components at the other positioner.

Improved safety and quality-of-work environment result from removing the human operator from an uncomfortable, fatiguing, and potentially dangerous work situation. As described above, the welding environment contains a number of serious hazards for human beings.

Greater product quality in robot arc welding results from the capability of the robot to perform the welding cycle with greater accuracy and repeatability than its human counterpart. This translates into a more consistent welding seam, one that is free of the start-and-stop buildup of filler metal in the seam that is characteristic of many welds accomplished by human welders.

The term process rationalization refers to the systematic organization of the work and the material flow in the factory. The design and installation of a robot welding cell forces the user company to consider such issues as the delivery of materials to the cell, the methods required to perform the welding process, the design of the fixtures, and the problems of production and inventory control related to the operation of the cell. Typically, these issues are not adequately addressed when the company relies on human welding stations.

**5a.3 SPRAY COATING**

Most products manufactured from metallic materials require some form of painted finish before delivery to the customer. The technology for applying these finishes varies in complexity from simple manual methods to highly sophisticated automatic techniques. We divide the common industrial coating methods into two categories:

1. Immersion and flow-coating methods
2. Spray-coating methods

Immersion and flow-coating methods are generally considered to be low-technology methods of applying paint to the product. Immersion involves simply dipping the part or product into a tank of liquid paint. When the object is removed, the excess paint drains back into the tank. The tanks used in the process can range in size from 1 or 2 gallons for small objects to thousands of gallons for large fabricated metal products. Closely related to immersion is the flow-painting method. Instead of dipping the parts into the tank, they are positioned above the tank and a stream of paint is directed to flow over the object. Both of these methods are relatively inefficient in terms of the amount of paint deposited onto the object. Although dipping and flow coating· are relatively simple processes, the methods for delivering the product to the painting operation may involve considerable mechanization. For example, conveyor systems are often used in high production to carry the parts down into the dipping tanks to apply the coating.

A more advanced immersion method is **electrodeposition**. This is a process in which a conductive object (the part or product) is given a negative electrical charge and dipped into a water suspension containing particles of paint. The paint particles are given a positive electrical charge, and consequently they are attracted to the negatively charged object (the cathode). The electrodeposition coating method is a highly sophisticated technique and requires close control over the process parameters (e.g., current, voltage, concentration of paint in suspension) in order to ensure the success of the operation. Its advantage is that it does not waste nearly as much paint as conventional immersion methods.

The second major category of industrial painting is spray coating. This method involves the use of spray guns to apply the paint or other coating to the object. Spray painting is typically accomplished by human workers who manually direct the spray at the object so as to cover the desired areas. The paint spray systems come in various designs, including conventional air spray, airless spray, and electrostatic spray. The conventional air spray uses compressed air mixed with the paint to atomize it into a high velocity stream. The stream of air and paint is directed through a nozzle at the object to be painted. The airless spray does not use compressed air. Instead the liquid paint flows under high fluid pressure through a nozzle. This causes the liquid to break up into fine droplets due to the sudden decrease in pressure in front of the nozzle.

The electrostatic spray method makes use of either conventional air spray or airless spray guns. The feature which distinguishes the electrostatic method is that the object to be sprayed is electrically grounded and the paint droplets are given a negative electrical charge to cause the paint to adhere to the object better. The spray-coating methods, when accomplished manually, result in many health hazards to the human operators. These hazards includes :

**Fumes and mist in the air.** These result naturally from the spraying operation. Not all of the paint droplets become attached to the surface of the object. Some remain suspended in the atmosphere of the spray painting booth. To protect the human operators, ventilation systems must be installed in the booth and protective clothing and breathing masks must be worn. Even with this protection, the environment is uncomfortable and sometimes toxic for humans.

**Noise from the nozzle.** The spray gun nozzle produces a loud shrill noise. Prolonged exposure by humans can result in hearing impairments.

**Fire hazards.** Flammable paint, atomized into a fine mist and mixed with air, can result in flash fires in the spray painting booths.

**Potential cancer hazards**. Certain of the ingredients used in modern paints are believed to be carcinogenic, with potentially unsafe health consequences to humans

**Robots in Spray Coating**

Because of these hazards to humans, the use of industrial robots has developed as an alternative means of performing spray-coating operations. Spray-coating operations to which robots have been applied include painting of car bodies, engines, and other components in the automotive industry, spraying of paint and sound absorbing coatings on appliances, application of porcelain coatings in bathroom fixtures, and spray staining of wood products. Some of the applications have consisted of a stand-alone robot spraying a stationary workpart that has been positioned in a paint booth by a human worker. However, these applications are generally less successful because they rely heavily on the human worker and the utilization of the robot is relatively low.

In most robot spray-coating applications, the robots are usually part of a system that includes a conveyor for presenting the parts to the robot, and a spray booth for shielding the spraying operation from the factory environment. Figure 14-7 illustrates a robot spray painting a part. When a conveyor-robot system is used, the operation of the robot and the conveyor must be closely synchronized. In the case of an intermittent (irregular/discontinuous) conveyor system, interlocks are used to coordinate the start and finish of the robot program with the movement of the conveyor. With a continuously moving conveyor, some form of baseline tracking system is required in order to synchronize the robot's motions with the movement of the conveyor. Another feature of many robot spray-coating applications is that the system must be designed to process a variety of part styles, each with its own unique configuration. This is usually accomplished by providing the workcell with a parts identification system. Once the part has been properly identified, the robot can then apply the correct spray cycle for that part

In general, the requirements of the robot for spray-coating applications are the following:

1. Continuous-path control. In order to emulate the smooth movement of a human spray paint operator, the robot must possess many degrees of freedom in its manipulator and it must have continuous path capability.
2. Hydraulic drive. Hydraulic drive is preferred over electric or pneumatic drive in spray-painting applications. In electric drive there is danger that a spark in the electric motor system may ignite the paint fumes in the spray booth environment. The motions generated in pneumatic drive are generally too jerky to be suitable for spray-coating applications.
3. Manual leadthrough programming. In most spray-coating applications, the most convenient method of teaching the robot involves leadthrough programming in which the robot arm is manually pulled through the desired motion pattern by a human operator who is skilled in the techniques of spray painting. During the programming procedure, a "teach arm" which is light and maneuverable, is often substituted for the actual robot arm, which tends to be heavy and difficult to manipulate smoothly.
4. Multiple program storage. The need for multiple program storage arises in paint production lines in which more than one part style are presented to the robot for spraying. The capability to quickly access the program for the current- part is a requirement for these lines. Either the robot itself must have sufficient memory for the programs required or it must be interfaced to the cell controller (computer or programmable controller) for random access to this memory capacity.

In robot spray-coating operations, the spray gun is the robot's end effector. Control over the operation of the spray gun system must be accomplished by the robot during program execution. In addition to on-off control over the spray gun nozzle, some of the important process variables that must be regulated during the spray cycle include paint How rate, fluid and/or air pressure, and atomization. These variables are regulated through the output interlock functions of the robot controller. The operation of the interlock functions is established during the programming procedure. Other parameters that must be controlled during the spray-coating process are related to the coating fluid (e.g., the paint). Viscosity, specific gravity, temperature, and other characteristics of the fluid must be maintained at consistent levels in order for the results of the finishing operation to be acceptable.

Another requirement for consistent quality in the finishing operation is that the spray gun must be periodically cleaned. This can be accomplished without significant loss of production time by programming a cleaning cycle into the workcell operation at regular intervals. The cleaning operation takes only a few seconds to complete and consists of the robot placing the spray nozzle under cleaning jets which spray solvent into the nozzle opening. Incorporating the cleaning operation into the work cycle should be planned to minimize the impact on the productive portion of the cycle.

**Benefits of Robot Spray Coating**

Use of robots to perform spray-finishing operations provide a number of important advantages. These advantages include:

1. Removal of operators from hazardous environment
2. Lower energy consumption
3. Consistency of finish
4. Reduced coating material usage
5. Greater productivity

Removing the human workers from the kinds of hazards which characterize the manual spray-finishing environment (i.e., fumes, fire hazards, etc.) is a significant health benefit of using robots. Also, because humans are not in the spray booth, the ventilation requirements are reduced below the levels needed when humans are present. Therefore less energy is needed to control the environment. Other advantages include better quality and fewer rejects. Because the robot performs the same spraying cycle on every workpart, the quality of the finishing job is more consistent compared to a human worker. As an added benefit, the amount of paint required to coat the parts is typically reduced by 10 to 50 percent when robots are used. These various features of the robot spray-coating cell result in substantial labor savings and improved productivity in the process.

The individual painting robots (GM calls them paint spray machines) are seven-axis manipulators that are operated under computer control. The supervisory computer communicates the correct work cycles to the individual machines on the line. Each spray paint machine is equipped with two spray guns especially designed by GM engineers for automatic operation in the cell. Each spray gun is provided with its own paint supply system so that one system can be in the process of being changed over (purged, cleaned, and refilled with the next paint) while the other system is in production. The parallel paint supply lines are contained inside the manipulator arm.

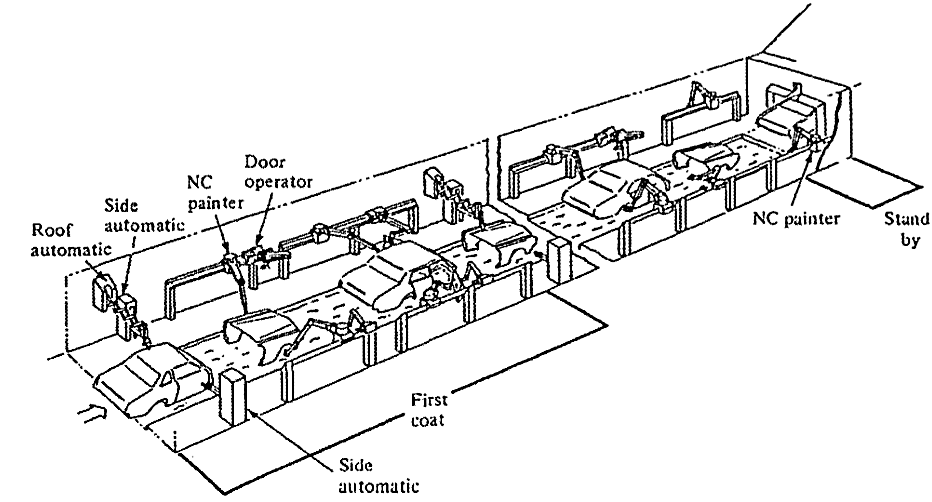


Figure 14-8 Typical configuration of GM paint spray cell. (Reprinted with permission from Akeel

**5a.4 OTHER PROCESSING OPERATIONS USING ROBOTS**

In addition to spot welding, arc welding, and spray coating, there are a number of other robot applications which utilize some form of specialized tool as the

Drilling, routing, and other machining operations

Grinding, polishing, dehurring, wire brushing, and similar operations

Riveting

Waterjet cutting

Laser drilling and cutting

We are excluding from this category applications in assembly, inspection, and nonmanufacturing operations which might employ a tool as end effector. From the preceding list, it can be seen that a typical end effector in this category is a powered spindle attached to the robot's wrist. The spindle is used to rotate a tool such as a drill or grinding wheel. The purpose of the robot is to position the rotating tool against a stationary workpart in order to accomplish the desired processing operation. In the other examples given in the above list (riveting, waterjet cutting, and laser operations), the end effector is not a powered spindle, but the job of the robot is still to position the tool relative to the part. Requirements of these applications vary, but one of the inherent disadvantages of robots in some of these operations is their relative lack of accuracy as compared to a regular machine tool. Small robots tend to be more accurate than large robots, but large robots are more likely to possess the strength and rigidity necessary to withstand the forces involved to hold the powered spindle against the part during the process.

**5a.5 ASSEMBLY AND ROBOTIC ASSEMBLY AUTOMATION**

The term assembly is defined here to mean the fitting together of two or more discrete parts to form a new subassembly. The process usually consists of the sequential addition of components to a base part or existing subassembly to create a more complex subassembly or a complete product. As such, assembly operations involve a considerable amount of handling and orienting of parts to mate them together properly. The difference between assembly tasks and other material-handling tasks is that value is added to the product through the assembly operation. Also, there are often interactions that take place between the two parts being assembled, between the gripper and the part, and between other elements of the workcell. When parts are fastened together (called parts joining), there are often additional interactions with the medium used to join the components (e.g., adhesive). All of these potential interactions can make assembly operations considerably more complex compared to the simpler task of moving a part from one location to another.

There are a variety of assembly processes used in industry today. These include mechanical fastening operations (using screws, nuts, bolts, rivets, swaging, etc.), welding, brazing, and bonding by adhesives. Some of these processes are more adaptable to automatic assembly. We will discuss the various assembly methods in more detail in a later section. It was mentioned in the introduction that assembly operations can be performed manually, or by high-speed automatic assembly machines, or by robots and other programmable systems. In addition, combinations of these techniques can be used in the design of an assembly system for a particular application.

In our coverage of the application of robotics to assembly, we will d1v1de the subject into three areas as suggested by the preceding discussion:

Parts presentation methods

Assembly tasks

Assembly cell designs

The following four sections will examine these three areas and some of the particular problems associated with them. We also present a discussion of a major development project devoted to the application of robots to assembly called the Adaptable-Programmable Assembly System (APAS). Our discussion of assembly will conclude with a section devoted to the topic of product design for automated assembly.

**5a.6 PARTS PRESENTATION METHODS**

In order for a robot to perform an assembly task, the part that is to be assembled must be presented to the robot. There are several ways to accomplish this presentation function, involving various levels of structure in the workplace:

Parts located within a specific area (parts not positioned or oriented)

Parts located at a known position (parts not oriented)

Parts located in a known position and orientation

In the first case, the robot is required to use some form of sensory input to guide it to the part location and to pick up the part. A vision system could be used as the sensory input system for this purpose. In the second case, the robot would know where to go to get the part, but would then have to solve the orientation problem. This might require the robot to perform an additional handling operation to orient the part. The third way of presenting the part to the robot (known position and orientation) is the most common method currently used, and is in fact the method used in automatic assembly that precedes the advent of robotics. This approach requires the least effort from the robot and sensor system, but it places the largest requirement on the parts feeding system.

**Bowl Feeders**

Bowl feeders are the most commonly used devices for feeding and orienting small parts in automated assembly operations. They are made by numerous companies and have been used to feed everything from delicate electronic parts to rugged metal castings. A bowl feeder consists of two main components: the bowl and the vibrating base. A track rising in a spiral up the sides of the bowl is used to deliver parts in the bottom of the bowl to an outlet point. This track is commonly located on the inside of the bowl. A typical bowl feeder is shown in Fig. 15-1.

The base of the bowl feeder is constructed of leaf springs and an oscillating electromagnet which causes the bowl and track to vibrate. The vibratory motion causes the parts to be driven up the spiral track until they reach the outlet point.

As the parts are driven up the track and approach the outlet point, they are oriented randomly and must be placed in the proper orientation for delivery out of the bowl feeder. This can be done by either of two methods, called selection and orientation. Selection (sometimes called passive orientation) involves taking parts that are not properly oriented and rejecting them from the track back into the bottom of the bowl, thus permitting parts that are properly oriented to pass through to the outlet point. Orientation involves taking parts that are not oriented properly and physically reorienting them as desired. Both methods are usually accomplished by means of a series of obstacles located along the track. These obstacles allow the parts to pass through only if they meet certain orientation criteria. In the case of part reorientation, obstacles or other mechanisms are used to physically change the orientation of the part as it moves along the track. Some of the techniques that are used in selection and orientation devices are pictured in Fig. 15-2. By providing a sufficient number of obstacles along the track, we can ensure that only parts that possess the desired orientation will successfully reach the outlet point.

Parts exiting the bowl feeder usually travel down a track or chute to some type of holding fixture. This fixture is located at an elevation which is below the outlet point of the bowl feeder so that gravity can be used to deliver the parts from the outlet to the holding fixture. The fixture is isolated from the vibration of the bowl feeder so the robot or other device that retrieves the part will not have to contend with the problems of a vibrating or moving target. The holding fixture maintains the desired position and orientation of the part until it is removed for the assembly operation. If a robot is to perform the assembly operation, the fixture must be designed with enough clearance for the robot gripper to grasp the part.

Another issue that must be addressed is the "back pressure" caused by the parts along the track leading to the holding fixture. Back pressure is the result of two forces: the force imparted on the parts by vibration in the track, and the force generated by the weight of all the parts in the track ahead of the fixture.

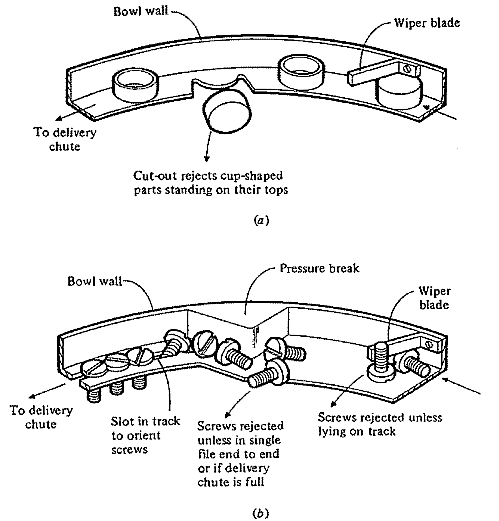


Figure 15-2 Part selection and orientation methods used in vibratory bowl feeders.

(a) Selection and orientation of cup-shaped parts

(b) selection and orientation of screws. (Reprinted with permission from Boothroyd and Redford)

If the back pressure is large enough, it can inhibit the robot from successfully removing the part from the fixture. The following relation describes when the robot will be unable to remove a part from the fixture:

*u*f *F*g < *u*p[*F*b + *n W*(sin *θ)*]

where *u*f = coefficient of friction between the gripper and the part

*F*g = gripping force of the gripper

*u*p = coefficient of friction between the parts in the track

*F*b = back pressure force due to track vibration

*n* = number of parts in the track leading up to the fixture

*W* = weight of each part

*θ*= angle that the track makes with the horizontal (the angle is assumed constant in the portion of the track containing parts leading to the fixture)

There are several ways to limit the back pressure at the holding fixture. The first is to turn off the vibration of the bowl whenever there are a large enough number of parts in the track that the back pressure would reach an undesirable level. On-off control of the bowl operation can easily be accomplished by providing a sensor in the track that detects the presence of the parts. A simple limit switch can be used for this purpose. When parts back up along the track to the point where the sensor is located, the bowl would be turned off. By properly positioning the sensor, the back pressure can be controlled to allow smooth pickup of parts by the robot from the holding fixture. A second way to reduce the back pressure is to make the angle of the track leading into the fixture relatively small. The risk here is that the parts will not properly slide down the track. A third way is to use an "escapement" device at the end of the track to individually place the parts into the holding fixture. This avoids the weight of many parts stacked against each other in the track. Figure 15-3 illustrates a number of escapement devices.

The first is to turn off the vibration of the bowl whenever there are a large enough number of parts in the track that the back pressure would reach an undesirable level. On-off control of the bowl operation can easily be accomplished by providing a sensor in the track that detects the presence of the parts. A simple limit switch can be used for this purpose. When parts back up along the track to the point where the sensor is located, the bowl would be turned off. By properly positioning the sensor, the back pressure can be controlled to allow smooth pickup of parts by the robot from the holding fixture. A second way to reduce the back pressure is to make the angle of the track leading into the fixture relatively small. The risk here is that the parts will not properly slide down the track. A third way is to use an "escapement" device at the end of the track to individually place the parts into the holding fixture. This avoids the weight of many parts stacked against each other in the track. Figure 15-3 illustrates a number of escapement devices.

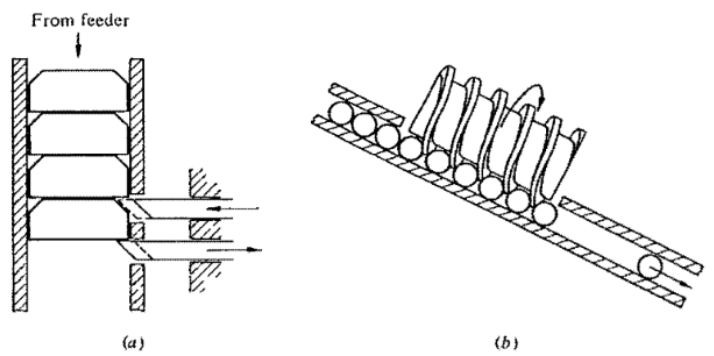


Figure 15-3 Several types of escapement devices used in automated assembly. (a) Linear motion escapement device for disk-shaped parts, (b) worm escapement device. (Reprinted with permission from Boothroyd and Redford)

**Magazine Feeders**

Bowl feeders are generally used to handle parts that are received at the workstation in bulk. Parts supplied in bulk are randomly oriented and one of the functions served by the bowl feeder is to deliver the parts to the track in proper orientation. An alternative to the use of a bowl feeder is to receive the parts at the workstation in a preoriented orderly fashion. The use of magazine feeders is one technique in which preoriented parts can be received at the workstation. Of course, this does not eliminate the problem of orienting and loading the parts; it simply transfers the problem away from the workstation. The most convenient way to load parts into the magazine in a proper orientation is to perform the loading operation as an integral element of the production process that makes the parts. What makes this possible is that the parts come out of the production process already oriented. For example, in a sheet metal stamping operation, the parts always come out of the press in the same way. It is therefore possible to load the stamped parts one on top of the next in the same orientation, into some kind of tube or other container. This container would constitute the magazine. The tube filled with parts could subsequently be attached to a mechanism (e.g., an escapement) designed to withdraw the parts and present them to the assembly workhead or the robot.

On the other hand, if the parts cannot be loaded into the magazine directly from the production operation and must be loaded instead manually, then the parts magazine loses much of its appeal. One of the disadvantages associated with the use of a parts magazine is that it typically holds fewer parts than a bowl feeder. Consequently, it must be replaced and refilled more frequently requiring a greater level of human attention at the workstation.

**Trays and Pallets**

Sometimes it is too expensive to use bowl feeders or parts magazines. In those cases, trays or pallets can be used. A specific advantage of using trays, pallets, and other similar storage containers is that they can be used for a variety of different part geometries. Bowl feeders and magazines must usually be custom engineered for a particular part configuration. However, there are certain conditions that generally must be satisfied in using trays and pallets in robotics. Namely, the parts must be located in known positions and orientations with respect to certain reference points on the device, usually the edges of the containers. This allows the trays to be registered correctly at the workstation and for the robot to be programmed to go to the known positions in the trays to retrieve the parts.

If the cycle time of the operation is relatively long, and the tray capacity is large, then the trays could be presented to the workstation by a human operator as required. If the cycle rate is fast, and a more automated operation is desirable, then some type of materials handling system must be devised to present the trays to the workstation automatically. In either case, an issue of great importance in the design of this type of container system (in robotics or any other form of automation) is that the containers must be positioned accurately at the workstation and the parts must be positioned precisely in the containers. If different part styles use the same basic container and material handling system, the information system supporting the operation must be sophisticated enough to handle the differences.

The alternative to the approach of precise part location is for the parts to be randomly oriented in the trays, and for the robot to perform some kind of "bin-picking" procedure in order to pick out the parts one at a time.

**5a.7 ASSEMBLY OPERATIONS**

Assembly operations can be divided into two basic categories: Parts mating and parts joining. In parts mating, two (or more) parts are brought into contact with each other. In parts joining, two (or more) parts are mated and then additional steps are taken to ensure that the parts will maintain their relationship with each other. In this section we discuss a number of assembly tasks that fall into these two categories, along with their implications for a robot system's capabilities.

Parts Mating The variety of parts mating operations include the following assembly situations:

**1. Peg-in-hole**. This operation involves the insertion of one part (the peg) into another part (the hole). It represents the most common assembly task. Peg-in-hole tasks can be divided into two types: the round peg-in-hole and the square peg-in-hole. The two types are illustrated in Fig. 15-4. With the round peg-in-hole, the robot needs only 5 degrees of freedom to insert the peg since there is no requirement to align the peg about its own axis. With the square peg-in-hole case, a full 6 degrees of freedom are needed in order to mate the corners of the square peg with the corners of the hole.

**2. Hole-on-peg**. This is a variation of the peg-in-hole task. Similar problems exist in defining the degrees of freedom needed to execute the mating of the two parts. A typical example of the hole-on-peg task would be the placement of a bearing or gear onto a shaft.

**3. Multiple peg-in-hole**. This is another variation on case 1 except that one part has multiple pegs and the other part has corresponding multiple holes. Consequently, •he assembly task always requires the ability of the assembly system to orient the parts in all directions. An example would be the assembly of a microelectronic chip module with multiple pins into a circuit card with corresponding holes, as illustrated in Fig. 15-5. This example represents a common assembly problem in the electronics industry.

4**. Stacking.** In this type of assembly, several components are placed one on top of the next, with no pins or other devices for locating the parts relative to each other. In a subsequent assembly operation, the group of parts would he joined together. An example of the stacking assembly operation would he a motor armature or a transformer in which the individual laminations are stacked.



Figure 15-4 Two types of peg-in-hole assembly tasks. (a) round peg-in-hole and (b) square peg-in-hole-orientation about z-axis required.

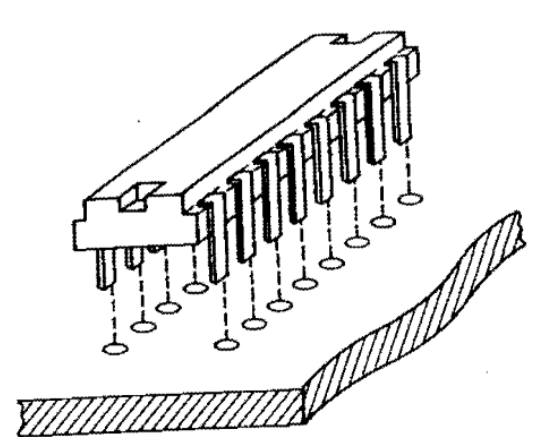
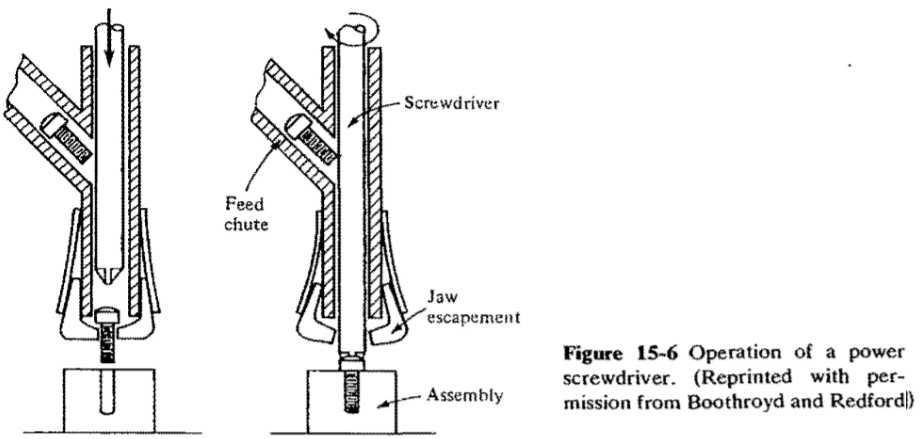


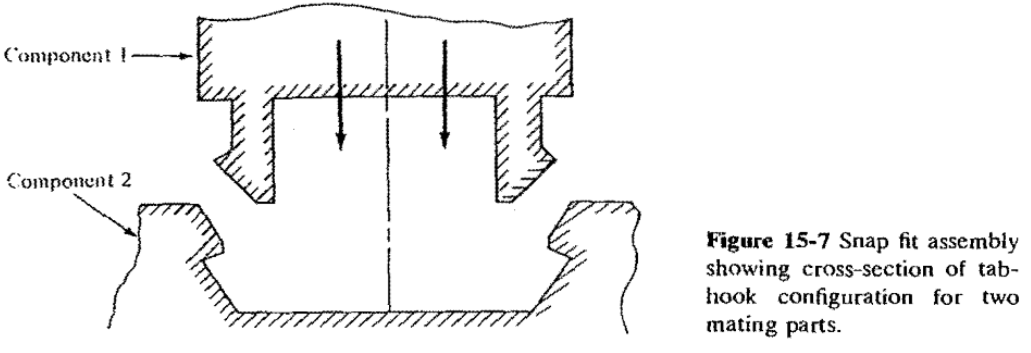
Figure 15-5 Multiple peg-in-hole assembly task: insertion of a semiconductor chip module into a circuit card

**Parts-Joining Tasks**

In parts joining, not only must the two (or more) components be mated, but also some type of fastening procedure is required to hold the parts together. The possible joining operations include the following:

1. Fastening screws. The use of screws is a very common method of joining parts together in manual assembly. Self-tapping screws are often used and this eliminates the need to perform the extra operation of tapping the threaded hole in the mating part. There are two ways in which a robot can perform the screw-fastening operation: it can drive the screw by advancing and simultaneously rotating its wrist, or it can manipulate a special end effector consisting of a power screwdriver. Power screwdrivers are available that not only drive the screws but also feed them automatically to the bit. Figure 15-6 shows a power screwdriver that can be attached to a robot wrist. Screw fastening without the aid of a power screwdriver turns out to be a relatively difficult task for a robot because of the complexity of the motions involved to rotate the screw and advance it into the hole at the same time. Also, when a screw is to be fastened into a threaded hole (in other words, a self-tapping screw is not used), there is the possibility for binding to occur between the screw threads if the mating hole threads are not properly aligned.
2. Retainers. Retainers can take a number of alternative configurations. They can be pins inserted through several parts in order to maintain the relationship among the parts. Another form of retainer is a ring that clamps onto one part to establish its relationship with another part. Common ring retainers are snap rings and C-rings.
3. Press fits. This is another variation of the peg-in-hole task except that the parts to be mated have an interference fit. This simply means that the peg is slightly larger than the hole into which it is to be inserted. Press-fitted parts can form a very strong assembly. However, a substantial force is required to accomplish the insertion operation. In most force-fit operations, the robot will not be able to provide the necessary force to press the parts together, and therefore the application will be designed so that the robot loads the parts into a power press which performs the actual press-fitting operation.
4. Snap fits. This joining technique has features of both the retainer and the press-fitting methods. A snap fit involves the joining of two parts in which the mating elements of the parts possess a temporary interference that only occurs during the joining process. When the parts are pressed together, one (or both) of the parts elastically deforms to accommodate the interference, then catches into the mating element of the part. The parts are usually designed so that a slight interference fit exists between the two parts even after they are snapped together. Figure 15-7 illustrates the snap fit assembly. This joining method turns out to be an ideal method for automatic assembly methods including robotics.
5. Welding and related joining methods. Continuous arc welding and spot welding are two common welding operations used to joint parts together. We have discussed these two joining techniques in the preceding chapter. In addition there are other similar joining techniques requiring heat energy that are used in assembly operations. These include brazing, soldering. And ultrasonic welding. All of these joining methods can he implemented by means of robots.
6. Adhesives. Glue and similar adhesives can be applied to join parts together by using a dispenser to lay down a bead of the adhesive along a defined path (for a robot, the motion cycle is similar to arc welding) or at a series of points (similar to spot welding). In most applications the adhesive dispenser is attached to the robot's wrist, while in other cases the robot manipulates the part and presents it to the dispenser.
7. Crimping. The term crimping, in the context of assembly, refers to the process of deforming a portion of one part (often a sheet metal part) to fasten it to another part. A common example of crimping is when an electrical connector is crimped (squeezed) onto a wire. To perform a crimping process, the robot requires a special tool or pressing device attached to its wrist. Staking operations and riveting operations are similar to crimping in that they involve deformation of one part to attach it to another. Staking usually refers to the use of metal tabs on one part that are bent over the joining part. Riveting involves specially designed fasteners (screws without threads) whose ends are flattened over the joining part.
8. Sewing. Although not typically considered as a robot application, this is a common joining technique for soft, flexible parts (e.g .. cloth, leather).





**5a.8 COMPLIANCE AND THE REMOTE CENTER COMPLIANCE (RCC) DEVICE**

Let us examine the peg-in-hole assembly task and consider the potential problems that are encountered during insertion. When a peg is inserted into a hole there are two possible positioning errors for the peg: a lateral position error and an angular error. These possibilities are illustrated in Fig. 15-8. When the parts are chamfered and there is a position error small enough to allow insertion to begin, it is still likely that an angular error will result during chamfer crossing as the peg rotates about the grip point at the top of the peg. Figure 15-9 shows how this problem can occur. The angular error allowable on the peg is a function of the clearance of the hole and the depth of the insertion. That is, the deeper the part is inserted into the hole, the less angular error can be tolerated; likewise, the smaller the clearance between the parts and the hole, the smaller the angular error. If the angular error is greater than the tolerable error, the parts will wedge into place much in the same way that a dresser drawer gets wedged when it becomes cocked in the drawer side. Conceptually, what must be done to perform a successful peg-in-hole insertion task is to correct for the lateral and angular errors during assembly. A common solution to the problem makes use of the Remote Center Compliance device.

The Remote Center Compliance device, or RCC, was developed during research on assembly at the Charles Stark Draper Laboratory in Cambridge, Massachusetts. Today, RCC products are commercially available and one such product is pictured in Fig. 15-10. The RCC device is typically mounted between the wrist of the robot and its gripper. Figure 15-11 shows a mechanical gripper attached to the RCC device.

The RCC device is capable of accommodating the lateral errors and angular errors encountered in an insertion operation and in other tasks requiring limited compliance. In the peg-in-hole insertion process, the RCC operates as though the part (peg) were being pulled into the hole by the tip, rather than being pushed from the top. The action is accomplished as illustrated in Fig. 15-12.7 Parts (a) and (b) of the figure show the accommodation of the lateral forces. Suppose there is a lateral error in position between the peg and the hole as shown in part (a). Because of the chamfer on the hole, the error will cause a lateral force on the peg. This force causes the RCC to translate the peg so that it can be inserted into the hole. Next, consider the possibility of an angular error. Suppose the axis of the hole is not parallel with the axis of the peg. The peg will enter the hole (assuming the necessary lateral compliance occurs), but its leading edge will contact one side of the hole, while the edge of the hole will contact the other side of the shaft as shown m Fig. 15-12(c). This will cause a moment on the peg. The RCC will accommodate the moment by means of a rotation about the compliant center as illustrated in part (d) of the figure.

RCCs are typically constructed using elastomer springs rather than the mechanical linkages shown in Fig. 15-12. This has resulted in designs that are simple, small, and lightweight. The parameters to be considered when selecting a remote center compliance device include the following:

**Remote center distance.** This is also called the center of compliance dimension. It is the distance between the RCC bottom surface and the compliant center of the RCC device. The compliant center (also called the elastic center) is the point about which the forces acting on the object being inserted are minimum. The remote center distance should be selected on the basis of the length of the part and the gripper.

**Axial force capacity**. This is the maximum force in the axial direction which the RCC device is designed to withstand and still function properly.

**Compressive stiffness**. This is also called the axial stiffness. It is the force per unit distance or spring constant required to compress the RCC device in the direction of insertion. Generally, the compressive stiffness is relatively high to allow for press fitting of parts together.

**Lateral stiffness**. This is the spring constant relating to the force required to deflect the RCC laterally (perpendicular to the direction of insertion). This parameter should be determined according to the stiffness of the robot and the delicacy of the parts being assembled.

**Angular stiffness**. This is also called the cocking stiffness. It is the rotational spring constant that relates to the force required to rotate the part about the elastic center by a certain amount.

**Torsional stiffness**. This is the torsional spring constant which relates to the moment required to rotate the part about the axis of insertion. This parameter becomes important when the insertion task requires orientation relative to the axis of insertion.

Other parameters to be considered in the specification of the remote center compliance device are the maximum allowable lateral and angular errors. These errors are generally determined by the relative size of the product and by its design (e.g., design of the chamfers). They must be sufficiently large to compensate for errors in the workcell due to parts, robot, and fixturing.

A second approach to provide compliance would be to measure the forces and moments encountered by the part and to servo the robot to compensate for these forces. Also, the Instrumented Remote Center Compliance (IRCC) device is a possible approach to this type of problem. The IRCC is an RCC device that has been instrumented to measure deflections. These deflections provide an indication of the forces and moments being applied to the wrist. Whereas most force sensors are very rigid and deflect very little under load, the IRCC is compliant in certain directions. This permits high-speed part insertion owing to the compliance of the RCC while allowing monitoring and data collection of forces during operation of the system.

**5a.9 ASSEMBLY SYSTEM CONFIGURATIONS**

There are two basic configurations of assembly systems, a single workstation, and a series of workstations (an assembly line). Combinations of these two basic types are also possible. For example, it is sometimes advantageous to design a series configuration with certain stations in parallel. The following subsections will cover the various possibilities.

**Single-Workstation Assembly**

In this configuration all of the parts which are required to complete the desired assembly are presented to the operator or robot at a single workstation. All of the parts mating and joining tasks for the assembly are accomplished at the single workstation. In manual assembly, this configuration is generally used for low-volume products (e.g., custom-engineered machinery). In robotic assembly, the conditions warranting the use of this configuration are different from those for manual assembly. A single-station robotic assembly system would typically be used for low- and medium-volume work in which there were a limited number of assembly tasks and parts to e handled. This means that the product is of low to medium complexity.

**5b) Autonomous Mobile Robots: Introduction, Planning & Navigation:**

**5b.1 Introduction**

**5b.2 Basic control scheme for mobile robots**

**5b.2.1 Perception**

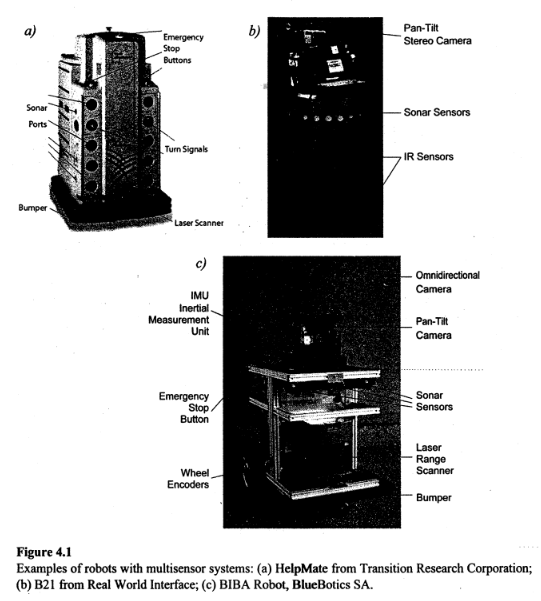
One of the most important tasks of an autonomous system of any kind is to acquire knowledge about its environment. This is done by taking measurements using various sensors and then extracting meaningful information from those measurements,

In this chapter we present the most common sensors used in mobile robots and then discuss strategies for extracting information from the sensors.

**Sensors for Mobile Robots**

A wide variety of sensors is used in mobile robots (figure 4.1). Some sensors are used to measure simple values such as the internal temperature of a robot's electronics or the rotational speed of the motors. Other more sophisticated sensors can be used to acquire information about the robot's environment or even to measure directly a robot's global position.

Here we focus primarily on sensors used to extract information about the robot's environment. Because a mobile robot moves around, it will frequently encounter unforeseen environmental characteristics, and therefore such sensing is particularly critical. We begin with a functional classification· of sensors. Then, after presenting basic tools for describing a sensor's performance, we proceed to describe selected sensors in detail.



1. **Sensor classification**

We classify sensors using two important functional axes: *proprioceptive/exteroceptive* and *passive/active.*

*Proprioceptive* sensors measure values interna l to the system (robot), for example, motor speed, wheel load, robot arm joint angles, and battery voltage.

*Exteroceptive* sensors acquire information from the robot's environment, for example, distance measurements, light intensity, and sound amplitude. Hence exteroceptive sensor measurements are interpreted by the robot in order to extract meaningful environmental features.

*Passive* sensors measure ambient environmental energy entering the sensor. Examples of passive sensors include temperature probes, microphones, and CCD or CMOS cameras.

*Active* sensors emit energy into the environment, then measure the environmental reaction. Because active sensors can manage. more controlled interactions with the environment, they often achieve superior performance; However, active sensing introduces several risks: the outbound energy may affect the very characteristics that the sensor is attempting to' measure. Furthermore, an active sensor may suffer from interference between its signal and those beyond its control. For example, signals emitted by other nearby robots, or similar sensors on the same robot, may influence the resulting measurements. Examples of active sensors include wheel quadrature encoders, ultrasonic sensors, and laser range finders.

Table 4.1 provides a classification of the most useful sensors for mobile robot applications. The most interesting sensors are discussed in this chapter. The sensor classes in table 4.1 are arranged in ascending order of complexity and descending order of technological maturity. Tactile sensors and proprioceptive sensors are critical to virtually all mobile robots and are well understood and easily implemented. Commercial quadrature encoders, for example, may be purchased as part of a gear-motor assembly used in a mobile robot. At the other extreme, visual interpretation by means of one or more CCD/CMOS cameras provides a broad array of potential functionalities, from obstacle avoidance and localization to human face recognition. However, commercially available sensor units that provide visual functionalities are only now beginning to emerge.

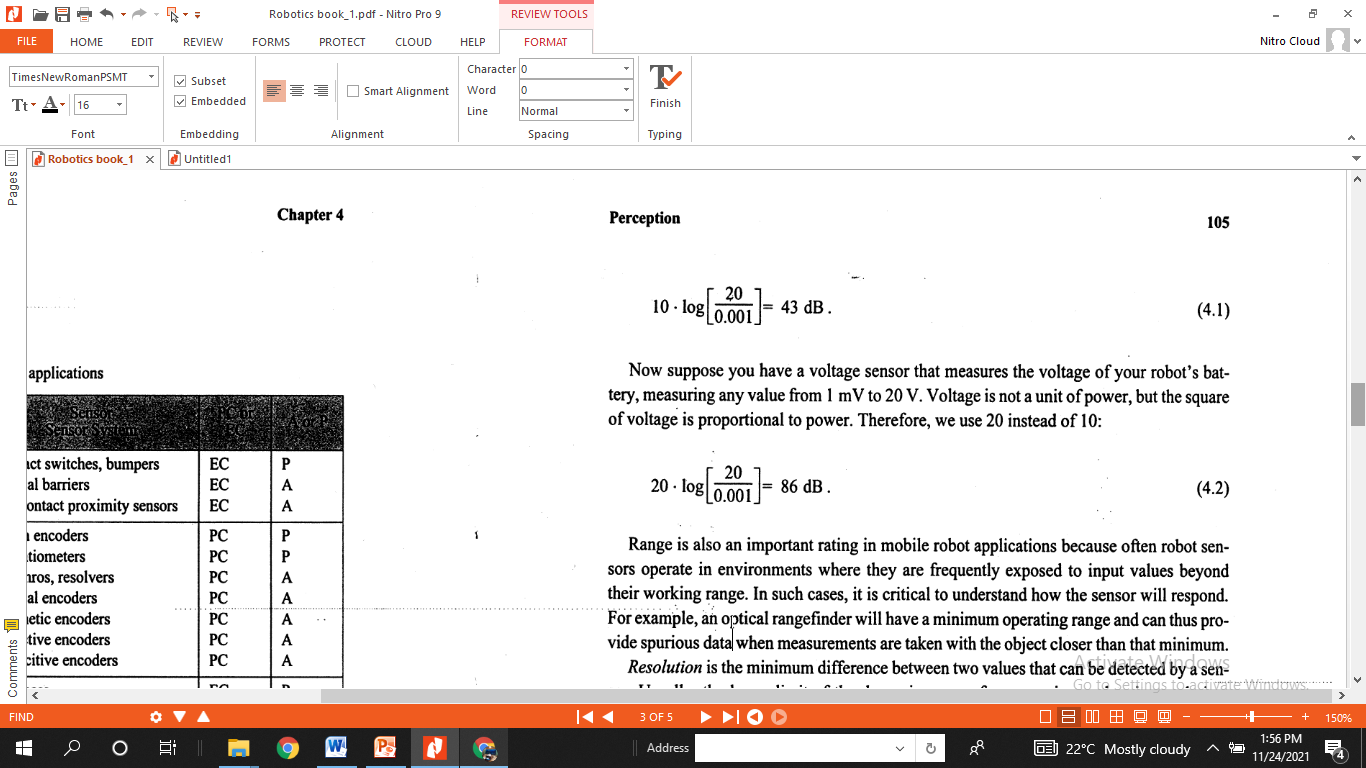
1. **Characterizing sensor performance**

The sensors we describe in this chapter vary greatly in their performance characteristics. Some sensors provide extreme accuracy in well-controlled laboratory settings but are overcome with error when subjected to real-world environmental variations. Other sensors provide narrow, high-precision data in a wide variety of settings.

**Basic sensor response ratings**

A number of sensor characteristics can be rated quantitatively in a laboratory setting. Such performance ratings will necessarily be best-case scenarios when the sensor is placed on real-world robot, but are nevertheless useful.

**Dynamic range** is used to measure the spread between the lower and upper limits of input values to the sensor while maintaining normal sensor operation. Formally, the dynamic range is the ratio of the maximum input value to the minimum measurable input value. Because this raw ratio can be unwieldy, it is usually measured in decibels, which are computed as ten times the common logarithm of the dynamic range. However, there is potential confusion in the calculation of decibels, which are meant to measure the ratio between powers, such as watts or horsepower. Suppose your sensor measures motor current and can register values from a minimum of l mA to 20 mA. The dynamic range of this current sensor is defined as



**Range** is also an important rating in mobile robot applications because often robot sensors operate in environments where they are frequently exposed to input values beyond their working range. In such cases, it is critical to understand how the sensor will respond. For example, art optical rangefinder will have a minimum operating range and can thus provide spurious data when measurements are taken with the object closer than that minimum.

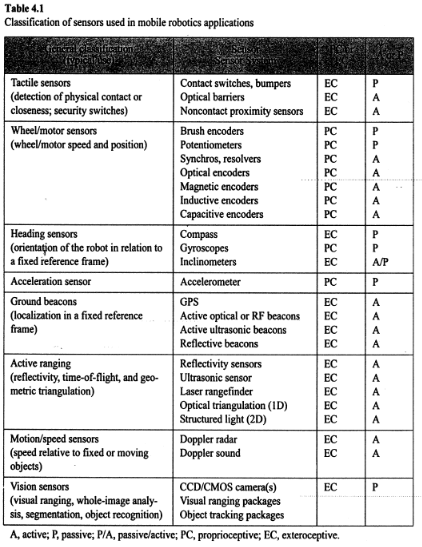
**Resolution**is the minimum difference between two values that can be detected by a sensor. Usually, the lower limit of the dynamic range of a sensor is equal to its resolution.

However, in the case of digital sensors, this is not necessarily so. For example, suppose that you have a sensor that measures voltage, performs an analog-to-digital (A/D) conversion, and outputs the converted value as an 8-bit number linearly corresponding to between 0 and 5V . If this sensor is truly linear, then it has 28-1 total output values, or a resolution of

5 V(255) = 20 mV

**Linearity**is an important measure governing the behavior of the sensor's output signal as the input signal varies. A linear response indicates that if two inputs *x* and *y* result in the two outputs *f(x)* and *f(y),* then for any values *a* and *b, f(ax+ by)* = *af(x)* + *bf(y).* This means that a plot of the sensor's input/output response is simply a straight line.

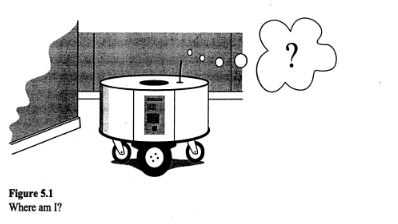
**Bandwidth**or **frequency**is used to measure the speed with which a sensor can provide a stream of readings. Formally, the number of measurements per second is defined as the sensor's frequency in **hertz***.* Because of the dynamics of moving through their environment, mobile robots often are limited in maximum speed by the bandwidth of their obstacle detection sensors. Thus, increasing the bandwidth of ranging and vision sensors has been a high priority goal in the robotics community.

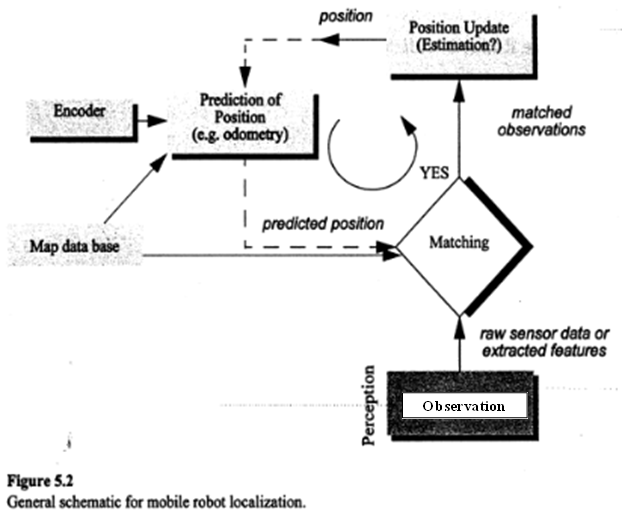


**5b.2.2 Localization**

**Navigation** is one of the most challenging competences required of a mobile robot. Success in navigation requires success at the four building blocks of navigation: ***perception***(the robot must interpret its sensors to extract meaningful data); ***localization***(the robot must determine its position in the environment, Figure 5.1); ***cognition***(the robot must decide how to act to achieve its goals); and ***motion******control***(the robot must modulate its motor outputs to achieve the desired trajectory).

Of these four components (Figure 5.2), localization has received the greatest research attention in the past decade, and as 'a result, significant advances have been made on this front.





**The Challenge of Localization: Noise and Aliasing**

If one could attach an accurate GPS (global positioning system) sensor to a mobile robot, much of the localization problem would be obviated. The GPS would inform the robot of its exact position, indoors and outdoors, so that the answer to the question, "Where am I?" would always be immediately available. Unfortunately, such a sensor is not currently practical. The existing GPS network provides accuracy to within several meters, which is unacceptable for localizing human-scale mobile robots as well as miniature mobile robots such as desk robots and the body-navigating nanorobots of the future. Furthermore, GPS technologies cannot function indoors or in obstructed areas and are thus limited in their workspace.

But, looking beyond the limitations of GPS, localization implies more than knowing one's absolute position in the Earth's reference frame. Consider a robot that is interacting with humans. This robot may need to identify its absolute position, but its relative position with respect to target humans is equally important. Its localization task can include identifying humans using its sensor array, then computing its relative position to the humans. Furthermore, during the ***cognition***step a robot will select a strategy for achieving its goals. If it intends to reach a particular location, then localization may not be enough. The robot may need to acquire or build an environmental model, a ***map****,* that aids it in planning a path to the goal. Once again, localization means more than simply determining an absolute pose in space; it means building a map, then identifying the robot's position relative to that map.

Clearly, the robot's sensors and effectors play an integral role in all these forms of localization. It is because of the inaccuracy and incompleteness of these sensors and effectors that localization poses difficult challenges. This section identifies important aspects of this sensor and effector suboptimality.

**To Localize or Not to Localize: Localization-Based Navigation Versus Programmed Solutions**

Figure 5.6 depicts a standard indoor environment that a mobile robot navigates. Suppose that the mobile robot in question must deliver messages between two specific rooms in this environment: rooms *A* and *B.* In creating a navigation system it is clear that the mobile robot will need sensors and a motion control system. Sensors are absolutely required to avoid hitting moving obstacles such as humans, and some motion control system is required so that the robot can deliberately move.

It i s less evident, however, whether or not this mobile robot will require a *localization system.* Localization may seem mandatory in order to navigate successfully between the two rooms. It is through localizing on a map, after all, that the robot can hope to recover its position and detect when it has arrived at the goal location. It is true that, at the least, the robot must have a way of detecting the goal location. However, explicit localization with reference to a map is not the only strategy that qualifies as a goal detector.

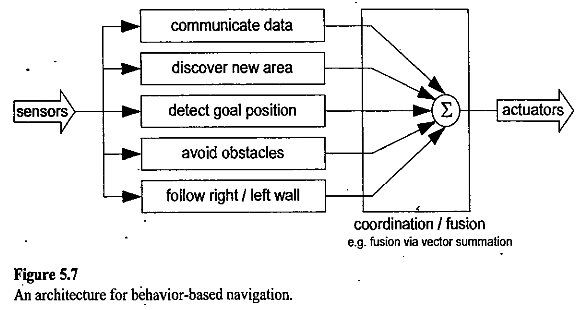
An alternative, espoused by the behavior-based community, suggests that, since sensors and effectors are noisy and information limited, one should avoid creating a geometric map for localization. Instead, this community suggests designing sets of behaviors that together result in the desired robot motion. Fundamentally, this approach avoids explicit reasoning about localization and position, and thus generally avoids explicit path planning as well.

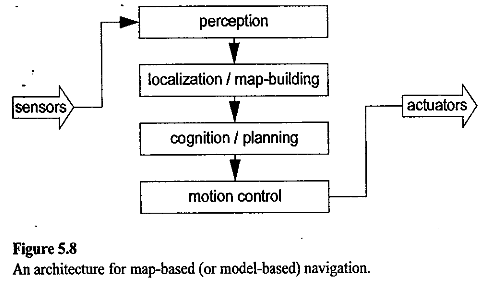
This technique is based on a belief that there exists a procedural solution to the particular navigation problem at hand. For example, in Figure 5.6, the behavioralist approach to navigating from room *A* to room *B* might be to design a left-wall following behavior and a detector for room *B* that is triggered by some unique queue in room *B,* such as the color of the carpet. Then the robot can reach room *B* by engaging the left-wall follower with the room *B* detector as the termination condition for the program.

The architecture of this solution to a specific navigation problem is shown in Figure 5.7. The key advantage of this method is that, when possible, it may be implemented very quickly for a single environment with a small number of goal positions. It suffers from some disadvantages, however. First, the method does not directly scale to other environments or to larger environments. Often, the navigation code is location-specific, and the same degree of coding and debugging is required to move the robot to a new environment.

Second, the underlying procedures, such as *left-wall-follow,* must be carefully designed to produce the desired behavior. This task may be time-consuming and is heavily dependent on the specific robot hardware and environmental characteristics.

Third, a behavior-based system may have multiple active behaviors at any one time. Even when individual behaviors are tuned to optimize performance, this fusion and rapid switching between multiple behaviors can negate that fine-tuning. Often, the addition of each new incremental behavior forces the robot designer to retune all of the existing behaviors again to ensure that the new interactions with the freshly introduced behavior are all stable.





In contrast to the behavior-based approach, the map-based approach includes both *localization* and *cognition* modules (see Figure 5.8). In map-based navigation, the robot explicitly attempts to localize by collecting sensor data, then updating some belief about its position with respect to a map of the environment. The key advantages of the map-based approach for navigation are as follows:

* + - The explicit, map-based concept of position makes the system's belief about position transparently available to the human operators.
    - The existence of the map itself represents a medium for communication between human and robot: the human can simply give the robot a new map if the robot goes to a new environment.
    - The map, if created by the robot, can be used by humans as well, achieving two uses.

The map-based approach will require more up-front development effort to create a navigating mobile robot. The hope is that the development effort results in an architecture that can successfully map and navigate a variety of environments, thereby amortizing the upfront design cost over time.

Of course the key risk of the map-based approach is that an internal representation, rather than the real world itself, is being constructed and *trusted* by the robot. If that model diverges from reality (i.e. , if the map is wrong), then the robot's behavior may be undesirable, even if the raw sensor values of the robot are only transiently incorrect.

In the remainder of this chapter, we focus on a discussion of map-based approaches and, specifically, the localization component of these techniques. These approaches are particularly appropriate for study given their significant recent successes in enabling mobile robots to navigate a variety of environments, from academic research buildings, to factory floors, and to museums around the world.

**5b.2.3 Path planning & Motion control**

Even before the advent of affordable mobile robots, the field of path planning was heavily studied because of its applications in the area of industrial manipulator robotics. Interestingly, the path-planning problem for a manipulator with, for instance, six degrees of freedom is far more complex than that of a differential-drive robot operating in a flat environment. Therefore, although we can take inspiration from the techniques invented for manipulation, the path-planning algorithms used by mobile robots tend to be simpler approximations owing to the greatly reduced degrees of freedom. Furthermore, industrial robots often operate at the fastest possible speed because of the economic impact of high throughput on a factory line. So, the dynamics and not just the kinematics of their motions are significant, further complicating path planning and execution. In contrast, a number of mobile robots operate at such low speeds that dynamics are rarely considered during path planning, further simplifying the mobile robot instantiation of the problem.