1988

NEAR EAST UNIVERSTY

DEPARTMENT OF
ELECTRIC AND ELECTRONIC
ENGINNEERING

E&E 400
GRADUATION PROJECT

SUBJECT : ROBOTICS CONTROL

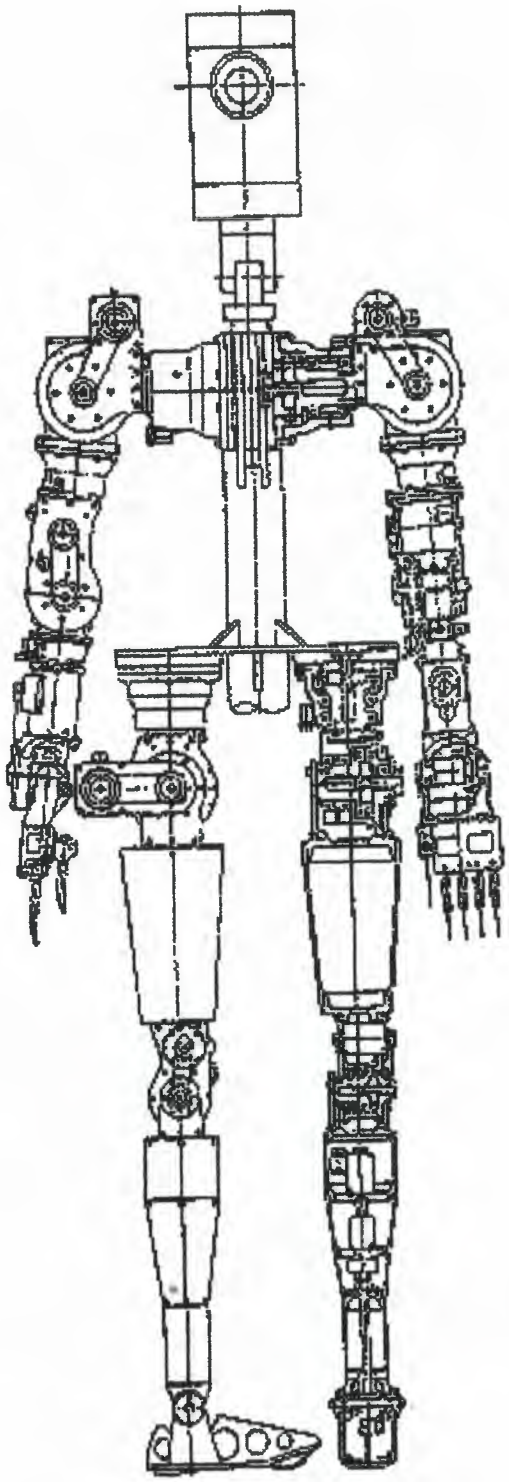
INSTRUC. : PROF. DR. KHALIL ISMAILOV
SUBMITTED BY: Z. ABIDIN ISIK 93368
TARIK OVACIKLI 93853

JUNE 1998

12.06.98

Acknowledgment

We would like to express our gratitude to our Prof. Dr. Khalil Ismailov who helped our a lot to accomplish this work. We are also obliged to all other people who helped our to achieve , what we have achieved. We pay homage to our parents, who helped our with necessary feedback to do the best.



CONTENTS

CHAPTER 1 - ROBOTIC CONTROL SYSTEMS

CHAPTER 2 - DRIVE METHODS

CHAPTER 3 - ROBOTIC SENSORY DEVICES

CHAPTER 4 - CONTROL

CHAPTER 5 - COMPUTER HARDWARE AND SOFTWARE FOR ROBOTS

CHAPTER 6 - INDUSTRIAL APPLICATION EXAMPLES

INTRODUCTION

When one first hear the word robot, the image that probably comes to mind is that of a mobile biped that is both humanoid in structure and capable of independent actions. The George Lucas (Star Wars) creation C3-P0 is an example. Such devices are, unfortunately, still relegated to science fiction novels and motion pictures. The truth is that we simply do not, at this time know how to create machines with this degree of intelligence and mobility. In fact it is likely that unless there is a significant breakthrough in a number of areas such as artificial intelligence, computers and power storage devices most of the readers of this project will never see a robot that has anywhere near the capability of C3-P0. For this reason in this project we choose to discuss industrial robots almost exclusively.

As we will see in subsequent chapters an industrial robot is a complex Electro-mechanical device that brings together a large number of disciplines in what could be termed a polygamous relationship. Despite the fact that this chapter will be relatively descriptive, nontechnical introduction to the subject, a variety of important questions must still be answered. For example, what types of mechanisms can be classified as industrial robots and what types cannot? Also since the robots about which we will be speaking will almost always be used in a manufacturing environment, what are the economic justifications for utilizing such devices for a given task? Moreover, what applications are appropriately handled by robots both now and in the future? Finally, what types of robots are available in the market place today? Besides these important questions that we will attempt to answer later this chapter there are other considerations worthy of our attention.

No study of robots would be complete without some discussion of their sociological consequences. It is clear that they will have a significant impact on the manufacturing environment. But will workers in industries utilizing robots be displaced from their jobs or will more jobs actually be created? How will these workers accept this new form of technology and what can management and/or the government do to eases any problems that result from the introduction of this type of automation into the workplace? We will attempt to answer this extremely difficult question, but the reader is warned not to expect easy answer: the authors' unfortunately do not have magical solutions to the inevitable social problems.

We now trace the origin of the modern industrial robot and indicate how these devices have evolved in relationship to other technological devwelopments.

CHAPTER 1

ROBOTIC CONTROL SYSTEMS

1.1 ROBOT CONROL SYSTEM

Industrial robots are frequently used in industry to improve productivity. The robot can handle monotonous jobs as well as complex jobs without errors in operation. The robot can work in an environment intolerable to human operators. For example, it can work in extreme temperatures (both high and low) or in a high- or low- pressure environment or under water or in space. There are special robots for fire fighting, underwater exploration, and space exploration, among many others.

The industrial robot must handle mechanical parts that have particular shapes and weights. Hence, it must have at least an arm, a wrist, and a hand. It must have sufficient power to perform the task and the capability for at least limited mobility. In fact, some robots of today are able to move freely by themselves in a limited space in a factory.

The industrial robot must have some sensory devices. In low-level robots, microswitches are installed in the arms as sensory devices. The robot first touches an object and then through the microswitches, confirms the existence of the object in space and proceeds in the next step to grasp it.

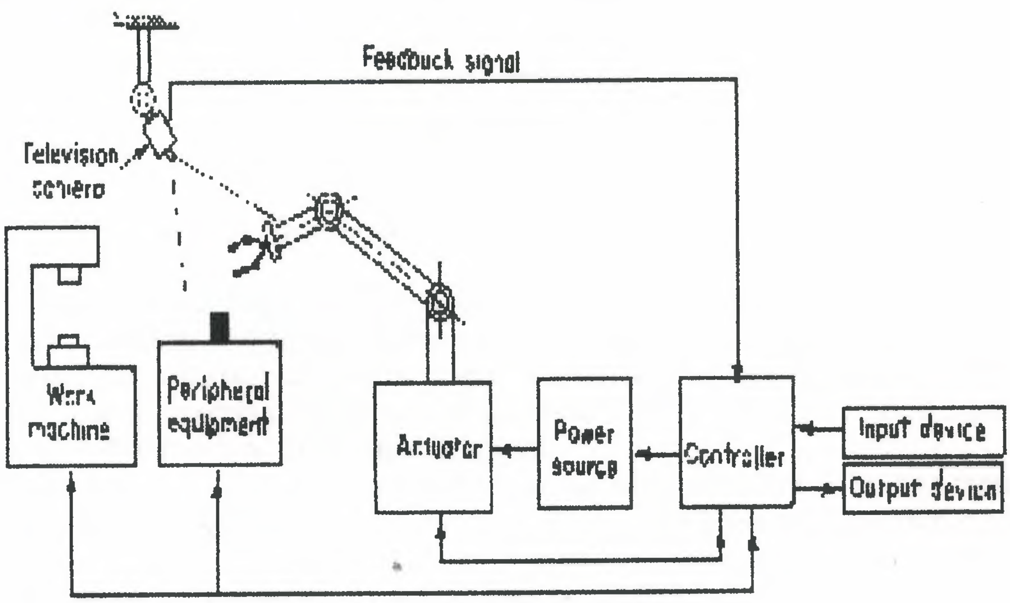


Figure 1-1 Robot using a pattern recognition process.

In a high-level robot, an optical means (such as a television system) is used to scan the background of the object. It recognizes the pattern and determines the presence and orientation of the object A computer is necessary to process signals in the pattern-recognition process (see Figure 1.1). In some applications, the computerized robot recognizes the presence and orientation of each mechanical part by a pattern recognition process that consists of reading the code numbers attached to it. Then the robot picks up the part and moves it to an appropriate place for assembling, and there it assembles several parts into a component. A well-programmed digital computer acts as a controller.

1.1.1 Robot arm control system.

Figure 1.2 shows a schematic diagram for a simplified version of the robot arm control system. The diagram shows a straight-line motion control of the arm. A straight-line motion is a I-degree-of-freedom motion. The actual robot arm has 3 degrees of freedom (up-and-down motion, forward-and-backward motion, and left-and-right motion). The wrist attached to the end of the arm also has 3 degrees of freedom, and the hand

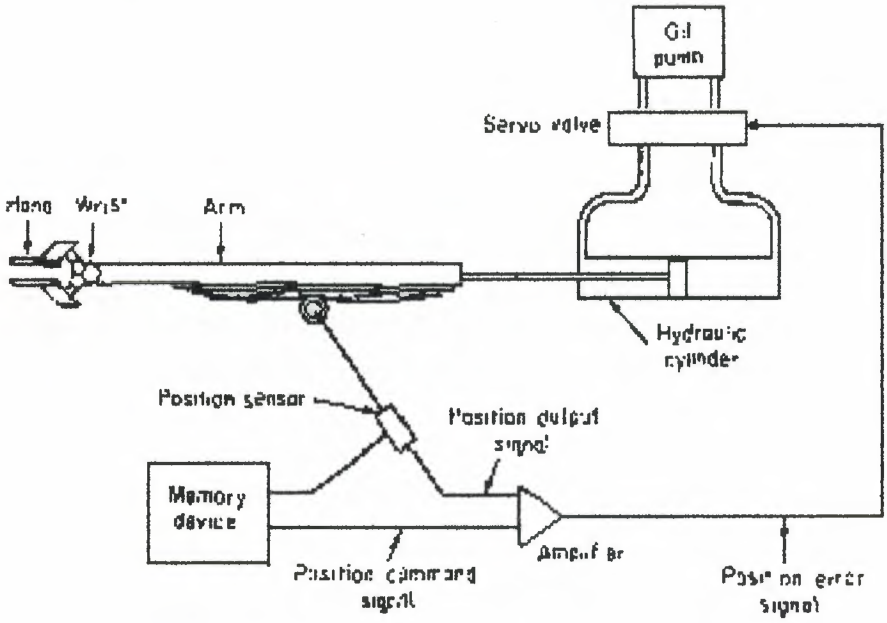


Figure 1-2 Robot arm control system.

has 1 degree of freedom (grasp motion). Altogether the robot arm system has 7 degrees of freedom. Additional degrees of freedom are required if the robot body must move on a plane. In general, robot hands may be interchangeable parts: a different type of grasping device can be attached to the wrist to serve as a hand to handle each different type of mechanical object.

A servo system is used to position the arm and wrist. Since the robot arm motion frequently requires speed and power, hydraulic pressure or pneumatic pressure is used as the source of power. For medium power requirements, dc motors may be used. And for small power requirements, step motors may be used.

For control of sequential motions, command signals are stored on magnetic disks. In high-level robot systems, the playback mode of control is frequently used. In this mode, a human operator first "teaches" the robot the desired sequence of movements by working some mechanism attached to the arm; the computer in the robot memorizes the desired sequential movements. Then, from the second time on, the robot repeats faithfully the sequence of movements.

1.1.2 Robot hand grasping force control system

Figure 1.3 shows a schematic diagram for a grasping force control system using a force-sensing device and a slip-sensing device. If the grasping force is too small, the robot hand will drop the mechanical object, and if it is too large, the hand may damage or crush the object. In the system shown, the grasping force is preset at a moderate level before the hand touches the mechanical object. The hand picks up and raises the object with the preset grasping force. If there is a slip in the raising motion, it will be observed by the slip-sensing device and a signal will be sent back to the controller, which will then increase the grasping force. In this way, a reasonable grasping force can be realized that prevents slipping but does not damage the mechanical object.

1.1.3 Numerical control systems

Numerical control is a method of controlling the motions of machine components by use of numbers. In numerical control the motion of a workhead may be controlled by binary information contained on a disk.

The system shown in Figure 1-5 works as follows: A magnetic disk is prepared in binary form representing the desired part P. To start the system, the disk is fed through the reader. The frequency-modulated input pulse signal is compared with the feedback pulse signal. The controller carnes out mathematical operations on the difference in the pulse signals. The digital-to-analog converter converts the controller output pulse into an analog signal that represents a certain magnitude of voltage, which in, turn

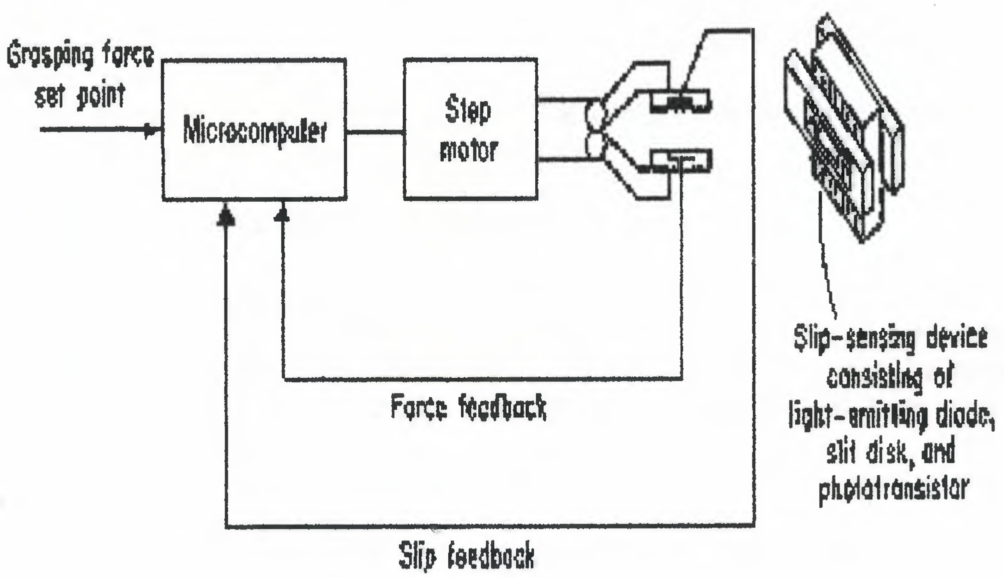


Figure 1.3 Robot hand grasping force control system.

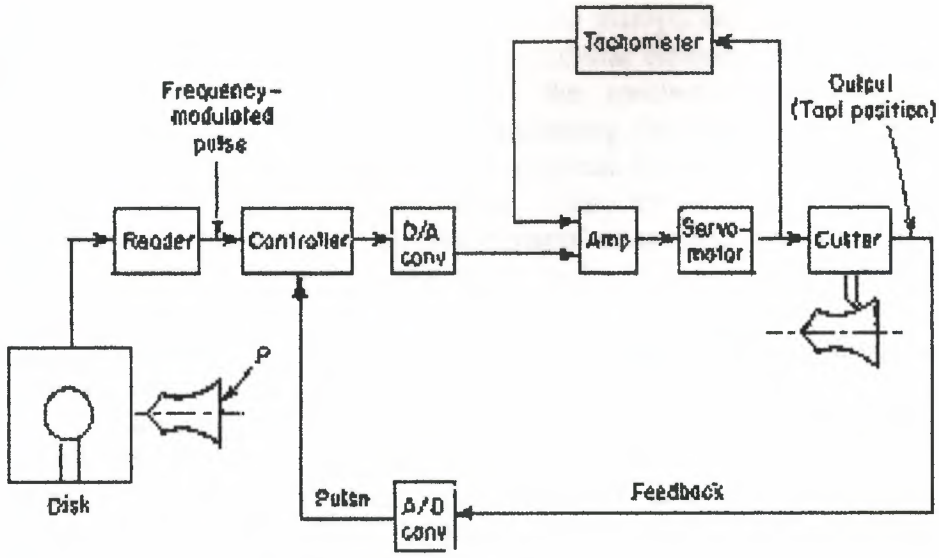


Figure 1-4 Numerical control of a machine.

causes the servomotor to rotate. The position of the cutterhead is controlled according to the input to the setvomotor. The transducer attached to the cutterhead converts the motion into an electrical signal, which is converted, to the pulse signal by the analog-to-digital converter. Then this signal is compared with the input pulse signal. If there is any difference between these two, the controller sends a signal to the servomotor to reduce it, as stated earlier.

An advantage of numerical control is that complex parts can be produced with uniform tolerances at the maximum milling speed.

1.2 MAJOR COMPONENT OF A ROBOT

Although the mechanical, electrical, and computational structure of robots can vary considerably, most have the following four major components in common: (1) a manipulator or arm (the "mechanical unit"), (2) one or more sensors, (3) a controller (the "brain"), and (4) a power supply. Let us briefly describe each of these in turn.

1. The Manipulator. This is a collection of mechanical linkages connected by joints to form an open-loop kinematics chain. Also include are gears, coupling devices, and so on. The manipulator is capable of movement in various directions and is said to do "the work" of the robot. In fact, the term "robot" and "manipulator" are often used interchangeably, although, strictly speaking this is not correct.

Generally, joints of a manipulator fall into one of two classes. The first is reevaluate, produces pure rotary motion. Consequently, the term rotary joint is often used to describe it. The second, prismatic, produces pure linear or translation motion and as a result, is often referred to as a linear joint. Each of the joints of a robot defines a joint axis about or along which the particular link either rotates or slides (translates). Every joint axis defines a degree of freedom (DOF). so that the total number of DOFs is equal to the number of joints. Many robot have six DOFs, three for positioning (in space) and three for orientation, although. It is possible to have as few as two and as many as eight degrees of freedom.

Regardless of its mechanical configuration, the manipulator defined by the joint-link structure generally contains three main structural elements: the arm. the wrist, and the hand (or end effector). Besides the mechanical components, most manipulators also contain the devices for producing the movement of the various mechanical members. These devices are referred to as actuators and may be pneumatic, hydraulic, or electrical in nature. They are invariably coupled to the various mechanical links or joints (axes) of the arm either directly or indirectly. In the latter case, gears, belts, chains, harmonic drives, or lead screws can be used.

2. Sensory Devices. These elements inform the robot controller about the status of the manipulator. This can be done continuously or only at the end of a desired motion. For example, in some robots, the sensors provide instantaneous position, velocity, and possibly acceleration information about the individual links that can be fed back to the control unit to produce the proper control of the mechanical system. More simply, the controller can be informed only when the individual links of the manipulator have reached their preprogrammed final or end positions. Regardless of how it is used, the information provided by the sensors can be either analog, digital, or a combination.

Sensors used in modern robots can be divided into two general classes:

* Nonvisual
* Visual

The first group includes limit switches (e.g., proximity, photoelectric, or mechanical), position sensors (e.g., optical encoders, potentiometers, or resolvers), velocity sensors (e.g., tachometers), or force and tactile sensors (for overload protection, path following, calibration, part recognition, or assembly work). The second group consists of vidicon, charge-coupled device (CCD), or charge injection device (CID) TV cameras coupled to appropriate image-detection hardware. They are used for tracking, object recognition, or object grasping.

3. The Controller. Robot controllers generally perform three functions:

* They initiate and terminate the motion of the individual components of the manipulator in a desired sequence and at specified points.
* They store position and sequence data in their memory.
* They permit the robot to be interfaced to the "outside" world via sensors mounted in the area where work is being performed (i.e., the workstation).

To carry out these tasks, controllers must perform the necessary arithmetic computations for determining the correct manipulator path, speed, and position. They must also send signals to the joint-actuating devices (via interfaces) and utilize the information provided by the robot's sensors. Finally, they must permit communication between peripheral devices and the manipulator.

Robot controllers usually fall into one of the following classes:

* Simple step sequencer
* Pneumatic logic system
* Electronic sequencer

* Microcomputer
* Minicomputer

The first three are generally used in less expensive, open-loop-control robots. The microcomputer-based robotic controller is the most commonly used device in the servo-controlled robots. Minicomputer controllers are not common because they are currently not as cost-effective as microcomputers.

4. The Power Conversion Unit. The purpose of this part of the robot is to provide the necessary energy to the manipulator's actuators. It can take the form of a power amplifier in the case of servomotor-actuated systems, or it can be a remote compressor when pneumatic or hydraulic devices are used.

Up to this point, we have been concerned primarily with the classification of robots according to their geometry or control scheme. In addition, we have briefly described in the current section the major components that one expects to find in any industrial robotic device. The remaining portions of the chapter are devoted to the reasons and justifications for using robots, the potential consequences of placing robots in the workplace, and finally, some current and possible future applications of these devices.

1.3 CLASSIFICATION BY CONTROL METHOD

As mentioned above, the second method of classification looks at the technique used to control the various axes of the robot. The two general classes are (1) non-servo controlled, and (2) servo controlled. We now consider each one separately.

1.3.1 Non-servo-controlled robots

From a control standpoint, the non-servo-controlled or limited-sequence robot is the simplest type. Other names often used to describe such a manipulator are end point robot, pick-and-place robot, or bang-bang robot. Regardless of mechanical configuration or use, the major characteristic of such devices is that their axes remain in motion until the limits of travel (or "end stops") for each are reached. Thus only two positions for the individual axes are assumed. The non-servo nature of the control implies that once the manipulator has begun to move it will continue to do so until the appropriate end stop is reached. There will be no monitoring (via external sensors) of the motion at any intermediate points. As such, one refers to this class of robot as being controlled in an open-loop manner.

"Programming" a limited-sequence robot is accomplished by setting a desired sequence of moves and adjusting the end stops for each axis accordingly. The manipulator "brain" consists of a controller/sequencer. The "sequencer" portion is generally a motor-driven rotary device (similar to the "timer motor" found in certain home appliances) with a number of electrical contacts. Unlike the timer motor on a washing machine, for example, a series of jumper plugs is used and permits the appropriate contacts to be enabled by the sequencer in the desired order. Each such enabled contact will cause power to be switched to an axis actuator (e.g., pneumatic or hydraulic valve/piston arrangement) by the controller portion. The energized axis will continue to move until the "programmed" end stop is reached. This information is then used to cause the sequencer to index to the next step in its "program." It is important for the reader to understand that this is the only time that information is "fed back" to the sequencer.

A typical operating sequence for a hydraulic or pneumatic non-servo-controlled robot is as follows:

1. A program "start" causes the controller/sequencer to signal control valves on the manipulator's actuators.
2. This causes the appropriate valves to open, thereby permitting air or oil to flow into the corresponding pistons (actuators) and the member(s) of the manipulator begin to move.
3. These valves remain open and the members continue to move until they are physically restrained from doing so by coming into contact with appropriately placed end stops.
4. Limit switches, generally located on the end stop assemblies, signal the end of travel to the controller/sequencer, which commands the open valves to close.
5. The sequencer now indexes to the next step and the controller again outputs signals to actuator valves, thereby causing other members of the manipulator to move. Alternatively, signals can be sent to an external device such as a "gripper," causing it to open or close as desired.
6. The process is repeated until all steps in the sequence are executed.

Other attributes and/or capabilities worthy of mention for this class of robot are as follows:

* Conditional modification of the programmed sequence is possible if some type of external sensor is employed. For example, if a simple optical interrupter is used, it may be possible to have the manipulator pause in its sequence until a peripheral tool (e.g., a punch press) has cleared the work envelope. Robots having this ability normally can perform one program.

* Open loop or non-servo control is often used in smaller robots because of its low cost and simplicity. An example of such a device is the Seiko PN-100.

* It is possible to have a number of "intermediate" stops for each of the axes. This allows the manipulator to be programmed for more complex paths and permits a limited degree of path control.

* Although the controller normally applies full power to an axis that is selected by the sequencer and turns this power off only when the limit stop is reached, it is possible to achieve a degree of deceleration into the stop by using shock absorbers or appropriate valving at the end stops. This results in less stress on the components of the manipulator and on the part being moved.

Even though limited sequence robots can be configured in a variety of ways (e.g., Cartesian, cylindrical, etc.), a number of characteristics are common to all such devices. In particular:

* They are relatively high-speed machines because of the small size of the arm and the full power applied to the axis actuators.

* They are low cost and easy to maintain and operate. Also, they are extremely reliable devices.

* They have a repeatability of about ±0.01 inch. That is, they have the ability to return to the same point within ± 10 mils. (A few small pneumatically actuated robots such as the Seiko PN-100 advertise repeatability of about ±0.5 mil).

* This class of robot has limited flexibility with respect to positioning and programming. Thus although more than one axis can be moved at a time, it is generally not possible to cause a tool held at the end of the manipulator to move in a straight line (except if the desired line happens to coincide with one of the robot axes). Also, coordinated motion cannot be produced whereby the axes reach the endpoint of the desired motion at the same instant.

1.3.2 Servo-controlled robots

Servo-controlled robots are normally subdivided into either continuous-path or point-to point devices. In either case, however information about the position and velocity (and perhaps-other physical quantities) is continuously monitored and fed back to the control system associated with each of the joints of the robot. Consequently, each axis loop is "closed." Use of closed-loop control permits the manipulator's members to be commanded to move and stop anywhere within the limits of travel for the individual axes. (The reader should contrast this with the turn-servo-controlled machines described above, where only axis extremes could be programmed.) In addition, it is possible to control the velocity, acceleration, deceleration, and jerk (i.e. the time derivative of acceleration) for the various axes between the endpoints. Manipulator vibration can, as a consequence, be reduced significantly. Besides the above, servo-controlled robots also have the following additional features and/or attributes:

* A larger memory capacity than in non-servo-controlled devices. This implies that they are able to store more positions (or points in space) and hence that the motions can be significantly more complex and smoother. It also means that more than one program can be created and stored, thereby permitting the robot to be used in a variety of applications with a minimum of downtime required for the changeover.

* The end of the manipulator can be moved in any one of three different classes of motion: point-to-point (where the endpoints of the motion are important but the path connecting them is not), straight line (where it is important to cause a specified location on the manipulator, often referred to as the tool point, to move from the initial point to the final one in a linear fashion (in three-dimensional space)], or continuous path (where points along the path are connected so that the instantaneous position and either its spatial or time (i.e., velocity) derivative are continuous]. Note that not every servo-controlled robot is capable of performing straight-line and/or continuous-path motion. Also, it may not always be possible to maintain a constant path velocity if all points along a desired path have been taught at the same speed.

* Within the limits imposed by the mechanical components, adjusting the gains of appropriate amplifiers in the servo loops can vary positional accuracy.

* Joint actuators are usually either hydraulic valve piston arrangements or servomotors, although until about 1985 there did exist at least one commercially available robot that used pneumatic servos.

* Programming is generally done in what is referred to as teach mode. The manipulator is manually moved to a sequence of desired points. The coordinates of each of these are stored in the robot's (semiconductor) memory. Some of the more sophisticated systems actually have a specialized computer language that permits

these stored points to be utilized in a variety of motions, paths, orientations, and so on. An example of such a language is Unimation's VAL.

* It is possible to program each axis to move to almost any point along its entire range of travel. Consequently, this affords the user with a great deal of flexibility in the type of motions that are possible. Moreover, "coordinated motion" can be achieved whereby two or more joints move simultaneously so that the end of the manipulator is capable of tracing out an extremely complex path. It is important to understand that such coordination among the robot axes is normally done "automatically" under mini- or microcomputer control.

* It is possible to permit branching operations whereby alternative actions are taken by the manipulator based on data obtained from external sensors. For example, it might be possible for the robot to repeat a particular set of moves if a part did not appear at a workstation because of a faulty feed mechanism. This capability arises from the extensive use of microprocessors in the robot controller.

* Because servo-controlled robots generally have considerably more complex control, computer, and mechanical structures than non-servo-controlled devices, they may be more expensive and somewhat less reliable. Their great flexibility makes them extremely attractive and cost-effective in a large number of applications.

With these features in mind, the following represents a typical operating sequence for a general servo-controlled robot (it is assumed that the desired points have been taught and stored in memory prior to running the program):

1. At the beginning of the program, the actual position of all of the manipulator joints is obtained from appropriately mounted sensors. The desired (or command) position information is sent out to the individual axes from a master computer.

2. For each joint, the actual and desired positions are compared and an "error" signal is formed. This is used to drive the individual joint actuators.

3. As a result, the members of the robotic manipulator move. Position, velocity, and any other physical parameter of the motion are monitored or estimated (again utilizing appropriate sensors), and this information is used to automatically modify the error signals accordingly.

4. When the error signals for all the individual axes are zero, the members stop moving and the manipulator is "home" .

5. The master computer then sends out the next taught point, and steps 1 through 4 are repeated. This process continues until all of the desired points (or actions, e.g., opening or closing of a gripper) have been reached (or per- formed).

Although most servo-controlled robots behave in the general manner described above, there are certain features that are specific to the point-to-point and the continuous-path robots. We next consider these briefly.

1.3.3 Point-to-point servo-controlled robots

Point-to-point robots are widely used for moving parts from one location to another and also for handling various types of tools. Although they can perform all of the tasks of the pick-and-place robot, they are far more versatile because of their ability to be multiply programmed and also because of their program storage capability. A typical point-to-point application might be the unloading or loading of a pallet of parts. In the former case, the robot would be taught (i.e.., programmed) each of the n locations on the pallet. (Alternatively, the first point and the x and y offsets for each of the other pallet locations would be taught.) It would then move to the first of these taught points, pick up the part, move to a position above the conveyor, and place the part onto the conveyor. The manipulator would repeat the action for each of the remaining (n - 1) locations on the pallet. Such an application, while possible with a simple, nonservo pick-and-place device, would probably require a servo-driven x-y table that would actually move the pallet relative to the fixed pickup point. An example of loading a pallet is shown in Figure 1.3.1.

For the class of closed-loop control robot being considered here, only the initial and final points are taught. The path used to connect the two points is unimportant and is, therefore, not programmed by the user. (The computer calculates the actual path of the manipulator.) More sophisticated point-to-point robots permit straight-line or piecewise-linear motions. Others also permit the velocity of the individual joints to be a continuous function of time and also to be changed by the user, that is, the speed with which the device performs a desired task is user selectable. If no changes in what the robot will do are expected, the initially taught points can be stored in a permanent or read-only memory (ROM). Alternatively, combination of temporary or random accesses memory (RAM)* ROM is used for teaching new points and storing the old ones.

In general, these robots have a working range and load capacity that is quite high, that is, loads of up to 500 Ib and reaches of 10 to 11 ft are possible. The most often use hydraulic actuators, although recently, the trend has been toward servomotor-actuated systems. Examples of this type
of robot are those made be ASEA Cincinnati Milacron (the T3), Unimation (the Unimate 2000).

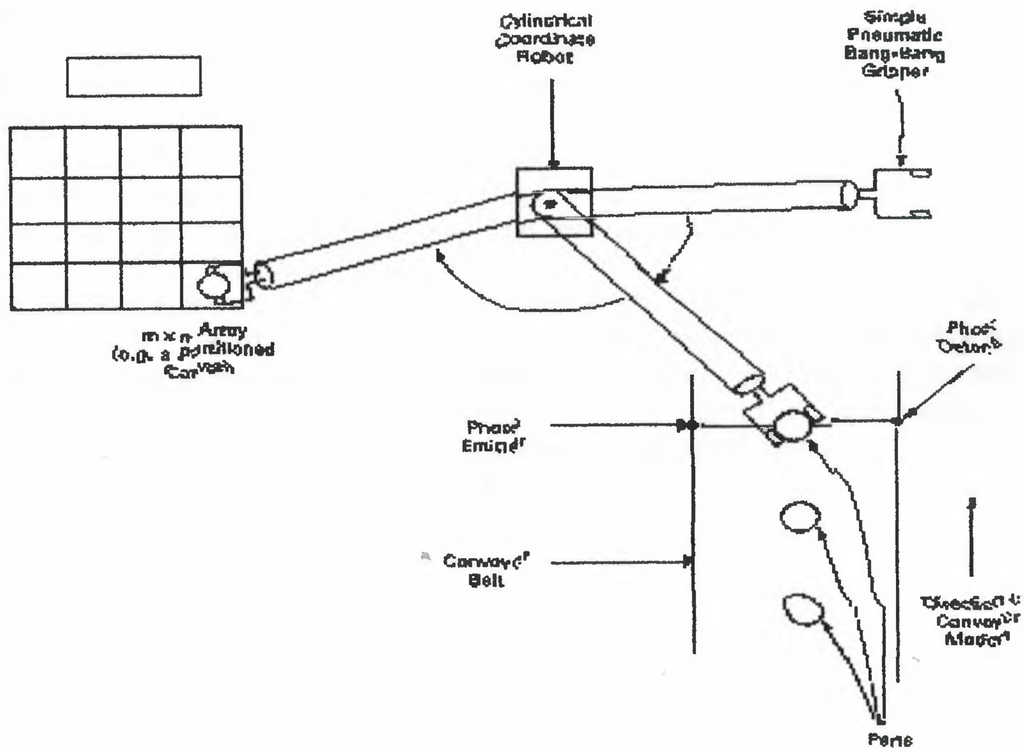


Figure 1.3.2. Use of a robot in a palletizing operation. When a part moving on a conveyor interrupts the light beam from a photo emitter, the controller commands the robot to acquire the part. This part is then moved to and placed in one of the (empty) locations in a partitioned carton. This process is repeated until all such locations are filled, at which time the carton is removed from the loading station, an empty one replaces it, and the operation is repeated.

1.3.4 Continuous-path servo-controlled robots

Many applications do not require that the manipulator have a long reach be able to carry a large load. In particular, there is an entire class of application where it is most important to follow a complex path through space and possibly to have the end of the arm move at high speeds. Examples of these applications include spray painting, polishing, grinding, and arc welding. In all instances, the tool carried by the manipulator is fairly light but the required motion to perform the task may be quite complex. A continuous-path (CP) robot is usually called for in these cases.

Although points must still be taught prior to executing a program, the method of teaching is usually quite different from that used for the point-to-point servo-controlled robot. Unlike the procedure described above, points are not recorded manually in the CP robot. What happens is that in the teach mode, an automatic sampling routine is activated which can record points (and/or velocity information) at a rate of 60 to 80 times a second for approximately 2 minutes. An operator simply moves the tool over the desired path with the sampler running. The sampling rate is usually high enough so that when the recorded points are "played back" (i.e., the program is run), extremely smooth motion results. It is clear that a large memory is required since as many as 9600 points may be recorded in the 2-minute period. To facilitate the accurate recording of complex paths (e.g., in arc welding applications), the tool can be moved over the desired path during the teaching phase at a slow speed. Playback, however, will be independent of the recorded speed, so rapid and accurate curve tracing is possible. An example of a CP robot is that produced by DeVilbiss.

It is important to understand that, in general, CP robots can be used for only a limited number of tasks and are often single-task devices (e.g., spray painting and welding).

* On the other hand, point-to-point robots sometimes have the ability to perform CP motion, although the method of teaching the large number of points is not nearly as convenient since each point must still be recorded manually. Examples of such devices are the PUMA and the Maker 110.

It should be apparent from the above that even though there are two general methods of robot classification, there are still a large number of different robot types. Despite this fact, many points of commonality among this diverse mechanism exist.

* As of this writing, the lack of flexibility is more than offset by the increase in productivity achieved with CP robots when used to perform these tasks

CHAPTER 2

DRIVE METHODS

An important distinguishing feature of robots is the drive method chosen. Drives are the source of motive power that drives the links to a desired position. Usually power is applied at the joints, either directly or through cables, gearing, belts, or other means. There are four major types of drive in use today: hydraulic, pneumatic, DC electric motors, and electric stepper motors. Each will be covered in detail in a separate section. Next, the criteria for selection of drive types are outlined, and the advantages and disadvantages of each type for different applications are discussed.

2.1 HYDRAULIC DRIVES

Hydraulic drives in robots use oil under high pressure as the working fluid. Drives may be open loop or closed loop and may be either linear or rotary. A simplified diagram of a hydraulic cylinder controlled by a servo valve is given in Figure 2.1.

Open loop drives can go from point to point accurately but cannot be controlled to stop at points in between. They come under the classification sometimes called bang-bang because they move from one position to another and 'bang" up against a stop at the other end of their movement.

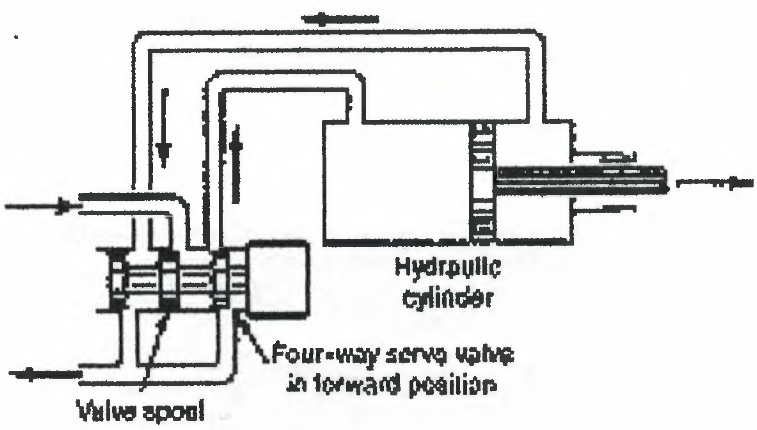


Figure 2.1 Simplified diagram of a hydraulic cylinder and servo valve used for control.

2.1.1 Linear Drive Cylinders

Linear hydraulic cylinders controlled by solenoid-operated valves are the simplest and least expensive open-loop hydraulic drives. It should be noted that a controlled stop is possible in linear cylinder operation by providing a controlled orifice to slow down the flow of oil when the piston approaches the end of its motion. Also, manual valves control many pieces of equipment so that controlled movement is possible. In this case, the human operator becomes part of the closed-loop system, and it is no longer an open-loop system. Lift masts on forklift trucks and construction equipment such as backhoes are of this type.

Hydraulic pistons mounted in large-diameter cylinders are in expensive to make and are capable of exerting large forces in a small space. Oil pressures of the order of 2,000 pounds per square inch are used. Therefore a piston one-inch in diameter can exert a force of 1,570 pounds.

Both linear and rotary hydraulic drives operate due to the pressure of oil-against a piston in the linear cylinder or against a rotary vane in the rotary hydraulic motor. Oil is admitted to one end of the piston by the action of the hydraulic valve, as illustrated in Figure 2.1. Valves in open-loop systems are opened and controlled by magnetic solenoids; closed-loop systems use electromagnetic servo valves or manually operated valves.

2.1.2 Rotary Actuators

A well-designed rotary actuator is shown in Figure 2.2. The actuator housing is made from solid aluminum alloy, and the actuator rotor is machined from steel. Pressure seals and dust seals are used to contain the hydraulic fluid and to protect bearings, respectively. Oil is allowed to enter through the fluid ports, under control of the electrohydraulic valve, and impinges on the rotating vane fixed to the actuator rotor, causing the rotor assembly to move. A stationary vane blocks the direct oil path and forces oil to operate the rotating vane. Position information is obtained by both a potentiometer and a resolver operated by an antibacklash gear train. Coarse position is supplied by the potentiometer signal, while the resolver measures fine position. In this way, the lower accuracy and larger range of the potentiometer augment the high accuracy and limited range of the resolver. Of course, the overall accuracy is no greater than that of the gear train that drives the potentiometer and resolve.

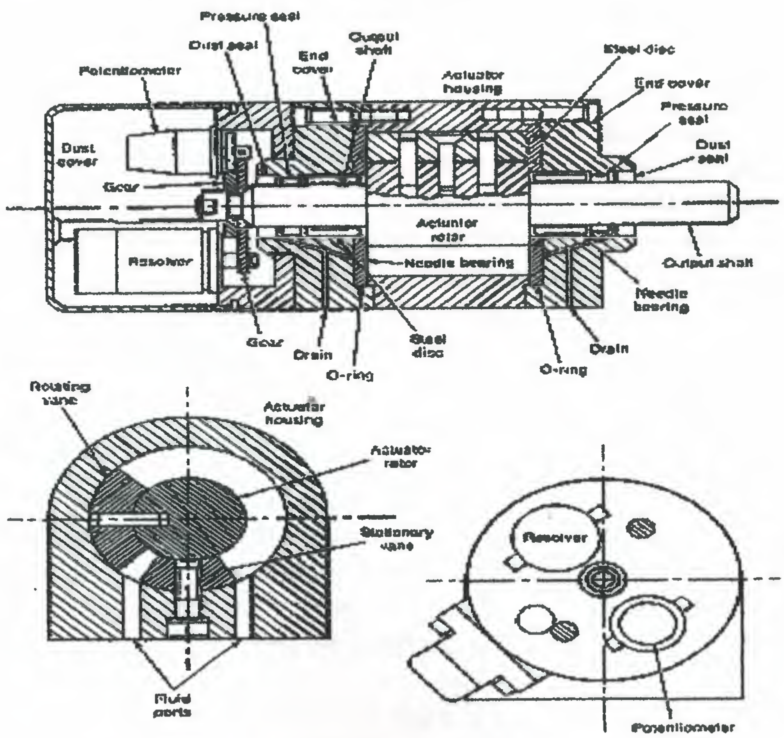


Figure 2.2 Rotary Actuator (from the TR-3500 Equipment Manuel)

2.1.3 Electrohydraulic Control Valves

Electromechanical or electrohydraulic servo valves (both terms are in common use are actually quite complicated. There are two major types: the flapper valve shown in Figure 2.3 and the jetpipe valve shown in Figure 2.4. Most industrial robots today use the flapper valve but the jetpipe valve, which has been more expensive, is now being used. Its manufacturer states that it has higher reliability and greater efficiency than the flapper valve.

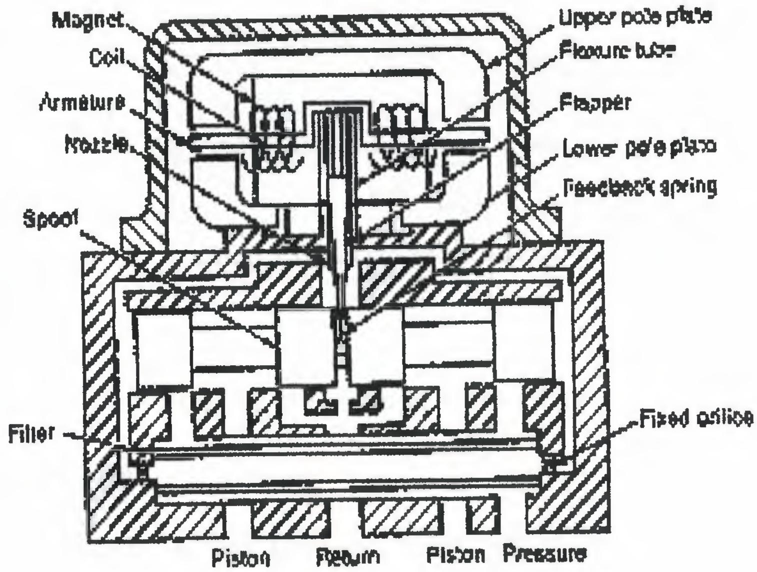


Figure 2.3 Flapper servo valve cross section (from the TR-3500 Equipment Manual)

In either type of valve, it takes only a few milliseconds to reverse the flow of fluid in the system. Each valve has a torque motor, a hydraulic amplifier first stage, and a four-way spool valve second stage. The torque motor has an armature that moves either flapper valve or a jetpipe assembly to control the flow of oil to the second stage, which controls the movement of the spool valve that actually controls the large oil flow to the driving cylinder or rotary motor. A relatively small current flow in the torque motor controls a flow of oil that causes the spool valve to move to control a large oil flow.

1. Flapper Servo Valve

In a flapper servo valve, the flapper is rigidly attached to the midpoint of the armature. The flapper passes between two nozzles, creating two variable orifices between the nozzle tips and the flapper. An electrical signal generates a magnetic field that moves the armature and the flapper to open one orifice and close the other. This action causes an oil flow that. Sets up a differential pressure on the ends of the spool valve and causes the spool valves to move. As the spool moves, it bends the feedback spring that opposes its motion. When the force due to oil pressure differential equals the spring force, the spool valve stops moving. Movement of the spool valve opens the main piston oil paths and allows oil to flow, driving the piston in the desired direction.

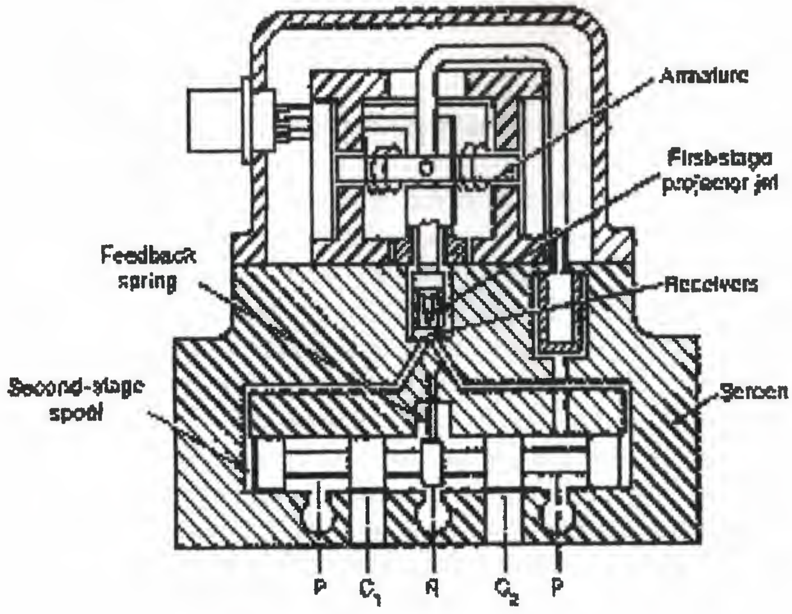


Figure 2.4 Jet pipe servo valve

2. Jetpipe Servo Valve

A jetpipe servo valve differs from the flapper servo valve in the way oil flow to the spool valve is controlled. When the torque motor is energized, it causes the armature and jetpipe assembly to rotate. More fluid is injected into one receiver than the other, causing the spool to move is. Otherwise the action is essentially the same. An advantage of the jetpipe valve is that the openings through which oil passes are larger and less subject to blockage by tiny metal particles in the oil.

One of the major problems in hydraulic systems is the need to filter the oil in order to remove contaminants. Despite extreme care in manufacturing, it is possible for particles a few microns in diameter to break off from welded spots or rough spots in the cylinders, tubing, and pistons. Careful filtration and frequent cleaning of the filters are required to reduce the potential for blockage of the sensitive servo valves.

2.1.4 Closed-Loop Servo Control

Precise control of a hydraulic system is obtained by use of a complete closed-loop servo system, as shown in Figure 2.5. An electrical servo amplifier, usually an operational amplifier, provides an electrical signal to the electromechanical servo valve located in the hydraulic fluid system. The pilot valve, which is, controlled electrically in turn controls one or two stages of hydraulic amplification, generating enough power to drive the mechanical linkage.

A feedback potentiometer, resolver, or encoder mounted on the mechanical drive generates an output signal that is fed back to the comparator. In the "repeat" mode, the encoder output is compared to the output from the memory drum, and an error signal is generated that is sent to the servo amplifier to act as a control signal. When the encoder output signal is equal to the command signal from memory, the servo valve is allowed to close and the mechanical motion stops. In a computer-controlled system, the comparator and memory are part of the computer but perform the same types of functions. Included in Figure 2.5 are the teach control, the memory drum, and the comparator used for teaching, as described in Section 2.1.5. Only one degree of freedom is shown for the robot.

2.1.5 Training a Hydraulic Robot

In training a hydraulic robot, auxiliary valves release the pressure in the hydraulic system or other means so that the robot operator is not working against the powerful actuators. In some systems, the "Train" switch operates all of the necessary auxiliary and bypass valves to allow the operator to guide the robot manually. Even though the hydraulic pressure has been released, a large robot arm would be difficult to move. Therefore, in some systems, springs are used to counterbalance the weight of the arm.

As the operator, he or she moves the arm through a cycle can record the coordinates of each joint at selected positions along the path or can have the control computer take sample coordinates periodically along the path. Coordinates are taken for all of the joints simultaneously and stored either in the computer memory or on the magnetic drum or disk used in the system. After training is completed, the operator can switch to the repeat mode and watch the robot go through the same sequence of steps just completed during the training mode (including all of the mistakes made.)

A further refinement in some of the newer robots is the use of a complete training arm that duplicates all of the position sensors of the actual robot arm and is much easier to move. Even when the robot arm is counterbalanced, it still has a considerable mass that is difficult to start and stop smoothly when training. The lightweight training arm avoids this difficulty and allows the operator to train the robot with greater speed and precision. During training, it is mounted beside the actual arm and at a measured distance from it. A computer program that calculates the necessary correction for all trained movements provides compensation for the separation between arms.

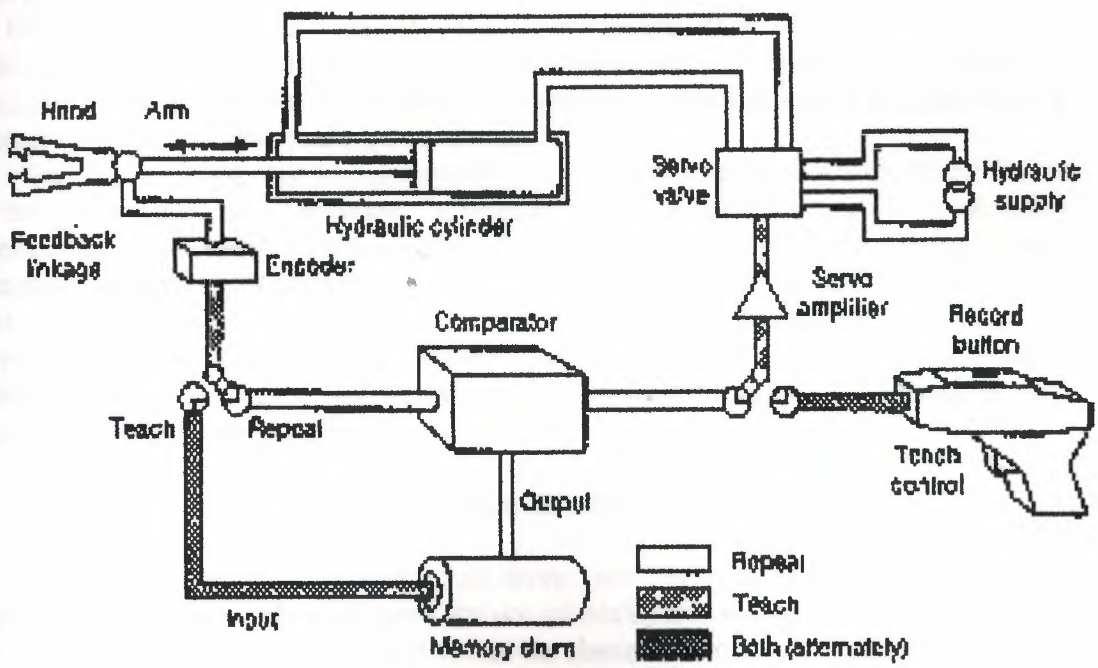


Figure 2.5 Simplified diagram of a hydraulic servo system (Unimate).

2.1.6 Hydraulic Power Source

As illustrated in Figure 2.3, the initial power source for a system is a powerful constant-speed electric motor driving a hydraulic pump. Oil under pressure is stored in an accumulator at pressures on the order of 1,500 to 2,000 psi. Cooling of the oil under pressure is provided by water from the local water supply, which is piped through the accumulator and back to the drain.

Filtering of the oil is necessary to prevent contamination in the oil from affecting the sensitive electrohydraulic valves, drive cylinders, and rotary actuators. Particles only a few microns in diameter break loose from the interior of the hydraulic system and cause jamming or scoring of valves and cylinders if not quickly removed.

2.1.7 Advantages and Disadvantage of Hydraulic Drive

Electrohydraulic servo valves used to control oil flow are quite expensive and require the use of filtered, high-purity oil to prevent jamming of the servo valve. In operation, the electrohydraulic valve is driven by a low-power electric servo drive that is relatively inexpensive. This low cost has offset the cost of the servo valve itself and the expensive and somewhat messy hydraulic tank, tubing, and filter system. Because of the high pressure, there is always a potential for oil leaks. Oil at 2,000 psi can quickly cover a large area with a film of oil, so this is a problem to be watched. As a result, the tubing and fittings used are expensive and require excellent maintenance for reliability.

Since the hydraulic cylinder provides precise linear motion, it is advantageous to use linear drives in the robot as often as possible. However, hydraulic rotary vane motors have been designed that provide good operation, although at somewhat higher cost. Rotary hydraulic motors can be made smaller for a given power output than electric motors. This is an advantage when a robot arm such as in a jointed configuration must carry the motor. The size and weight advantage is offset somewhat by the need for rotary joints to carry the high-pressure oil to other parts of the arm. Also, the newer designs of electric motors are beginning to be competitive in size and weight because of the use of new magnetic materials. Although expensive, the electric motors are more reliable and require less maintenance.

Perhaps the most significant advantage of hydraulic drives over electrical drives is the intrinsically safe operation. In explosive atmospheres such as in paint spray booths, there is a rigid requirement for safety. OSHA requires that electric equipment in certain defined areas carry no more than 9 volts because of the possibility of arcing and explosive action. Hydraulic systems do not have a problem with arcing and are now chosen almost exclusively for use in explosive atmospheres. An alternative would be the use of sealed electric motors, but at present the cost and weight of such motors are prohibitive for the power required.

2.2 PNEUMATIC DRIVES

Pneumatic drives, the simplest of all drives, are widely used in industry for many types of applications. Many of them are not robots by any of our definitions, but there are also many pneumatic systems that can be classed as robots. Both linear cylinders and rotary actuators are used to provide the motion required.

Several robot manufacturers are making very flexible robots using pneumatic drives. These are much like hydraulic systems in principle, but vary considerably in detail. The working fluid compressed air, but the valve is simpler and less expensive and the pressures are much smaller.

Most pneumatic drives provide motion between stops. Air is so compressible that fine control of motion is difficult. Even when high-pressure air is applied to the opposite end of a piston, the inertia of the piston and load causes it to continue to move until it hits a mechanical stop or the air force finally overbalances the inertial force.

Mechanical stops can be used to provide high accuracy of positioning in pick-and-place application. Accuracies of 0.005 inch are achieved readily. Cushions in the air cylinder or associated with the mechanical stops slow the pneumatic cylinder at the end of its motion to prevent damage to the equipment or parts being handled. With this simple, easily understood, and easily programmed equipment, it is possible to do a large number of pick-and-place tasks. Pick and place is used to describe the operation of picking up a part at one location, oving to a new location and placing the part for a particular operation. This simplicity of operation is one of the great virtues of pneumatic systems.

High precision is difficult to achieve using pneumatic servo drives, but where the precision is adequate, pneumatic drives are lightest in weight and lowest in cost of all available robots.

One manufacturer, International Robomation/Intelligence, has developed a new approach to pneumatics using air servomotors. These are vane-type motors controlled closely by individual microprocessors. The manufacturers expect to obtain high positioning accuracy. The maximum load to be handled is 50 pounds. Low cost is the major advantage of this technique. The manufacturers are quoting a price of less than $10,000 per complete robot arm, including the computer control system. Repeatability of 0.040 inch is specified. Such a robot will be very competitive with the older hydraulic and electric motor-drive robots if it can be made to work with high accuracy and reliability.

Modularity is one of the biggest assets of pneumatic drives. Since the operating fluid is air, it is easy to run many air hoses to separate drives and build up an arbitrarily complex system from standard parts.

Power for the pneumatic system is supplied by compressed air from a good-quality air compressor. All the modules of the pneumatic system through a common air chamber called a manifold share this air supply. Solenoid valves mounted on the manifold control the airflow to individual pneumatic actuators. In the simplest systems, stepping switches, limit switches, or part sensing switches controls the solenoid valves. Several actuators can be assembled to provide three to six separate motions. Cylinders of different lengths can be used, for example, to build up a positioning mechanism.

Programmable controllers are frequently used for control with pneumatic systems. These controllers are usually microprocessors programmed to emulate relay systems. Relay systems are well understood by a generation of shop foremen and electricians. They find it easier to plan a relay sequence and enter it into the programmable controller than to learn to program a microprocessor. As a result, many simple control operations in factories use programmable controllers.

Pneumatic robots can be trained like other robots. Point-to-point operations can also be controlled by a pendant controller, which has control switches to move the robot through a sequence of tasks.

2.3 DC ELECTRIC MOTORS

We distinguish between electric direct current (DC) motors and stepper motors because of their inherently different methods of operation. DC motors run continuously in one direction, reverse perhaps, and run continuously in the opposite direction. Motion is smooth and continuous. There is no inherent control of position in a DC motor. Stepper motors are able to step precisely to a designated position but have other disadvantages, as described in Section 2.4.

Precise control of positioning in DC motors requires that a closed-loop servo be used with some type of positional feedback. Because of the closed-loop control, the smooth operation possible, and the ability to generate large torque's, electric DC motors are used in those robots in which precise control and high power are required.

DC motors are driven and controlled either by relay switching or by the use of power amplifiers that electronically switch the direction of current flow through the motor armature to reverse the direction of operation. Either the field or armature current or both can be controlled.

DC motors today can achieve very high torque-to-volume ratios, much higher than those of stepper motors, and are competitive with hydraulic motors except for very high power drives. They are also capable of high precision, fast acceleration, and high reliability. The present DC motors have benefitted from the development of rare earth magnets, which have made possible large magnetic fields in compact motors. Also, the improvements made in the fabrication of brushes and commutators have made DC motors reliable. Another important factor is the improvement in the power-handling capacity of solid-state devices, which has made possible the control of large currents at a reasonable cost.

2.3.1 Types of DC Electric Motors

DC motors are torque transducers that convert electric energy to mechanical energy. The torque developed by the motor shaft is directly proportional to the magnetic field flux in the stator field and the current in the motor armature. In practice today, the stator field in DC motors used for robots is nearly always a rare earth permanent magnet (PM) that provides a strong, stable magnetic flux in which the armature shaft rotates. Controllable field motors with wound field coils are used so rarely that we will not discuss them in this text.

Three types of PM motors are in use, each with specific advantages and disadvantages. These are discussed in the following sections.

1. Iron-Core PM DC Motors

Iron-core PM DC motors has a laminated iron rotor structure that has slots as in conventional DC motors. Armature conductors or windings are placed in these slots. On the periphery of the motor is the permanent magnet material, which may be made of Alnico, ferrite, or a rare earth alloy. Motors of this type have high inductance, high inertia, high reliability, and low cost.

Due to the action of the slots, these motors have "cogging," or a tendency to start and stop as each slot passes through the edge of the magnetic field. Since the inertia is high, they cannot be controlled to start and stop as precisely as is sometimes required for robot use. Figure 2.7 shows this motor configuration.

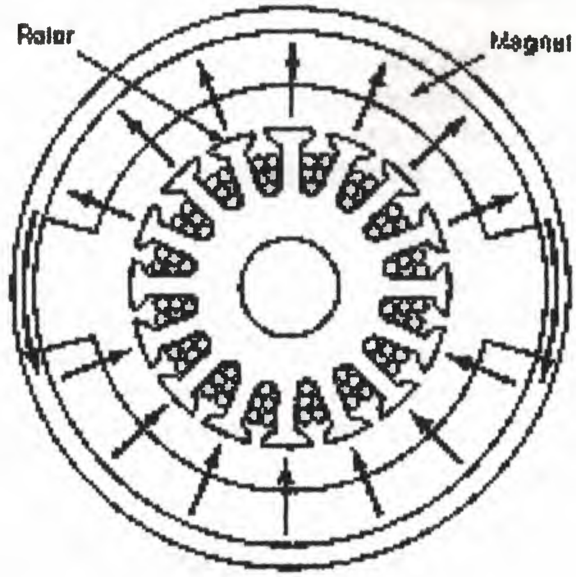


Figure2.7 Iron-Code PM DC motor in cross section.

2. Surface-Wound PM DC Motors

In an attempt to develop more responsive motors for control tasks, engineers developed the surface-wound DC motors. Armature conductors are not placed in slots but are bonded to the surface of a cylindrical rotor. The rotor is made up of iron laminations to reduce the flow of eddy Currents induced in the motor, as in the iron-core motor. However, since there are no slots, there is no cogging effect. These motors have higher inductance, higher cost, and larger outside diameters, and require a more powerful magnet than the iron-core motor. Figure 2.8 shows this motor configuration.

3. Moving-Coil PM DC Motors

In order to attain very low armature inductance and small moments of inertia, engineers eliminated all magnetic material in the armature. This step required the armature conductors to be supported by nonmagnetic, nonconducting material such as epoxy resin or fiberglass. Since there was no magnetic material to provide a magnetic flux path, it was necessary- to use larger, more powerful magnets to obtain the desired toque.

Armature inertia is extremely low due to the small mass. These motors are called moving -coil DC motors. They provide the most rapidly responding actuators for high-performance systems. Motor inductance less than 100 microhenries is commonly achieved with the configuration shown in Figure 2.9. Also available are thin, large-diameter motors in which the armature is a flat disc about 0.020 inch in thickness and less than 1 foot in diameter. These are called rinted-circuit motors by their manufacturer and provide excellent response, high torque, and good reliability.

2.3.2 DC Motor Analysis and Modeling Conductors

As previously discussed, the output torque of a DC motor is proportional to the magnetic field of the PM and the current in the armature. Motion of the armature due to the generated torque is opposed by the inertia of the armature rotor, the voltage generated by the moving coil in the magnetic field, and the load on the motor shaft due to friction, damping, or work done.

Voltage generated by the moving coil in the magnetic field is called back electromotive force (back emf). This voltage rises with rotor velocity until the back emf plus the voltage lost in the armature due to current flow through the armature resistance becomes equal the driving voltage of the motor. Initally, the back emf is zero, so that extremely high currents can flow through the low resistance of the armature. Some type of current limiter is required in the motor to protect the motor under these conditions.

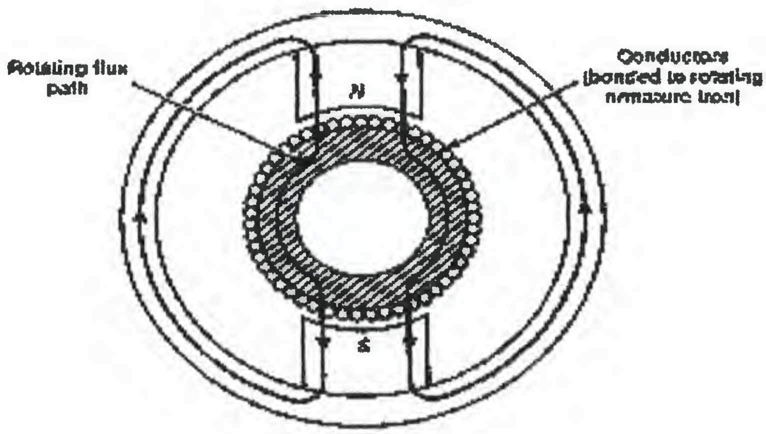


Figure 2.8 Surface-wound PM DC motor in cross section.

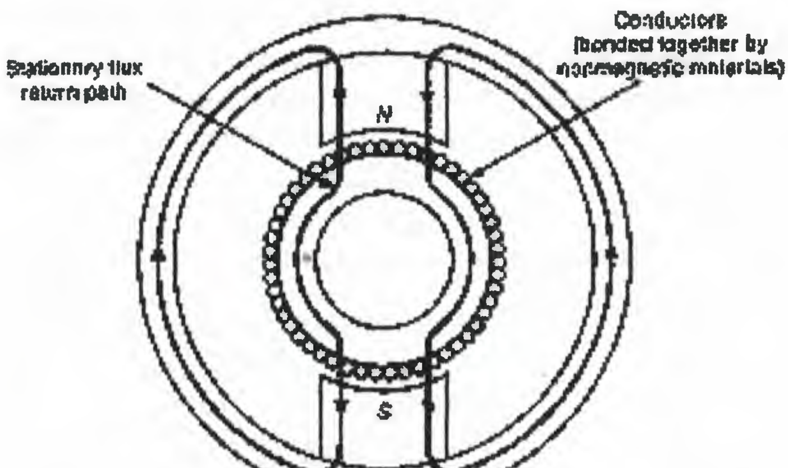


Figure 2.9 Moving-coil PM DC motor in section.

An example calculation of torque, angular velocity, and other parameters in a PM motor follows. In this example, we have neglected the effects of motor inductance, viscous damping, and nonlinear effects. However, this example will serve to illustrate the relation-ships among the parameters in these motors. A mathematical model of the motor is given in Figure 2.10 using the nomenclature provided in the next paragraph.

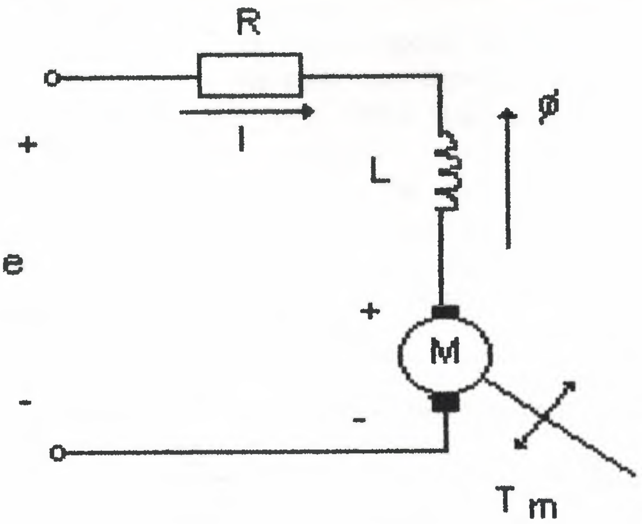


Figure 2.10 Model of a DC motor.

Terminology in motor and servo system analysis has been standardized to a considerable extent. Some standard terms are given below:

J_m = rotor inertia of the motor, fixed
J_r = inertia of the robot link being driven, time varying
R_a = electrical resistance of the motor armature, fixed
V = voltage applied to the motor
K_t = torque constant, fixed
K_b = back emf constant, fixed
T_m = torque developed by the motor, time varying
T_s = load torque due to friction or other causes, time varying
ϕ = Magnetic flux in the field, fixed by a permanent magnet
Θ = Rotor angular displacement, time varying
I_a = armature current, in amperes, time varying
W = rotor angular velocity, radians per second, time varying
am = rotor angular acceleration, radians per square second, time varying
ea = voltage across the armature, time varying
eb = back emf, time varying
B_m = viscous frictional coefficient, fixed
L_a = armature inductance, fixed

Using the terms just defined, we can list the equations that govern the action of the motor and its load. Torque, Tm , varies with time and is determined by multiplying the current in the armature at any specific time, ia , by the torque constant, Kt .

$$Tm = Ktia$$

Back emf, the voltage generated by the armature rotating in the magnetic field, is eb = Kb W since it is a linear function of the rotor angular velocity.

Other equations allow us to calculate the current and angular acceleration from the applied voltage and the known constants.

V = ia Ra + Kb Tm = Kt ia - Jm am

Some tasks performed by robots require speed control. One example is spray painting, in which the thickness of the paint coat depends on both the spray rate and the speed of movement of the robot arm. Other tasks require control of torque or force, as in tightening a bolt or lifting an object. As can be seen from the preceding equations, rotor velocity, which determines the speed of the arm link, is itself determined by the armature current and the voltage across the armature. Koren and Ulsoy point out that it is not possible to control both the speed and the torque of a robot arm at the same time. We can control one or the other, but not both at once, at least with the simple model we are considering.

2.3.3 Open-Loop Torque Control with Current Amplifier

Torque control can be achieved by controlling current in the DC motor with the use of a current amplifier. In this case, the voltage applied to the motor is forced to vary in such a way that it maintains the current through the armature at a constant amount. As can be seen in the preceding discussion, torque depends on both the current and the changing moment of inertia of the load. However, by measuring these values and compensating for them in the controller, it becomes possible to maintain a constant torque.

2.3.4 Open-Loop Torque Control with Voltage Amplifier

Using a voltage amplifier to apply a controlled voltage to the drive motor can control the speed of the robot arm link. Variations in load inertia do not affect the final position error of such a system but do affect the time constant. However, torque is not controlled and the current can rise to high values if the is stopped or resisted so that the back emf is reduced. A current overload circuit can protect the motor from excessive current at the cost of stopping the operation.

2.3.5 Closed-Loop Torque Control

Either voltage amplifiers or current amplifiers can drive robot arm links in closed-loop or servo control.

2.3.6 DC Power Sources and Power Amplifiers

Power sources and a power amplifier is an important and complex subject. In this section, we will review briefly the types of circuits used in robots and refer the reader to the references given for more detail.

Usually the power supply available in industrial plants is single-phase or three-phase, 60-cycle AC at nominal voltages of 110, 220, or 440 volts. Power converters are required to convert this AC to DC in order to operate electric drive motors on the robot joints. Then the DC must be regulated to a reasonably constant voltage level and applied to a power amplifier or switching circuit, which controls the current or voltage and direction of power applied to the drive motors.

Three types of circuits are used: rectifiers, power regulators, and power amplifiers or switching circuits. Input from an AC source is fed into a transformer that changes the input voltage to the desired level (usually lower) required operating the power control circuits.

1. Rectification

The transformer output is still AC and must be converted to DC by being passed through a rectifier. The rectifier allows current to pass through in only one direction, so its output is a DC of varying voltage level. Either half-wave or full-wave rectification can be used. Half-wave rectification uses only one solid-state diode, so the voltage is off half of the time and much energy is lost. A full-wave rectifier circuit, as shown in Figure 2.11, uses two diodes in a bridge arrangement, so the current goes to a peak twice per cycle, or 120 times per second, for a 60-cycle input.

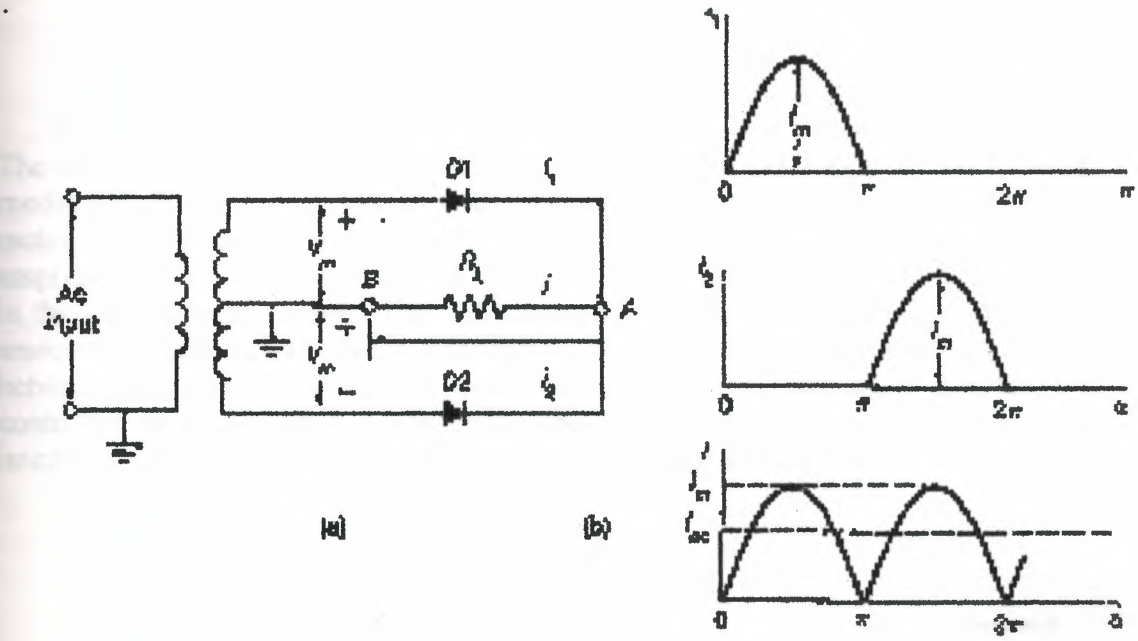


Figure 4.11 (a) A full-wave rectifier circuits. (b) The individual diode currents and the load current i. The output voltage is vo = iRl

2. Regulation

This varying voltage from the rectifier is termed the unregulated power. It is passed through another circuit that stabilizes the voltage level to a nearly constant value and applies it to the load. The voltage level can be stabilized either by filtering or by providing negative feedback that opposes the high voltages and enhances the low voltages to generate a nearly constant output voltage level. Filtering is done by putting large inductance's in series with the current flow and large capacitors in parallel with both the input and output. In the circuit of Figure 2.12, the negative feedback method is used to produce a regulated voltage across the load resistor, Rl.

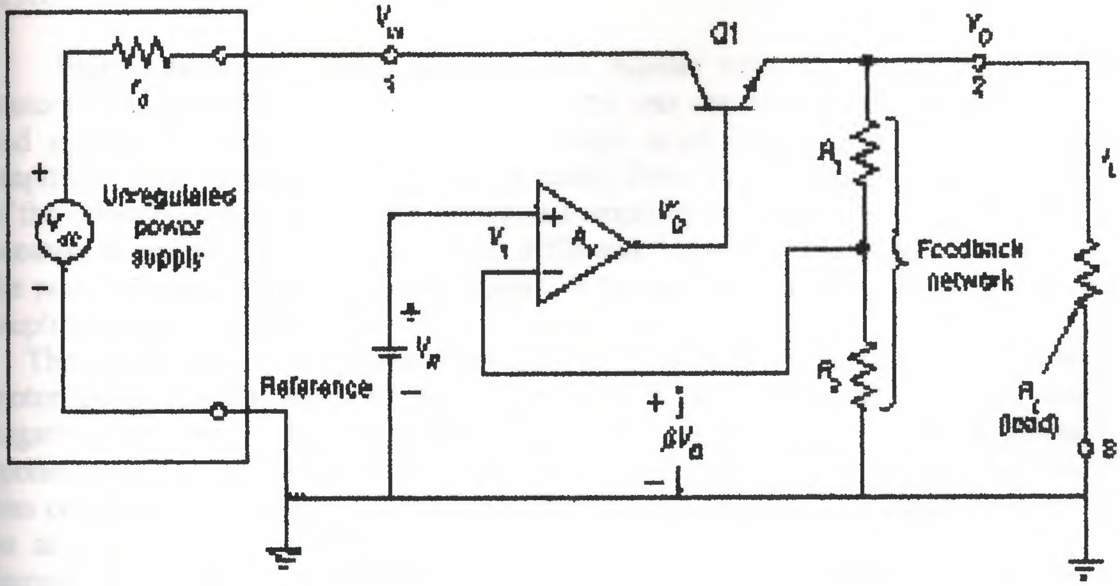


Figure 2.12 A regulated power supply system.

An improved regulation method is the switching regulator shown in Figure 2.13. The conversion efficiency of this circuit can exceed 90%. This type of circuit can be modified to control the direction and amplitude of the drive current applied to the motor. In the diagram, the pulse-width modulator (PWM) generates a constant amplitude pulse of varying width so that the average current can be varied from zero to the maximum available. The inductance, L, and the capacitor, C, to provide a smoothly varying output current filter this output. Since the power switch is switched between ground level and the Vin level, the current can be reversed in direction and controlled in magnitude. In the figure, the outlined area is provided on a single integrated-circuit chip. Other components are discrete elementsadded externally.

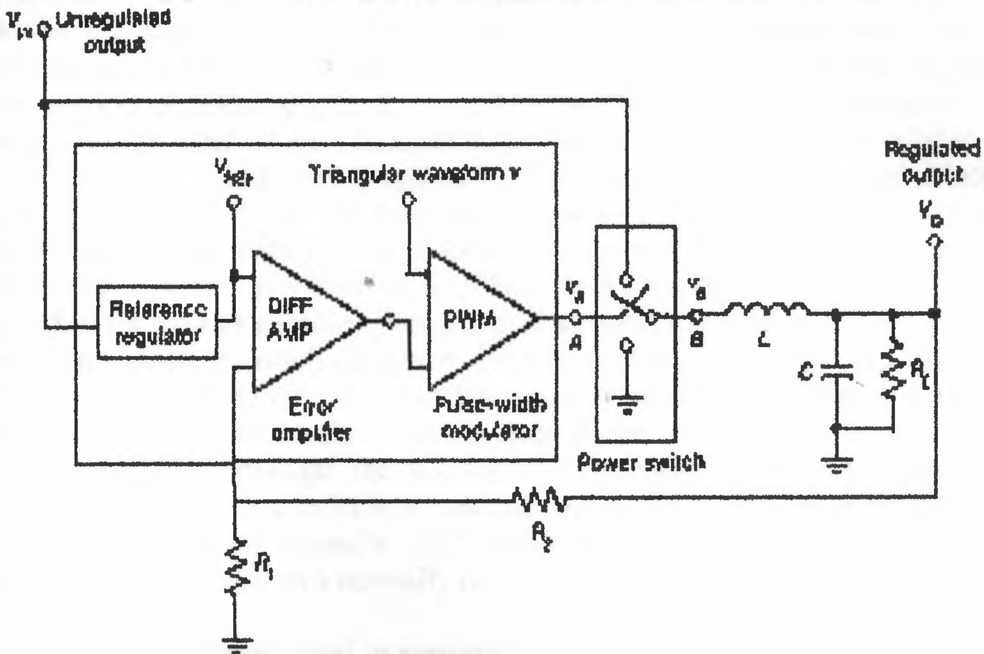


Figure 2.13 Basic switching regulator circuit.

3. Amplifiers and Switching Circuits

Early robots used power supplies with bipolar transistors to drive electric motors. The feedback from the position encoder was converted to an analog voltage and applied to the one input of a differential amplifier, a form of operational amplifier, while the control signal, also in analog form, was applied to the other input of the differential amplifier. This differential amplifier was called the error amplifier because its output was a measure of the difference between the desired position and the position actually existing at that instant. It operated in the same way as the error amplifier shown in Figure 2.13.

The error voltage could be used to control the direction of current flow through the motor by turning on either of two power amplifiers. Two power supplies, one negative and one positive, were used; they were connected to a common ground connection. One side of the motor was connected to the ground, and the other side was connected to the output of both amplifiers. Only one amplifier could be turned on at any one time; otherwise they would short each other out and cause great damage. It was therefore necessary to control carefully the switching of the power amplifiers so that only one was connected at any time.

A new type of switching converter developed by S. Cuk (pronounced " chook ") and R. Middlebrook of Cal-Tech can be used either as a switching regulator or as a combined regulator and power amplifier. An ingenious switching scheme is used to switch the AC input through large inductances so that the current through the load can be controlled in both magnitude and direction. Output voltages can be larger or smaller than the input, efficiency is high, and the electromagnetic interference (EMI) is reduced to a small value. Although the actual circuit is relatively simple, the theory of operation is complex; the reader is referred to reference, for further details. The circuit of a bi-directional power amplifier is given in Figure 2.14. A drive motor can be used where the speaker is shown as the load. Switching can be done with power transistors or with the power MOSFETs described in the next section. A circuit similar to the full-wave rectifier circuit of Figure 2.11(a) could be used to provide the DC supply voltage identified as Vg. Bipolar transistors were used in conventional power amplifier design for several years. They were then used in pulse-width modulators and other improved circuits to improve switching performance.

Recently, a number of new semiconductor devices have become available, as listed in Table 2.1, which was prepared by B. J. Baliga of the General Electric Company. The MOSFET devices have turned out to be ideal for power switching circuits because of their high switching speed, low drive power requirement, and low cost. Several companies now produce MOSFET devices at costs ranging from $5 per unit up to $30 for devices capable of handling 30 amperes at 100 volts.

Power dissipation in switching devices is an important factor. Since the MOSFET switches at a 2-megahertz rate, it has a dissipation about one-tenth that of a bipolar transistor. It is also considerably cheaper, since it can be made by simple (MOS) masking techniques. Although the (GaAS FET) devices appear to have better capability, they are not yet available at low cost. MOSFET s, bipolar transistors, and gate turn-off transistors are normally "off," while the other three devices may be "on." It is safer and easier to turn on a normally off device than to ensure that a device is off by other means.

Bipolar transistors were used in conventional power amplifier design for several years. They were then used in pulse-width modulators and other improved circuits to improve switching performance.

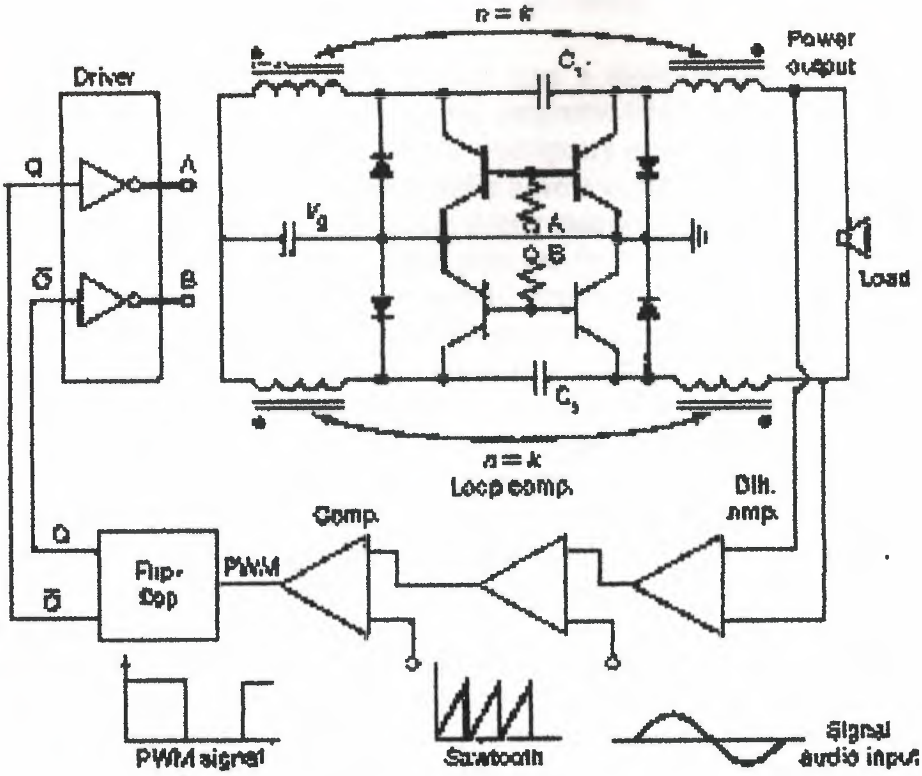


Figure 2.14 Cuk push-pull switching power amplifier.

Recently, a number of new semiconductor devices have become available, which was prepared by B. J. Baliga of the General Electric Company. The MOSFET devices have turned out to be ideal for power switching circuits because of their high switching speed, low drive power requirement, and low cost. Several companies now produce MOSFET devices at costs ranging from $5 per unit up to $30 for devices capable of handling 30 amperes at 100 volts.

Power dissipation in switching devices is an important factor. Since the MOSFET switches at a 2-megahertz rate, it has a dissipation about one-tenth that of a bipolar transistor. It is also considerably cheaper, since it can be made by simple (MOS) masking techniques. Although the (GaAS FET) devices appear to have better capability, they are not yet available at low cost. MOSFET s, bipolar transistors, and gate turn-off transistors are normally "off," while the other three devices may be "on." It is safer and easier to turn on a normally off device than to ensure that a device is off by other means.

2.4 ELECTRIC STEPPER MOTORS

Stepper motors are used in robots when open-loop precise control is required and the torqueses required are small. These motors can step in precise increments of motion under the control of electrical pulses. Computer printers and disk drives often use steppers to provide positioning of the print head or the disk head. In small robots, stepper motors are sometimes used for the main drive motors. Encoders or potentiometers can be used to provide position feedback, so that stepper motors can also be used in closed-loop servos.

2.4.1 Stepper Motor Operation

Stepper motors, like other electric motors, have a stator, or stationary magnetic element, and a rotor that rotates in the varying magnetic field produced by the stator and drives the load. Usually the stator is on the periphery of the motor and is made up of multiple electromagnetic poles. Stator poles have a central magnetic core and are wound with many turns of wire to produce a strong magnetic field. In stepper motors, there may be a large number of poles. Each pole can be energized to be either a north or south pole by applying current in the specified direction.

The multiple magnetic poles, arranged like gear teeth around the periphery of the stepper motor armature, provide separate latching points, or steps, for the armature. Motion from step to step is caused by pulsing electric currents into the motor stator coils, which cause the polarity of the stator coils to change from north to south. It is this change of polarity that provides the motive power of the stepper motor. Since stepping is inherent in the operation, it is possible to move from step to step and go to a desired position merely by moving a specified number of steps. A price paid for this simplicity is the relatively large size of the stepper motor as compared to a DC motor for the same power output.

Rotors have two sets of poles machined in two separate sections along the length of the rotor. Each set of teeth looks like a gear on a separate hub along the rotor, with one set of teeth offset by the width of half of a tooth from the other set. One set of teeth comprise the north poles of the rotor, the other set the south poles. In reality, there is one magnet with multiple teeth or poles.

Each stepper motor is built to rotate through a specified number of steps in one revolution, although it car rotate in half steps under proper drive conditions at reduced torque. Since there are many poles in both the stator and the rotor, there are many possible steps of rotation.

When a drive current is applied to a selected set of stator windings, the north poles of the rotor line up with the south poles of the stator. By changing the stator poles from north to south, the rotor is forced to step from one stable position to another. A typical energization sequence is shown in Figure 2.15. When the polarity of current applied to the stator windings is changed, the PMs on the rotor is made to step from one pole to another. Each step is a separate movement. Reversal of the polarity sequence causes the rotor to move in the reverse direction. Steps are precise and typically range from 1.8 to 30 degrees apart. When the electromagnets are changed rapidly, the stepper motor steps at an essentially continuous rate. However, there is always a cogging, or step effect, in even the smoothest stepper drives.

Stepper motors can be moved in either direction of rotation by properly phasing the currents applied to the stator windings. Most stepper motors are designed for two-, three-, or four-phase operation.

Two types of pulse sequences are commonly used. In wave drive, only one phase of the stator is pulsed each time. In two-phase drive, pulses are applied simultaneously to two stator phases. Two-phase drive produces greater torque from the motor but is somewhat more complex to provide. It is possible to mix the two pulsing modes to get half step driving, which produces a smoother response.

Resonance is an inherent tendency to oscillate at a particular frequency due to the mechanical and electrical characteristics of a design. It may occur in stepper motors. It can cause strange effects, including backward rotation, low torque, and vibration. Usually, resonance occurs at 50 to 150 steps per second. Half step driving helps prevent resonance. Friction damping and special control circuitry can also be used.

The best solution is to avoid prolonged operation at the resonance frequency by careful design. It is possible to accelerate or decelerate through resonance without causing problems.

Figure 2.16 shows the winding diagram and the current in the windings for a wave drive and a two-phase drive. Pulses are applied to the stator windings in a specific sequence, as shown, to control the operation of the stepper motor. Controllers are necessary, therefore, to provide the correct sequence of drive currents in the proper phase relationship.

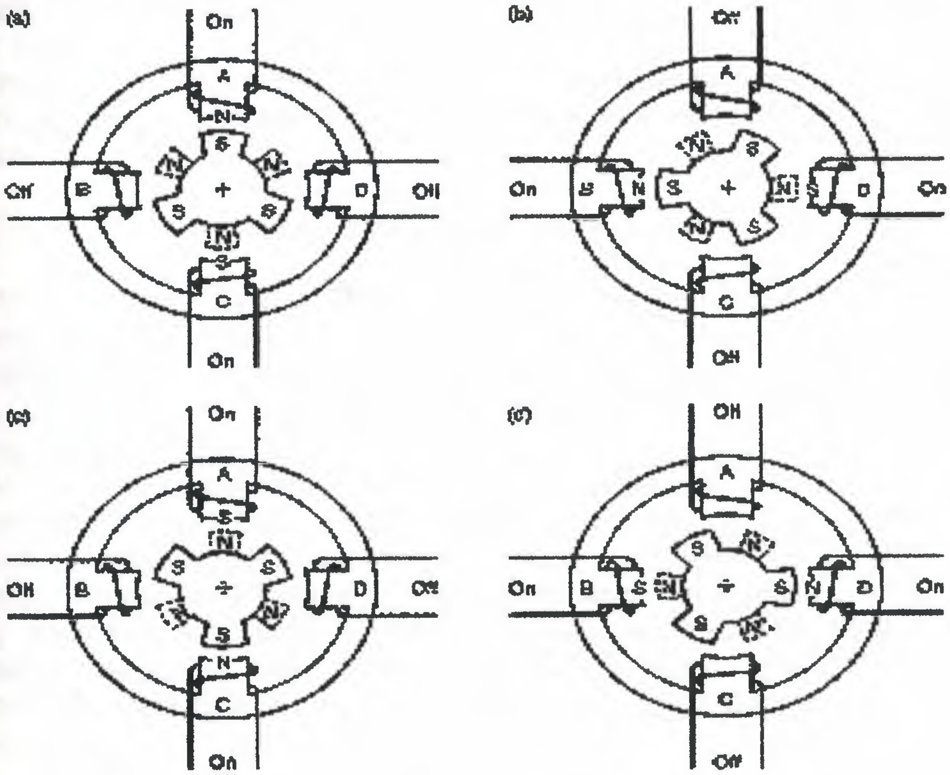


Figure 2.15 Energization sequence for a stepper motor.

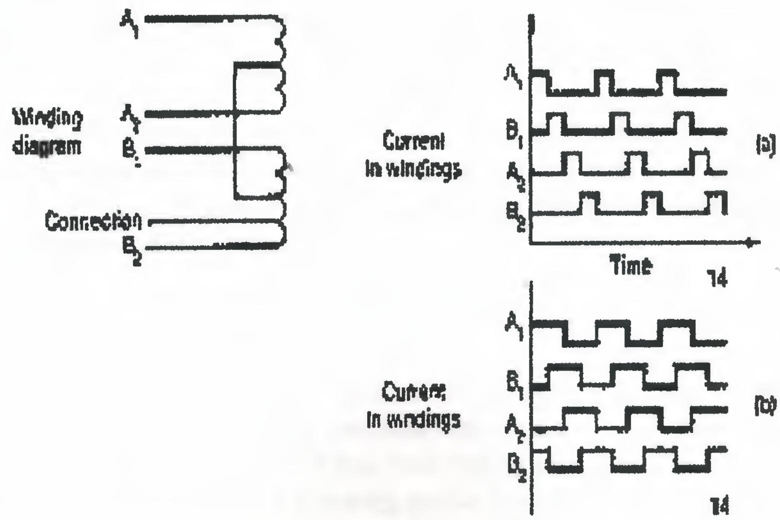


Figure 2.16 Sequencing of pulses applied to a stepper motor. (a) Wave drive. (b) Two-phase drive.

Controllers contain a combination of electronic circuits to generate and apply the drive currents required. Several manufacturers produce stepping motor electronic circuit drives cards that accept two input signals-a drive pulses and a direction of rotation signal. Every time the drive pulse is applied to the drive card, the drive card generates the pulse sequence in the correct sequence to drive the motor in the direction specified by the direction of rotation signal. The pulse sequence is selectively applied to solid state power amplifiers, which supply current to the stator windings as required. Power amplifiers used may be single power transistors or paralleled transistor drivers controlled by operational amplifiers, depending on the power required.

Powerful MOS-FET power transistors now available can control more than 25 amperes at voltages up to 100 volts. Thus, the drive problem has become much easier to solve.

2.4.2 Types of Stepper Motors

PM-type motors have a PM rotor as previously described, and one set of windings on each pole. There may be only a few pole windings on the stator, but each may have multiple teeth machined into the pole pieces, so that there are, in effect, as many as 200 or 400 poles per revolution. This arrangement reduces the cogging effect and smoothes the operation.

Bi-filar motors are similar to permanent magnet (PM) motors but have two sets of windings on each pole. Each winding is of smaller wire than in the PM motor, and so has higher resistance. Therefore, the value of L (Inductance)/R (Resistance), which affects motor response, is smaller and the response time is improved. Only one power supply is required to drive the Bi-filar motors, since the direction of rotation can be switched by switching power to different windings rather than by reversing the direction of current flow in one winding. Both a negative and a positive supply are required to drive PM motors.

A third type of stepper motor is the variable reluctance motor. It uses an unmagnetized rotor, so that the rotor position is independent of the polarity of the stator excitation. Only one power supply is needed. Direction of motion is controlled by the phasing sequence only. It has a slower response to pulsing than the PM motor but has a lower inertia, which is desirable in some cases. However, the variable reluctance motor has no holding torque when power is turned off, as the motors with PM rotors do.

An excellent discussion of stepper motor types and methods of operation is given by Giacomo. S discusses the theory of operation and some advanced techniques. Motiwalla of the General Electric Corporation. Kuo provides a complete analysis of stepper motor design and application.

2.4.3 Open-Loop Operation

The pulsing sequence shown in Figure 2.16 is usually generated with integrated circuits that have been designed to perform this function, as described in Section 2.4.1. Typically, the integrated circuit has two inputs and four outputs. Inputs specify the direction of motion and supply a timing pulse to control the timing of the motion. Outputs are the square wave pulses to be applied to the stator windings.

In the simplest type of stepper motor operation, it is only necessary to hook this circuit to the stepper motor and enter a series of pulses from an oscillator or a counter circuit. Pulsing may be done under computer control by setting up a digital number in the computer and counting it down to zero, emitting a pulse each time the number is decremented. It is this simplicity that has made the stepping motor valuable for open-loop operation. Although no direct position feedback is used in these circuits, it is possible to position the stepping motor to high accuracy. However, positioning accuracy depends on the stepping motor's moving one step for every pulse applied. If this does not happen-if the motor is stalled or blocked-there is no information regarding the position of the object being driven.

2.4.4 Closed-Loop Control

Position information is required to provide assurance of high-precision positioning with a stepper motor.

Potentiometer output is passed through an analog-to-digital converter to generate continuously a digital count corresponding to the position of the potentiometer shaft, which is geared to the stepper motor shaft. The incremental encoder is counted into a register to keep a record of how many steps have occurred. In either case, a register or counter has a record of how many steps the stepper motor has made. Assume that an encoder is used to count the steps made and that this count is stored in an encoder register.

Positioning is initiated by setting a computer (or other position control device) to a specified digital number corresponding to the distance to be moved. A pulse is emitted from the computer and applied through the stepper motor controller to the stepper motor, causing it to rotate. The digital number from the encoder register is continuously compared to the computer register specifying the distance to be moved. When the two numbers agree, the comparator output goes to zero and signals that the movement has been completed as specified.

Since an arithmetic element is used as the comparator, there is automatic correction of overshoot. A negative number is generated that changes the direction of motion of the stepper motor to drive it backs into the correct position.

2.4.5 Specialized Control Electronics for Stepper Motors

Maximum operating efficiency is obtained from stepping motors by ramping-controlled acceleration to a high speed, followed by controlled deceleration to a lower speed. Either linear or exponential acceleration is used. During acceleration, speed is increased rapidly so that the pulse rate is high enough to generate the minimum torque required to move the load. Then it is cut back so that a desired average speed is obtained. The reverse process is used during deceleration. Special ramping circuits are available to produce the proper pulse rate so that the total numbers of pulses, which control position, are obtained. In operation, the desired total number of pulses is loaded into a register; then the ramping circuit converts these pulses to a series of fast and slow pulses so that the total number pulses is obtained as desired. Figure 2.17 is the block diagram of a typical ramping circuit. The pulse source shown in the diagram could be the control register in the computer.

The basic reason special circuits are needed for optimal control is that the stepper motor acts like an oscillating pendulum. It applies torque to a load and then stops, so that the spring force of the load tends to backdrive the stepper motor. Therefore, it is necessary to maintain an average torque above a minimum value to obtain a reasonably smooth response.

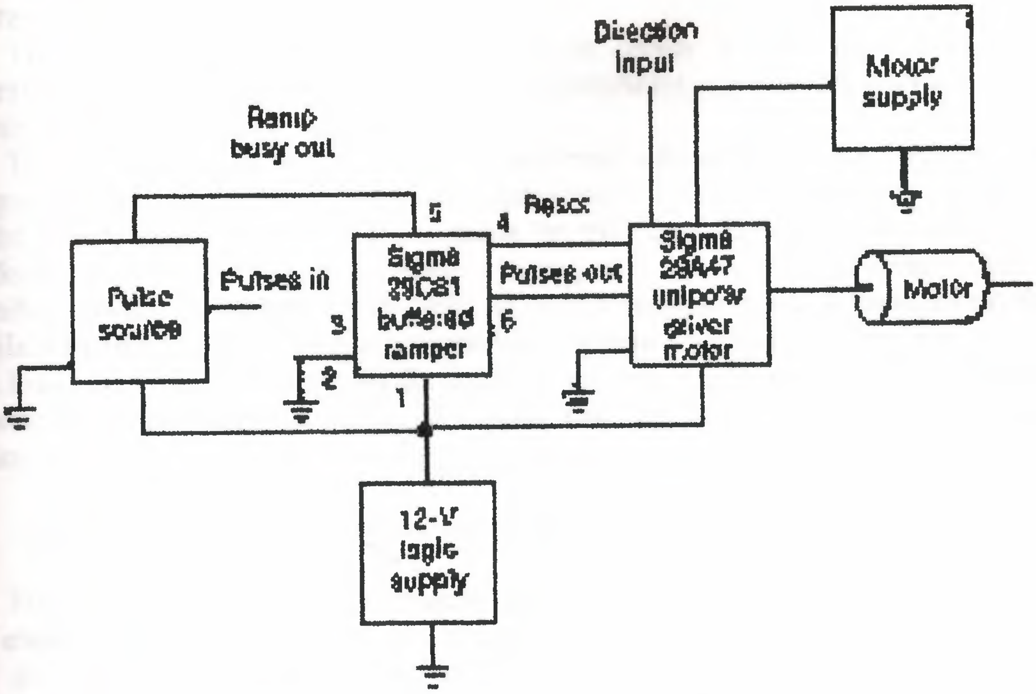


Figure 2.17 Ramping circuit for a stepper motor.

2.5 SELECTION OF DRIVE METHODS FOR DIFFERENT OPERATION

For any application of robots, we must decide which of the available drive methods is most suitable. Positioning accuracy, reliability, speed of operation, cost, and other factors must be considered. This section summarizes and outlines some of these important factors and their importance in each type of operation.

2.5.1 Hydraulic versus Electric Drives

Electric motors are inherently clean and capable of high precision if operated properly. In contrast, hydraulic motors require the use of oil under pressure and pneumatic drives are not capable of high precision for continuous-path operation.

Hydraulic drives can generate greater power in a compact volume than can electric motor drives. Oil under pressure can be piped to simple motors capable of extremely high torque and rapid operation. Also, the power required to control an electrohydraulic valve is small. Essentially, the work is done in compressing the oil and delivering it to the robot arm drives. One can supply all the power powerful, efficient electric motor driving the hydraulic pump at the base of the robot or located some distance away. Power is controlled in compact electrohydraulic valves. However, high-precision electrohydraulic valves are more expensive and less reliable than low-power electric amplifiers and controllers.

Electric motor drives must have individual controls capable of controlling the power for those drives. In large robots, this requires switching of 10 to 50 amperes at 20 to 100 volts. Current switching must be done rapidly; otherwise there is large power dissipation in the switching circuit that will cause excessive heating of the switching system. Small electric motors use simple switching circuits and are easy to control with low-power circuits. Stepper motors are especially simple for open-loop operation.

The single biggest advantage of hydraulic drives is their intrinsically safe operation. They can be used in hazardous atmospheres, such as are found in paint booths and some chemical operations.

To summarize: Hydraulic drives are preferred where rapid movement at high torques is required, at power ranges of approximately over 5 horsepower unless the slight possibility of an oil leakage cannot be tolerated. Electric motor drives are preferred at power levels under about 2 horsepower unless there is danger due to possible ignition of explosive materials. At ranges between 1 and 5 horsepower, the availability of a robot in a particular coordinate system with specific characteristics or at a lower cost may determine the decision. Reliability of all types of robots made by reputable manufacturers is sufficiently good that this is not a major determining factor.

2.5.2 Compression of Electric DC Motors and Electric Stepper Motors

Suppose we have made the tentative decision that motors are required. How do we choose between electric DC and electric stepper motors? Table 2.2 summarizes the discussions of these two types of motors to aid in making evaluation. High positioning accuracy, on the order of 0.01 1 inch is assumed to be required.

Table 4.2 Electric DC motors and electric stepper motors: Advantages and disadvantages

DC MOTORS FOR HIGH-ACCURANCY POSITIONING

Advantages	Disadvantages
Linear torque-to-power ratio.	Must operate in a closed loop for accurate positioning
Larger torque-to-volume ratio than stepper motor.	Reverse current switching required To reverse motor direction (or two windings)
Precise positioning possible	Standard motors use commutators and brushes that wear. (Brushless operation is possible.)
Rapid response due to low inductance in armature.	Simple DC positioner is more expensive than for stepping motor drive.
High holding torque at a specified position.	

STEPPING MOTORS FOR PRECISE POSITIONING

Advantages	Disadvantages
Fully synchronous, so can be locked in speed to any reference with no long-term speed position error.	Larger for a given torque requirement than a DC motor.
Can operate to position precisely with open-loop control. Stable operation.	Stiffness, or position holding strength, depends on position.
Either unipolar or bipolar drives can be used. (Will operate with one battery, for example.)	Ramping and special circuits required for maximum efficiency. Not very efficient without them.
Excellent for light loads and precision open-loop positioning.	Not suitable for heavy loads high torques, and high disturbance torques.
Less expensive motor and drive for simple positioning tasks.	Accelerates and decelerates at each step. Not a smooth drives.

2.5.3 Pneumatic Drives for Lowest Cost

As mentioned before, if a pneumatic drive will meet the requirements for an application, it will almost certainly involve the lowest cost. Available point-to-point robots, such as the Auto-Place 10, which has six axes of motion, sell for about $12,000 and the basic International/Robomation robot sells for less than $10,000. These prices do not include the air compressor required, however.

2.5.4 Drive Selection Calculations

Some simple mathematical calculations are needed to determine the torque, velocity, and power characteristics of drive motors for different applications. These values are all related, of course.

Torque is defined in terms of a force times distance or moment. A force, F, at distance, L, from the center of rotation has a momentor torque, T. (Both T and M are used to designate the value of the torque, we will use T in this book.)

$$T = FL \qquad (1)$$

Torque is in ft.-lbs. when L is in feet and F is in pounds.

In general terms, power P is transmitted in a drive shaft is determined by the torque, T, multiplied by the angular velocity, w, in radians per second. (Angular velocity in degrees per second may be converted to radians per second (by dividing by 57.3).

$$P = TW \qquad (2)$$

The torque capability of a motor is determined by use of the equation

$$T = 63,000(hp/1\square \qquad (3)$$

where hp= number of horsepower, N = revolutions per minute (rpm) of the motor, and T = inch-pounds (in-lbs.). Given the torque, T , and the revolutions per minute, N, we can calculate the horsepower by solving equation (3) for hp:

$$hp = TN/63,000 \qquad (4)$$

where T is in-1bs. And N is in revolutions Per minute.

In International System (SI) units, this equation is written

$$T = 9.55\ (P/\ N) \qquad (5)$$

where T is in Newton-meters, P is in watts, and N is in revolutions Per minute. The conversion between units is:

$$1\ hp = 746\ watts \qquad (6)$$

$$Newton = 0.225\ pounds \qquad (7)$$

The horsepower hpof any drive is defined in terms of ft-lb. per second or per minute (ft-lb/min):

$$1\ hp=550ft\text{-}Ib/sec=33,OOOft\text{-}Ib/min \qquad (8)$$

For example, a calculation can tell us what horsepower is required in a motor used to drive a 6-foot robot arm lifting a 50-pound weight at 60 degrees per second. We will assume that the arm has zero mass.

Conversion of w degrees per second (deg./sec) to rpm or revolutions per minute (rev/min) is done by

$$N = w\ deg/sec\ X\ (\ 1\ rev/360deg\ )\ X\ (\ 60\ sec/min\ ) =w/6 \qquad (9)$$

Using equation (4) and substituting N from equation (9) when w = 60,we obtain

$$hp = \frac{TN}{63,000} = \frac{50\ X\ (6\ X\ 12)\ X\ (60/6)}{63,000} = 0,576\ hp \qquad (10)$$

The use of simple equations of this type is often sufficient to make a useful approximation of a needed value. More detailed calculations can take in all of the appropriate data, using the equations of static and dynamics that apply.

CHAPTER 3

ROBOTIC SENSORY DEVICES

In this chapter we described the operation of a variety of sensor devices that either are now used on robots or may be used in future. In general, it is found that some are inherently digital devices, whereas others are essentially analog in nature. Sensors can be divided into two basic classes. The first called internal state sensors, consist of devices used to measure position, velocity or acceleration of robot joints and/or the end effector.

The second class called external state sensors is used to monitor the robots geometric and/or dynamic relation to its tasks, environment, or the objects that it is handling. Such devices can be of either the visual or nonvisual variety. In this chapter we discuss techniques that perming the monitoring of (1) distance from an object or an obstruction, (2) touch/slip and (3) force/torque.

3.1 NONOPTICAL-POSITION SENSORS

In this section we discuss the operation and applications of simple internal state sensors that can be used to monitor joint position. Included are the, synchro, resolver, and LVDT. It will be seen that some of these devices are inherently analog and some are digital in nature.

3.1.1 Synchro

As mentioned above. a significant practical problem with the pot is that it requires physical contact in order to produce an output. There are. However, a variety of sensing devices and techniques that avoids this difficulty. The first one that we discussis the syncro a rotary transducer that converts angular displacement into an ac voltage or an ac voltage into an angular displacement. Historically this device was used extensively during World War II, but technological innovations that produced other position-sensing elements caused it to fall from favor. In resent years, however, advances in solid-state technology have again made the synchro a possible alternative for certain types of systems. among them robots.

Normally. a synchro systems is made up of a number of separate three phase components (e.g., the control transmitter (CX), control transformer (CT), and control differential transmitter (CDX)]. These elements all work on essentially the principle of the rotating transformer. Typically, two or three of the devices are used to measure angular position or the difference between this and a command Position (i.e., the position error). For example, consider the two-element system shown in Figure 3.1.2. It is observed that an ac voltage is applied to the rotor of the CX and that the we-configured stators of the CT and CX are connected in parallel. Using elementary transformer theory, it can be shown that the magnitude of the transformer rotor voltage Vou(t) is dependent on the relative angle θ between the rotors of the CX and CT. In particular, this output voltage is

$$\text{Vout}(t) = \text{Vm} \sin\theta \sin \text{Wact} \qquad (3.1.1)$$

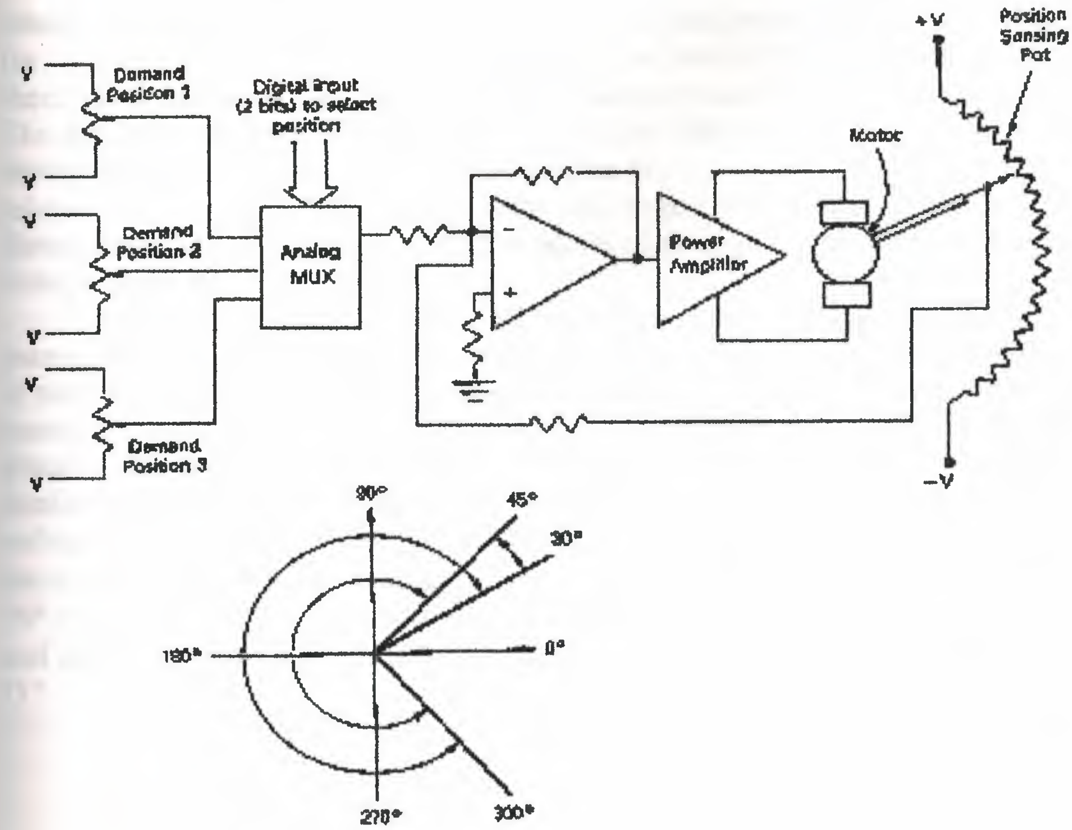


Figure 3.1.1. Teachable stops points using pots. The three pots are set to output voltages that will rotate the motor's shaft 30,15, or 300 °(all with respect to zero degrees) depending on the digital input to the analog MUX. Other moves are also possible with this configuration including 15° (from 30 to 45°) or 270° (30 to 300°).

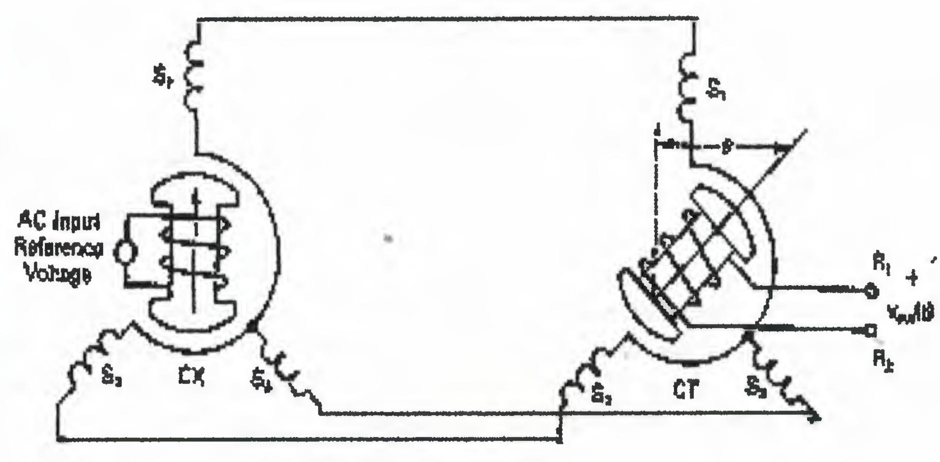


Figure 3.1.2. A two-element (control transmitter CX and control transformer CTsynchro system used to measure angular displacement. θ is the relative angle between the rotors of CX and CT.

where Vm and Wac are, respectively, the amplitude and radian frequency of reference (or "carrier") ac voltage. Those readers familiar with elementary communications theory will recognize that Eq. (3.1.1) represents an amplitude-modulated function. The difference between the radio AM and synchro AM signals is, of course, that the modulation of the carrier in the latter case is due to
relative angular position θ of the CT rotor with respect to that of the CX rotor. In the former case, however, the modulation is achieved through the application of another voltage signal that varies with time.

From Eq. (3.2.1) and Figure 3.2.2 it is seen that the output voltage has its maximum magnitude when the two rotors are at right angles to one another and that it is zero when they are at either parallel or antiparallel. As a consequence the CT is sometimes referred to as a "null detector." It is important to understand that in practice, the null is never exactly zero when the two rotors line up because of nonlinearities and electrical imbalances in the windings. These can produce "residual voltages" on the order of 60 mV (for a 115-V ac input). Due to the mathematical nature of a sine function , Vout(t) will be approximately linearly related to θ if -70° <= θ <= 70°. It is for this reason that where a linear relationship between output and angular position is important the synchro must used about an operating point θ= O °.

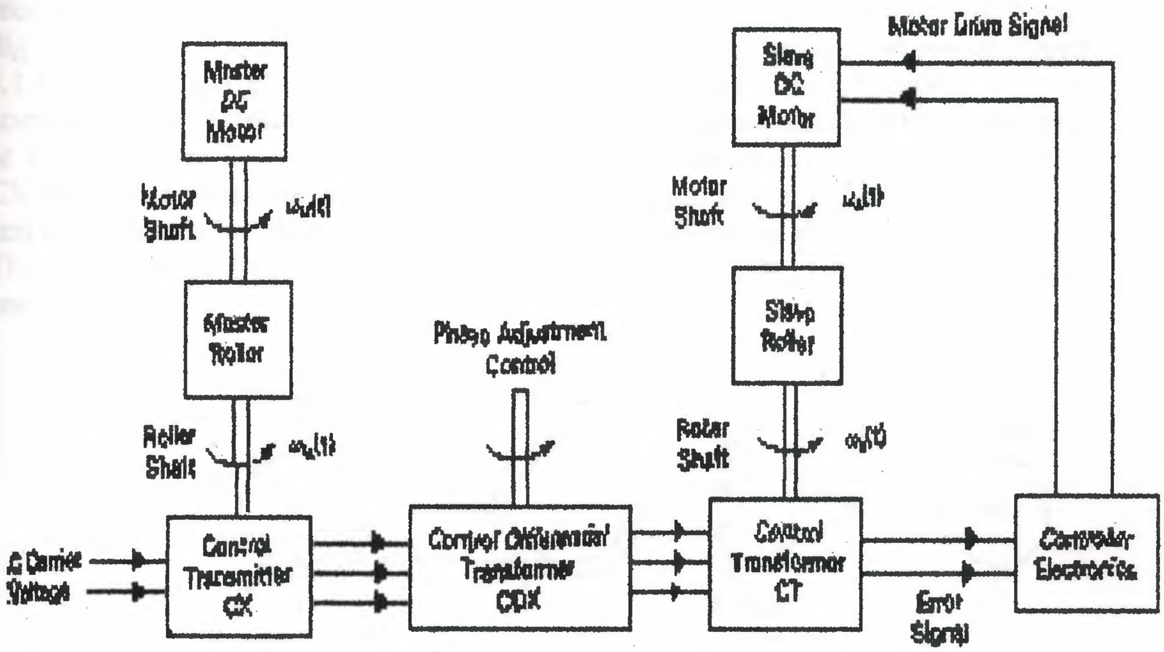


Figure 3.1.3. An example of a servo using a three element synchro system. To maintain a uniform product (e.g., steel sheets), the slave rolter's speed Ws(t) must be synchronized to that of the master, i.e., Wm,(t). The CDX is used to provide the desired angular relationship between the master and slave. The output signal of the CT is the difference between this desired and the actual master-to-slave angles, i.e., the error, and is used to provide the slave motor drive signal.

Ideally, the ac signals from the CX are in phase with those produced at the CT. However, physical differences in the structures of the two devices that are inevitably present produce phase shifts that may be undesirable. A synchro control differential transmitter (CDX) is sometimes used to adjust the phase shift between the two-synchro units. Such a device may also be used to produce a variable phase shift in applications where this is required. This is illustrated in Figure 3.1.3 Here the angular relationship between the master and slave rollers can be adjusted during the running of the process by rotating the shaft of the CDX

The use of a two-element synchro in a classical position servo application is illustrated in Figure 3.1.4. It is observed that the command or input (i.e., the angle $\theta 1$) will produce a command voltage from the CX. The CT will then produce an error voltage in accordance with Eq. (3.1.1), where $\theta = \theta l - \theta 2$. This error signal is amplified and causes the servomotor to rotate until θ is again zeroed. In such an application, the two-element synchro provides a rugged, reliable, and cost-effective method of the monitoring position error. However, the reader can readily appreciate that because of the need to convert the command position into a physical angular rotation of the CX rotor, such a system is not always practical in applications requiring the interfacing to digital devices. Thus, as mentioned above, it is not surprising that with the advent of microprocessor-controlled systems, synchros were quickly discarded in favor of other position-sensing methods more compatible with digital systems.

Recently, however, a number of advances in digital and hybrid technologies have produced a variety of devices that permit synchro systems to be easily interfaced with digital systems. For example, the digital-to-synchro (D/S) converter shown in Figure 3.1.5 replaces the CX in the position servo of Figure 3.1.4 A digital position command signal from a computer (e.g., the master) is transformed into a three-phase ac voltage by the D/S converter. (This voltage corresponds to that produced by the CX due to a physical rotation of el.) The CT once again acts as a position error sensor and the system behaves in a manner that is identical to that of the one in Figure 3.1.4 The use of the D/S converter produces a position servo that is part digital and part analog.

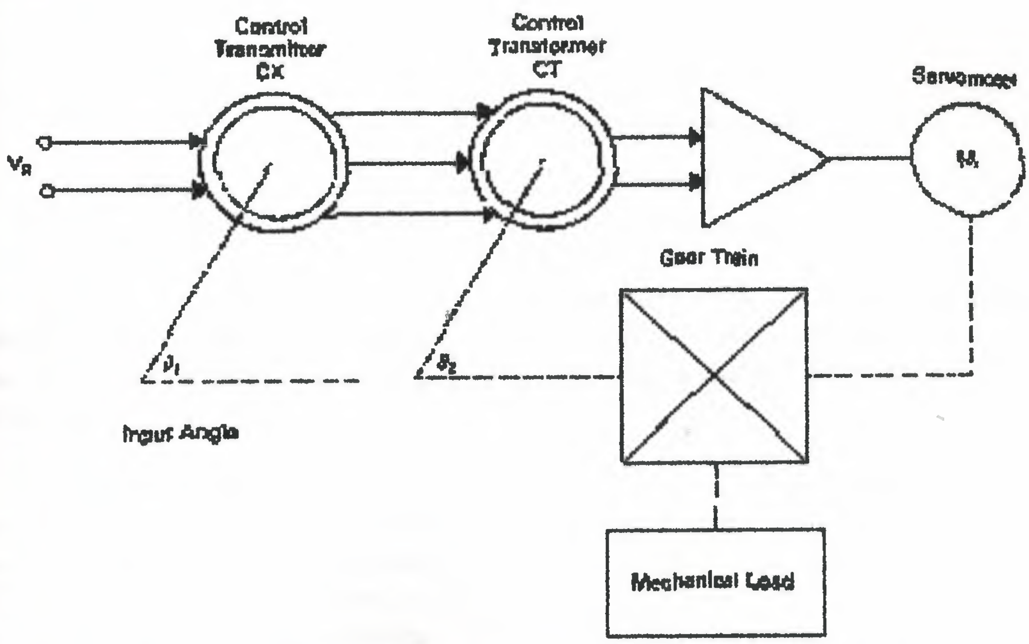


Figure 3.1.4 A synchro used in a position servo loop. The desired angular position is θ1, whereas θ2, is the actual angular position of the motor shaft. VR is the ac reference (carrier) voltage.

3.1.2 Resolvers

The resolver is actually a form of synchro and for that reason is often called "synchro resolver." One of the major differences between the two devices is that the stator and rotor windings of the resolver are displaced mechanically 90° to each other instead of 120° as is the case with the synchro. The most common form of resolver has a single rotor and two stator windings, as shown in Figure 3.1.5 With the rotor excited by an ac carrier voltage B sin watt, the two stator voltages become

$$V1\text{-}3\ (t) = V \sin \theta \sin Wac\ t \tag{3.1.2a}$$

$$V2\text{-}4\ (t) = V \cos \theta\ \mathrm{Sln}\ Wa\ t \tag{3.1.3b}$$

where θ is the resolver shaft angle. It should be clear to the reader that such a device could, and often is, used in much the same way as the synchro CX to monitor shaft angle.

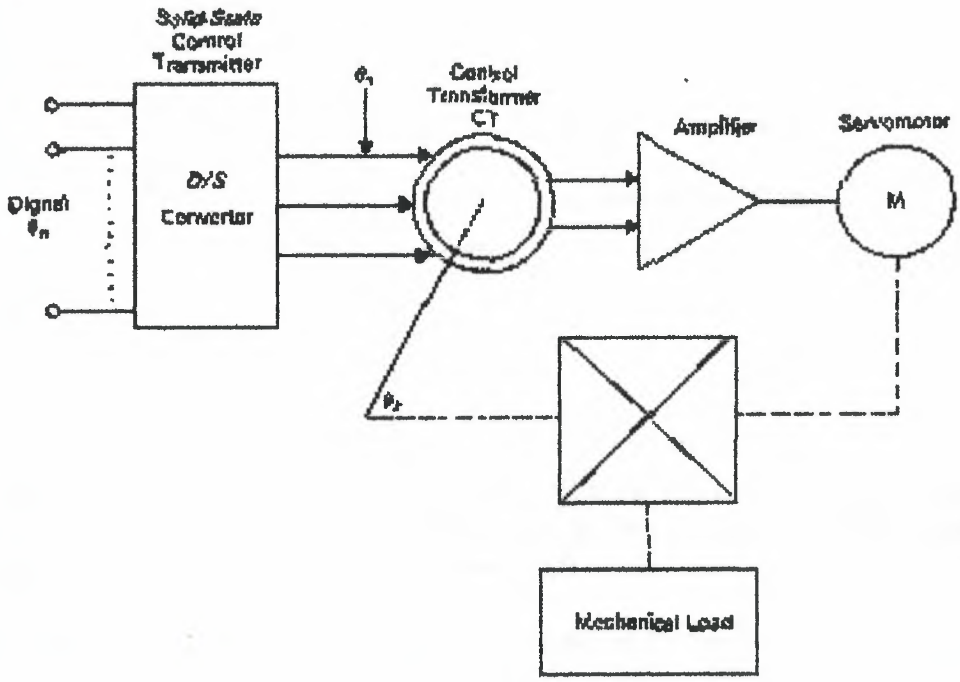


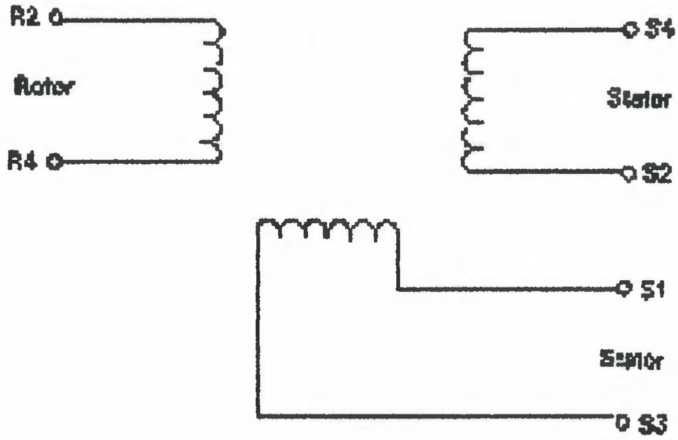
Figure 3.1.5. The synchro control transmitter CX of the position servo shown in Fig. 3.1.4. is replaced by a D/S converter. This scheme perroits the desired input θin to be a digital quantity, i.e., makes the system microprocessor compatible.

An alternative form of a resolver has two stator and two rotor windings. In actual use, the carrier voltage may be applied to any of these. For example, if the former is used as an input, the unused stator winding is normally shorted . The output voltages are identical to those given in Eqs . (3.1.2.a) and (3.1.3.b) and are monitored across the rotor windings. Alternatively, one rotor winding can be used as the input with the two stator windings being used as the outputs.

To utilize a resolver in a servo system, it is usually necessary to employ two resolvers in much the same way as was done with the synchro system of Figure 3.1.3. Figure 3.1.7 shows a resolver transmitter (RX) and resolver control transformer (RT)

in a simple position servo. Again, the reader should note that RX and RT are used to obtain the difference between the actual and desired angles
(i.e., θ1, - θ2). It is important to understand that although angular position can be monitored using a single resolver (see Eqs. (3.1.2.a) and (3.1.3.b), this is usually not done in servo-controlled devices because of the need to utilize an error signal to drive the system actuator.

As in the case of the synchro, there has recently appeared a series of special purpose chips that permit one of the elements of a resolver servo system to be eliminated. For example, the Analog Devices Solid-State Resolver Control Transformer (RSCT 1621) shown in Figure 3.1.8. can be used in place of an RT. As can be seen, a 14-bit digital representation of a command input θ and the analog output of an RX, representing the actual angle θ, are input to the D/R converter .The output of this device is then an analog voltage that is proportional to θ - ϕ. This chip is a hybrid since it not only includes the digital and analog circuits necessary to process the two input angles but also has on board the appropriatte input and output transformers. The only significant difference between a D/R and a D/S converter is in the transformer configurations.



3.1.6. Electrical circuit of a simple resolver. With an ac carrier volt age input to the rotor, the output voltage amplitude of the two stator windings will be dependent on the sine or cosine of resolver shaft angle

A position servo utilizing such a chip is shown in Figure 3.1.9. Note that since the output of the D/R converter (or DRC) is an ac voltage, it is necessary to use an ac amplifier, together with a phase-sensitive detector and integrator to obtain the appropriate drive signal to the servo amplifier. As in the case of a comparable synchro system, this servo is functionally a hybrid since the command signal is digital, whereas the monitored position (and the error) is analog in nature.

It was seen in Chapter that it is often convenient, in the control systems used in robots, to have a digital representation of the actual angular position of either the actuator shaft or the joint itself. The tracking RDC shown in Figure 3.1.10. accomplishes this. Here the RX is connected, either directly or through a gear train, to the shaft that is to be monitored. The converter then "tracks" the shaft angle outputting a digitized version of it. Thus it can be seen that the RDC takes the place of both an RT and an ADC. Unlike the ADC, however, the tracking RDC automatically performs a conversion whenever the input voltage from the RX changes by a threshold value, as determined by the resolution of the RDC. Note that unlike many A/D converters, there is no need to trigger the R/D externally.

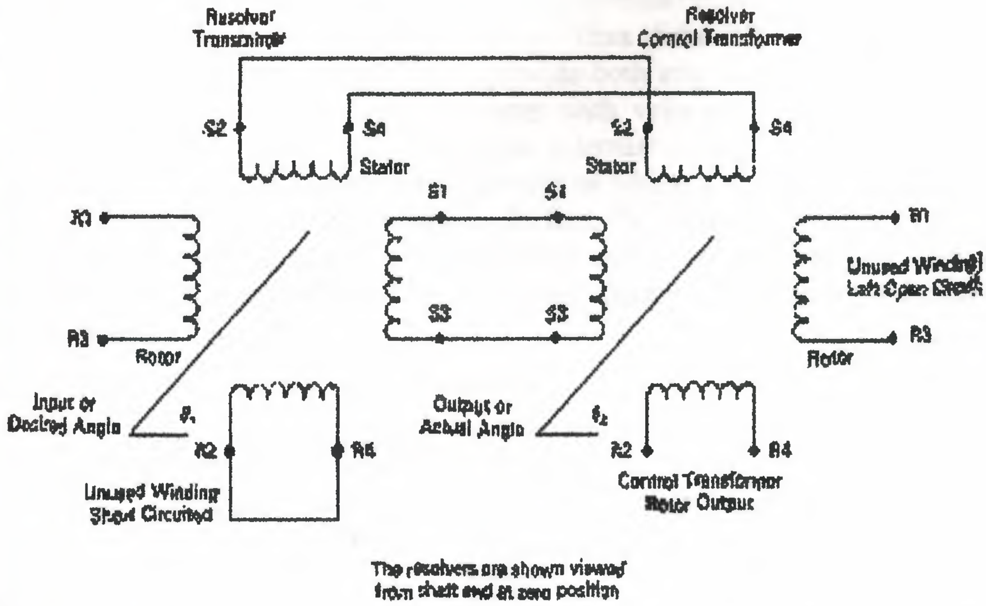


Figure 3.1.7. Resolver transmitter connected to a resolver control transformer.

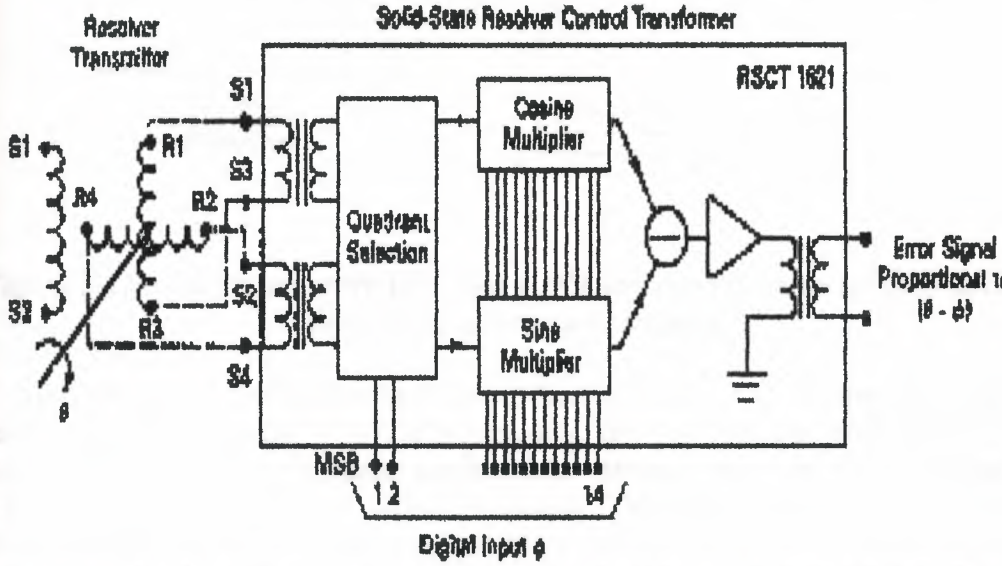


Figure 3.1.8 Resolver transmitter and RSCT 1621 Solid-State Resolver Control Transformer functional diagram. Use of this hyhrid device permits elimination of separate input and output transformers. θ is the measured or actual angular position and ϕ is the desired angular position.

Tracking synchro-to-digital (S/D) eonverters are also now available. The only difference between these devices and the RDC discussed above is that configuration of the input transformer on the chip is different since it must accept a three-phase rather than a two-phase voltage. Insofar as the user is concerned, however, the devices are identical.

The reader may wonder whether there is an advantage to using a tracking R/D (or S/D) converter over a D/R (D/S) device since the decision seems to depend on the nature of the servo system configuration. However, with reference to Figure3.1.10. it is observed that the R/D converter has a velocity output. Available on many types of tracking R/Ds (or S/Ds), the analog voltage at this terminal represents the time rate of change of the input angle θ or angular velocity. Thus if the RX is connected to the shaft of a robot joint actuator, the R/D will provide both angular position and velocity information. Since an R/D (or S/D) converter with velocity output provides this information, it seems to represent a possible alternative to a system that utilizes separate sensors to monitor position and velocity or where a digital representation of velocity is found from the digital position. In fact, the velocity signal output of the R/D has better low-speed behavior and is more nearly linear than other velocity sensors and produces a better overall velocity signal than that derived from digital Position-sensors.

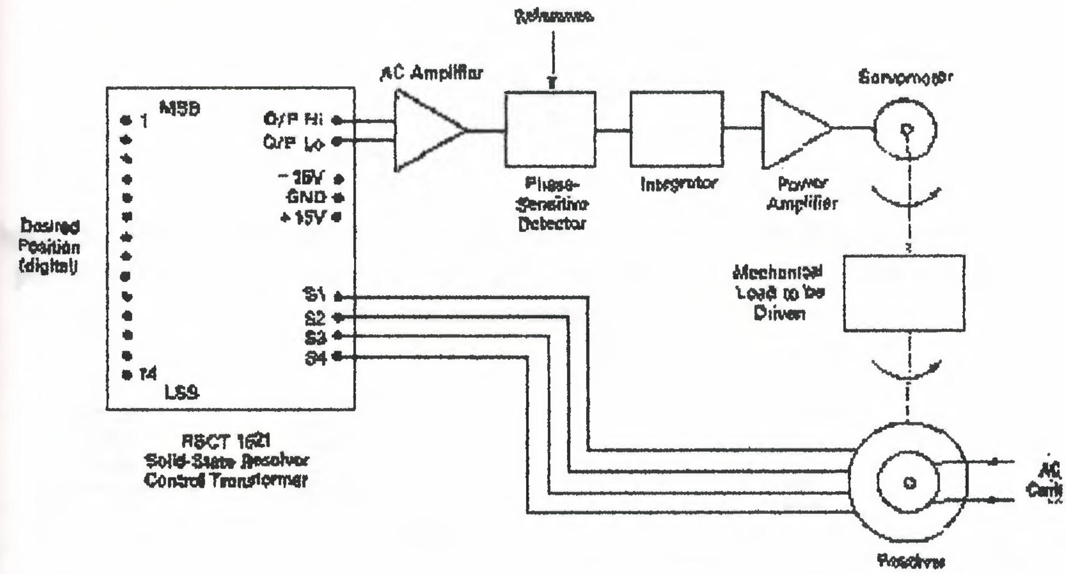


Figure 3.1.9. A position servo that uses a D/R converter. An ac amplify because the D/R output is an ac voltage.

Application of such a device in a position servo used to control one axis of a robot is shown in Figure 3.1.11. Theoretically, the development of R/D, S/D, D/R, and D/S chips to monitor angular position and velocity appears to make the resolver or synchro attractive for use in robotic systems. The rugged nature of the RX or CX is a particularly useful trait, and it is possible to obtain resolvers or synchros that have better angular resolution (i.e., the ability to sense smaller angular increments) than most other position sensors (e.g., incremental encoders or pots). However, there are a number of reasons why these devices are not commonly used in robots today. The first is cost since the converter chip and RX (or CX) combination is usually significantly higher than that of an optical encoder package (including the associated electronics) for the same resolution. A second is the potential problem of electromagnetic interference (EMI) due to the ac carrier signal. Although this problem can be overcome, careful shielding of certain critical subsystems of the robot controller is generally required to accomplish this and thus manufacturing costs may be increased. Finally, it is usually necessary to bring out a larger number of wires than with other position-monitoring techniques. This can be especially trouble-some with the moving joints of a robot. All things considered, however, it seems logical that if the overall cost of resolver/synchro systems is reduced, such devices will indeed be used in robot servo systems.

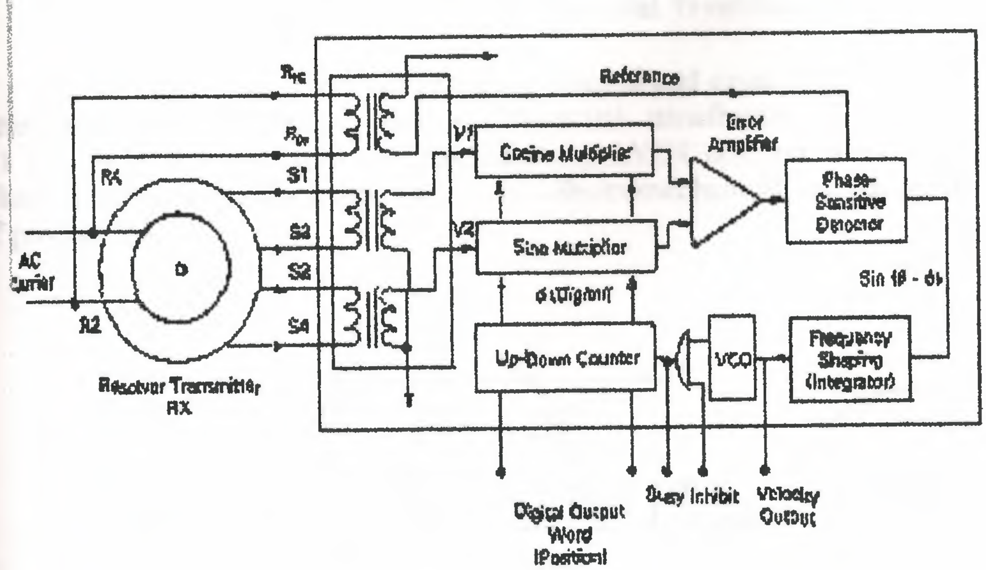


Figure 3.1.10. A tracking resolver-to-digital converter. The RX senses the actual position θ. The chip outputs a digitized version of this angle. The velocity is dθ/dt and is an analog quantity.

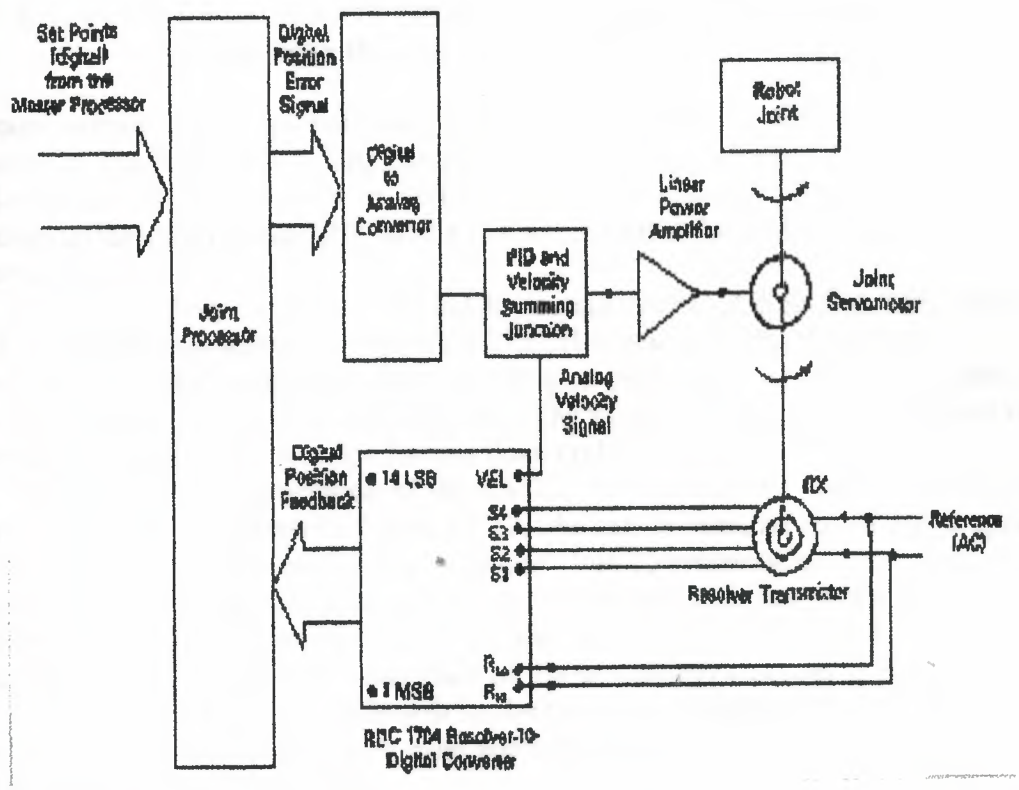


Figure 3.1.11. Tracking R/D converter with velocity-output used in the position servo of a single robotic joint. This is a hybrid servo since the position information is digital whereas the velocity information is analog.

3.1.3. Linear Variable Differential Transformers

Another device that is both extremely rugged and capable of accurate position determination is the linear variable differential transformer (LVDT; see Figure 3.1.11). It is observed from this figure that the LVDT consists of two parts, one of which is movable and the other fixed. This electromechanical transducer is capable of producing a

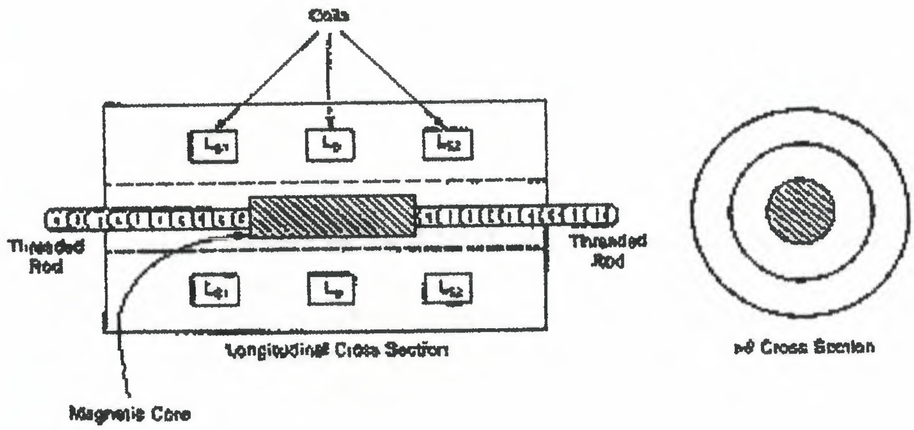


Figure 3.1.12. A linear variahle differential transformer (LVDT) showing the single primary and the two sets of secondary coils. The magnetic core is generally the moving element of this sensor.

voltage output that is proportional to the displacement of the movable member relative to the fixed one. Units having sensitivities on the order of 1 mV/mil with full-scale ranges of ±25 mils to several inches are available. Because LVDTs are analog devices, they essentially have a resolution that is limited only by the external monitoring device .

A common design of the LVDT has three equally spaced coils (Lp, Ls1, and Ls2) on a cylindrical coil form (see Figure 3.1.12). This is usually the stationary element. A rod-shaped magnetic core is also positioned axially inside the coil assembly and is free to slide back and forth. The purpose of this moving element is to provide a magnetic path for flux linking three coils.

To understand the operation of the LVDT, we consider the equivalent electrical circuit of the device shown in Figure 3.1.13. As can be seen, an ac voltage is applied to LP, the primary side of the coil structure (this corresponds to the center coil in Figure 3.1.12). Since Lsl and Ls2 on the secondary side are connected in series opposing (note the position of the dots on the windings), Vout (t) will be zero if the coupling between the primary and each of the secondary windings is the same (i.e., the voltage induced in these coils will be the same). A little thought should convince the reader that this condition will exist when the magnetic core is positioned exactly in the center of the coil assembly.

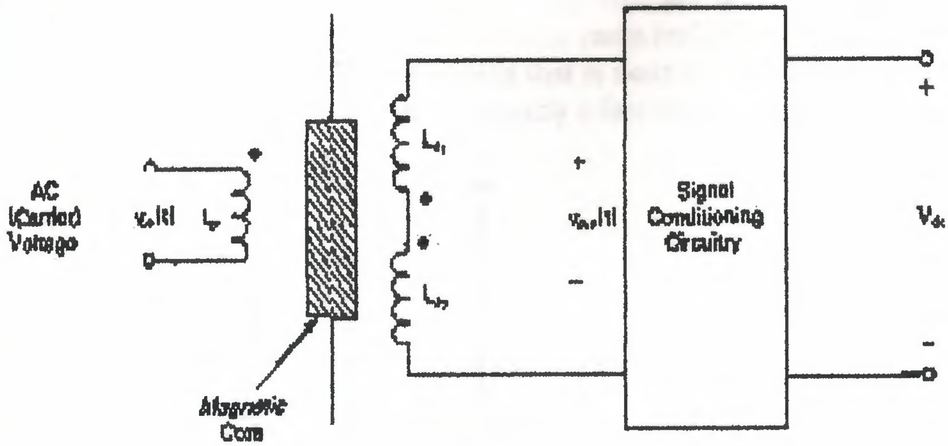


Figure 3.1.13. Electrical circuit of an LVDT showing the magnetic core. The secondary coils are connected in series opposing so that when the core is at or near the center of the LVDT Vout (t) is zero.The signal conditioner is used to demodulate Vout (t) and produce a dc voltage that is proportional to the core's linear distanceaway from the null (center) position.

If, however, the core is moved away from the central position, the coupling between Ls1 and Lp will differ from that of Ls2 and Lp. For example, the former will increase, whereas the latter will decrease. Consequently, the voltage induced in Lsl and Ls2 will increase and decrease, respectively, with respect to their center core values. Thus Vout (t) will be nonzero. With reference to Figure 3.1.12, it principle as the LVDT but has a rotational configuration. When it comes to rob however, neither device has been much used because of a number of problems.

The first difficulty with the LVDT in a robot application is that it is necessary to operate the device about its center point. Such an alignment is often quite difficult to perform and could create major manufacturing problems, thereby creasing costs. Difficulties with calibration of the robot on power-up could also be encountered. A second and associated problem with the LVDT is that center or "null" position has a tendency to drift with time and temperature. Unless this is prevented (by the use of appropriate temperature compensation) a change in the robot s calibration would result, a totally unacceptable occurrence in most manufacturing environments. Possibly the major difficulty with using an LVDT and/or RVDT as a position sensor is that the joints of most robots do not move in pure straight lines or circular arcs. Motion is normally a complex combination of these trajectories. As a result, it is quite difficult, if not impossible, to configure the LVDT/RVDT so that the magnetic core is always collinear with the axis of the coils. Note that binding and probable damage to the unit will result if collinearity is not maintained. Finally, the RVDT has one additional limitation that makes its application to robots problematical. That is it can only sense rotary motions of approximately ±60 . Since most robotic axes are re uired to move more than 120°, this is most certainly a severe restriction.

Even if these difficulties were to be overcome the fact that the LVDT is an analog device would make it inconvenient to utilize the device in a microprocesscor controlled servosystem. In addition, the cost of the sensor, signal-conditioning circuitry, and A/D converter would be, no doubt, significantly higher than that of an equivalent optical incremental encoder and its circuitry. Thus it seems likely that despite its good resolution capabilities, the LVDT will not soon be used on robots themselves as an internal position sensor. It seems more realistic to expect that such devices will be utilized in equipment that is used in conjunction with a robot (e.g., in parts presentation). Indeed, this is already a fact in some instances.

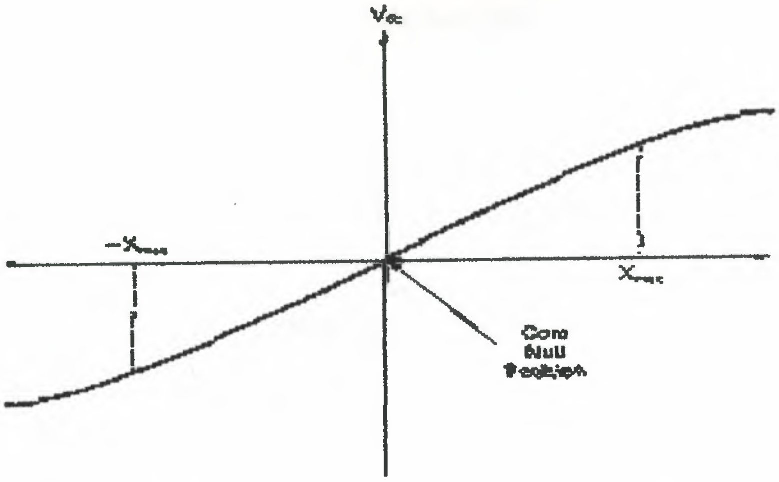


Figure 3.1.14. Output of an LVDT signal conditioner as a function of the core position x. For |x|<= X max the characteristic is extremely linear. Outside of this range, however, linearity suffers.

3.2 OPTICAL POSITION SENSORS

As we have seen, the sensors discussed in the previous sections can theoretically be used to determine the position of a robotic joint. However, for one or more practical reasons, doing so is not possible or often difficult and/or inconvenient. Another class of sensor, utilizing Optical hardware and techniques can quite frequently be used to perform the position determination task with relative ease and surprising accuracy. We now discuss such devices and their application to robotics.

3.2.1 Opto-Interrupters

It will be recalled that point-to-point-type robots require only that the beginning and end points be accurately determined. The actual path between these points is not important, and hence little or no position information is utilized by the robot's control system except at the trajectory endpoint. The actuators drive the joints of the robot until the final position is sensed, at which time the actuating signals are removed. In effect, an open-loop control scheme is used. "Programming" is accomplished by moving the endpoint sensors to different locations.

It might appear that a simple mechanical switch (or micro switch) is an ideal device for this application. However, because of the need to interface the switch with a microprocessor, the inevitable contact bounce problem and the limited life expectancy make this approach relatively impractical for commercial robots.

An optical technique can be used to produce the required ability to sense "end of travel" without the problems associated with mechanical switches. Called an opto-interrupter, its operation is quite easily understood. Consider the arrangement shown in Figure 3.2.1.A transparent disk with at least one dark sector is placed between a light emitter (e.g., an LED) and a light receiver or sensor (e.g., a phototransistor). Light will reach the receiver until rotation of the disk causes the "black flag" to block it. A binary or "on-off" signal can be generated anti used to sense the endpoint of travel. For example, the output (i.e., the collector) of the phototransistor will be low as long as light impinges on the transistor's base. On the other hand, the collector voltage will be high when there is no light.

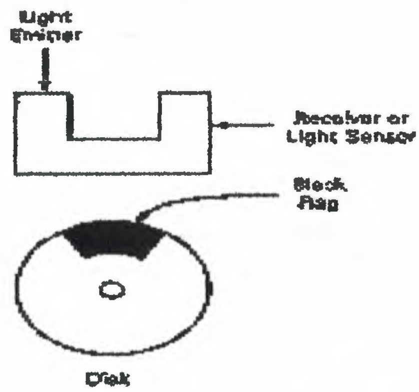


Figure 3.2.1. Simple opto-interrupter showing light emitter-receiver assembly and and disk with "black flag."

The block diagram of a simple electronic circuit that makes use of such a sensor to drive a robot axis to the end of travel is shown in Figure 3.2.2. Here the system is actuated by momentarily closing the start switch. The motor will continue to rotate until the black flag on the disk prevents light from reaching the light sensor. When this occurs, the motor voltage is turned off and the axis coasts to a stop.

A possible realization of the logic and sensor electronics is shown in Figure 3.2.3. The waveforms of the digital signals SI, S2, and S3 are shown in Figure 3.2.4. To understand the operation of this circuit, recall that the output of a NAND gate will be low (i.e., 0 volts or "logical zero") only when both inputs (Si and Sz in this instance) are high (i.e., "logical 1" or for TTL logic circuits, 5 V). Any other combination of input signals will cause the output of the NAND gate to be high. Thus if the black flag on the disk is initially placed in the slot of the opto-interrupter the collector

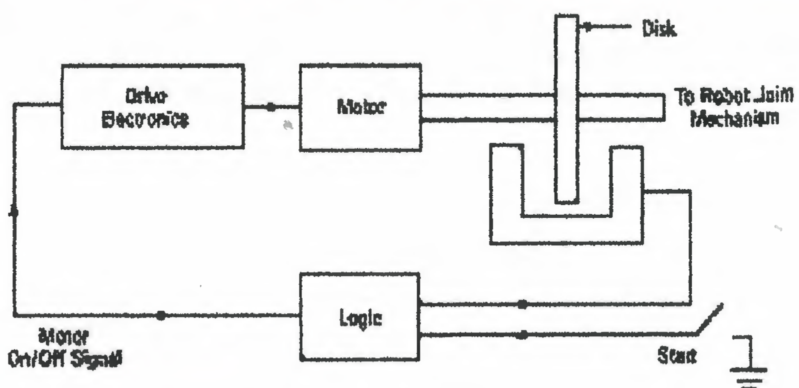


Figure 3.2.2. Block diagram of a simple unidirectional motor control circuit. The motor begins to rotate when the switch is closed.

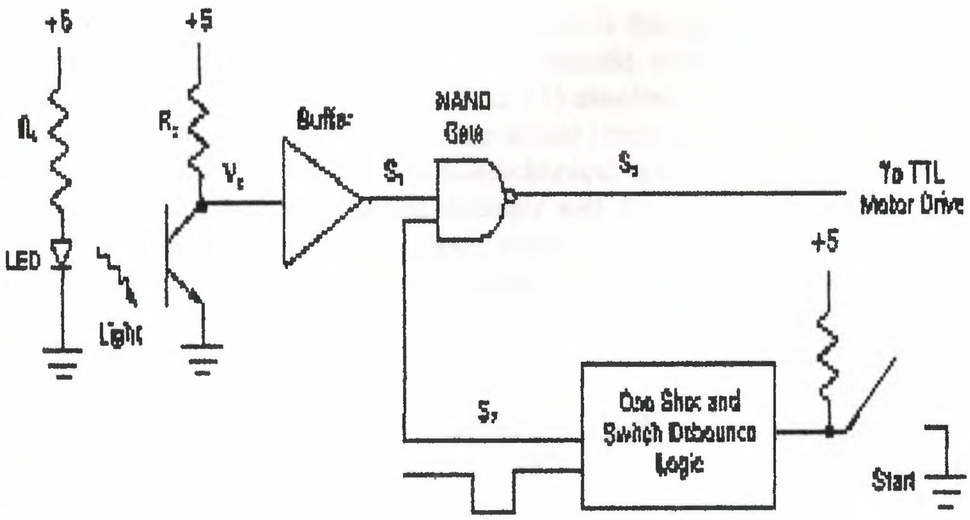


Figure 3.2.3. Possible realization of sensor and logic circuits for simple motor controller of Fig. 3.2.2.

Of the phototransistor will be about 5 V, so that St will be high. In addition, if the one-shot and debounce circuit is designed so that its output is normally high and goes low only when the one-shot is triggered by the start switch being grounded, SZ will normally be high also. Therefore, signal to the motor drive circuitry is low and the motor does not turn.

As seen in Figure 3.2.4, when the start switch is depressed, SZ goes low, which in turn causes S3 to go high. The motor begins to rotate and will continue to do so until the black flag again interrupts the light, reaching the base of the photo-transistor.

It is important to note that this simple circuit permits only unidirectional rotation of the motor. Thus if it were used to actuate an axis of a simple robot, the manipulator would be limited to motion in one direction only. More complex circuitry would be required to produce bi-directional motion. In addition, as shown in this example, such a robot would be quite limited since there would be only a single endpoint. More endpoints could be obtained simply by utilizing more than one flag placed at appropriate places on the disk. In fact, "programming" such a robot axis would consist of producing a special disk with the correct number of flags at the proper locations.

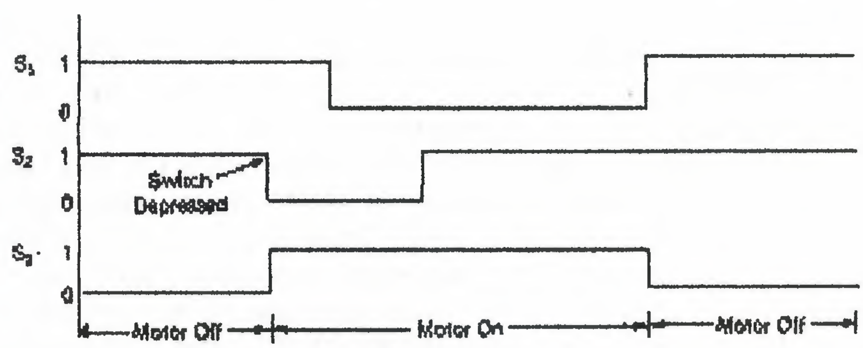


Figure 3.2.4. Timing diagram for the digital signals associated with the sensor and logic circuit of Fig. 3.2.3.

3.2.2 Optical Encoders

One of the most widely used position sensors is the optical encoder. Capable of resolutions that are more than adequate for robotic applications these noncontact sensory devices come in two distinct classes: (1) absolute and (2) incremental. In the former case, the encoder is able to give the actual linear or rotational position even if power has just been applied to the electromechanical system using the sensor. Thus a robot joint equipped with an absolute encoder will not require any calibration cycle since the controller will immediately, upon power-up, know the actual joint position.

This is not so in the case of the incremental encoder, however. Such a sensor only provides positional information relative to some reference point. A robot utilizing an incremental encoder must, therefore, first execute a calibration sequence before "true" positional information can be obtained.

Although either linear or rotary encoders for both of the foregoing classes are available, the rotary device is almost exclusively used in robotic applications. One of the most important reasons for this is that revolute joints far outnumber prismatic ones in robots currently being manufactured. Even for joints that move in a linear fashion, as in the case of a spherical coordinate manipulator, the linear encoder is normally much more costly, and so rotary encoders are still employed. Therefore we restrict the discussion to the latter type, although much of what is said will apply directly to the linear sensor.

3.2.2.1 Rotary absolute encoders

As mentioned above, the absolute encoder is capable of giving the correct rotary position at all times even after power-up has occurred. The device produces a separate and unique coded word for each shaft position, and unlike the incremental encoder, every reading is independent of the preceding one. A major advantage of the absolute encoder is that even if system power is accidentally lost (due to a power outage or relay trip, for example) the device will "remember" where it is and will report this to the system as soon as power is restored. Calibration of machines using this type of encoder is therefore, maintained even if the position of the rotating member is moved when the power is off.

Absolute encoders usually consist of three major elements:

1. A multiple-track (or channel) light source
2. A multiple-channel light receiver
3. A multiple-track rotary disk

Normally, light emanating from a linear, N-element light source (e.g., LEDs) is made to pass right angles through the disk and is received (or collected) by a corresponding linear array of N light sensors (e.g., phototransistors) mounted on the opposite side of the disk (see Figure 3.2.5). The disk is divided into circumferential tracks and radial sectors. Absolute rotational information is obtained by utilizing one of several possible code formats. For example, Figure 3.2.6 shows a four-track 16-sector pure binary-coded disk Other coding schemes that can be used include binary-coded decimal (BCD) and Gray code. It can be seen from Figure 3.2.6 that the resolution of the disk is 22.5o (360I16) since one complete disk revolution is 360o and there are 16 sectors.

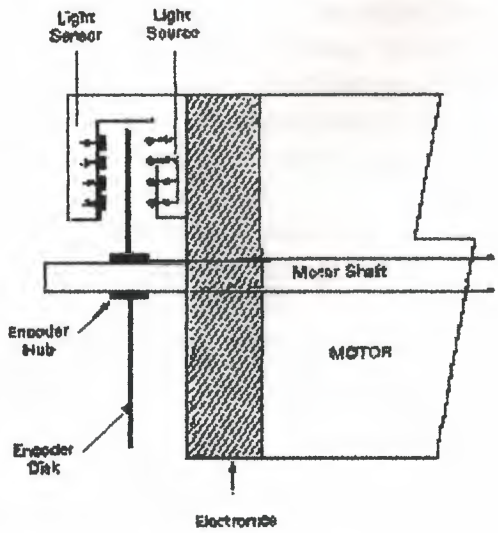


Figure 3.3.5. An absolute encoder mounted on a motor. Shown are the various components of the encoder including the disk, light sources, light detectors, and electronics.

If the shaded areas are assumed to represent a binary "1" and the clear areas a binary "0," the outputs of each of the four light sensors will represent a 4-bit sequence of ones and zeros. For the binary code used in Figure 3.2.6, the decimal equivalent of this number is the actual sector number. As an example, if sector 11 is in the region of the LEDs, the output of the photo transistors will be 1011 or decimal 11. It is clear from this discussion that the absolute disk position is known simply by reading the photo-detector outputs.

In ractice, it is possible to produce absolute encoders with u to 13 separate channels (i.e.,13 bits) which means that resolutions of up to 360/2(13) = 0.044° are possible for a single complete rotation of the disk.

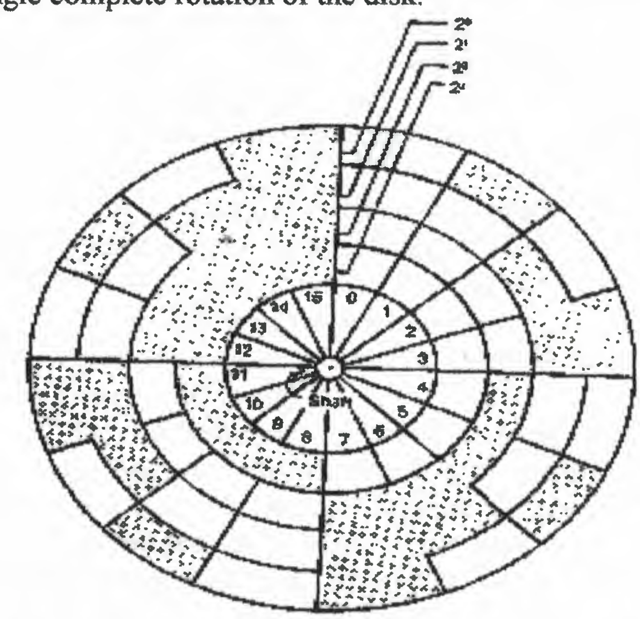


Figure 3.2.6. A four-trach, sixteen-sector pure binary-coded disk used in an absolute encoder.

Often, however, it is necessary for the device being monitored by the encoder to undergo many rotations. Since it is clear that the coded binary sequence repeats for each complete disk cycle, something else is needed. In this case it is possible to use a second disk placed on the same shaft as the first but geared down so that a complete revolution of the first moves the second only a distance of one sector. The first one is used for absolute positional information for any single shaft revolution whereas the second disk gives the actual rotation number.

Although their ability to "remember" position is an extremely attractive feature for robot applications, absolute encoders are generally not used because of their excessive cost. For example, a 10-bit device can run several hundred dollars. A comparable incremental encoder, on the other hand can be purchased for about $80 to $100. It is therefore not surprising that most robots utilize incremental devices, which we now discuss in some detail.

3.2.2.2 Optical incremental encoders

As mentioned above, optical incremental encoders are widely used to monitor joint position on robots. In addition they are the sensor of choice in a variety ofmachine tools, including lathes, x-y tables, and electronic chip wire and hybrid die bonders. The major reason is that they are capable of producing excellent resolution at a significantly lower cost than a comparable absolute device. However ,absolute position information can be obtained only by first having the robot or other machine tool perform a calibration operation.

If the encoder disk is mounted on a rotating shaft (e.g., of a servomotor as shown in Figure 3.2.5), then as the disk turns, light to the photodetectors will be interrupted by any line on the disk that passes in front of the LED source. It can be shown that the detector's output will be a waveform that is approximately sinusoidal. Often, a comparator is used to convert these signals to TTL pulses, thereby making them more suitable for digital systems. There are two problems with this arrangement. The first is that although a single photodetector will produce a sequence of N TTL pulses per revolution, it should be clear that it will be impossible to determine the direction of rotation of the disk. A second difficulty arises due to variation or drift in light source and/or ambient light intensity. Since a comparator is used for TTL conversion, the width of the pulses will be quite sensitive to the amount of light collected by the detector. This is an undesirable condition, especially in cases where the disk is spinning at a high rate of speed.

Both of these problems can be overcome by employing multiple light sensors. For example, a second photodetector separated from the first by 90° (electrical) will produce a second, or B output channel which is identical to the first, or A channel, except that it yields TTL signals approximately 90° out of phase with the original ones. Clockwise or counterclockwise rotation of a motor shaft can be determined simply by noting whether A leads or lags B .

The solution of the light-variation problem requires the use of additional photosensors. To understand this, consider the single-channel encoder (with only a small, magnified section of the disk indicated) shown in Figure 3.2.8. Here we have placed a stationary plate or reticle in front of the light sensor. This component consists of a number of optical "slits" (i.e., lines) and is used to direct light from about 20 lines on the encoder disk to the single photodetector. An overall improvement in performance is realized by reducing the encoder's sensitivity to both dirt and variation in line placement.

In actual operation, when the disk is rotating, the photosensor voltage output will vary theoretically in a triangular fashion, as shown in Figure 3.2.9. Actually, the waveform is more nearly sinusoidal, primarily due to the finite line widths in the shutter assembly (i.e., the disk and reticle). The maximum sensor output voltage Emax is proportional to the intensity of the LED. The minimum voltage Em;" is not zero because light cannot

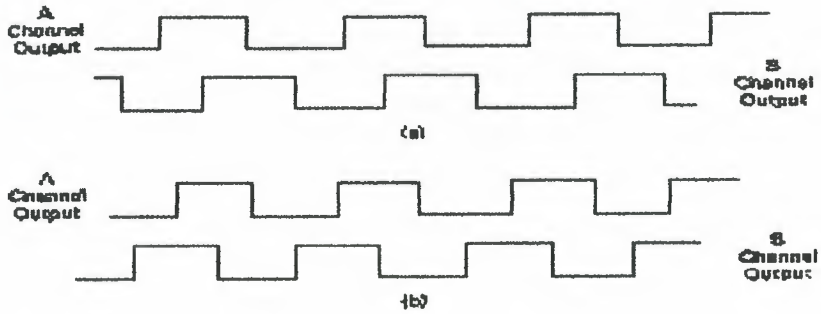


Figure 3.2.7. TTL outputs of the A and B channels of an incremental encoder: (a) A leads B when clockwise rotation occurs; (b) A lags B when counterclockwise rotation occurs

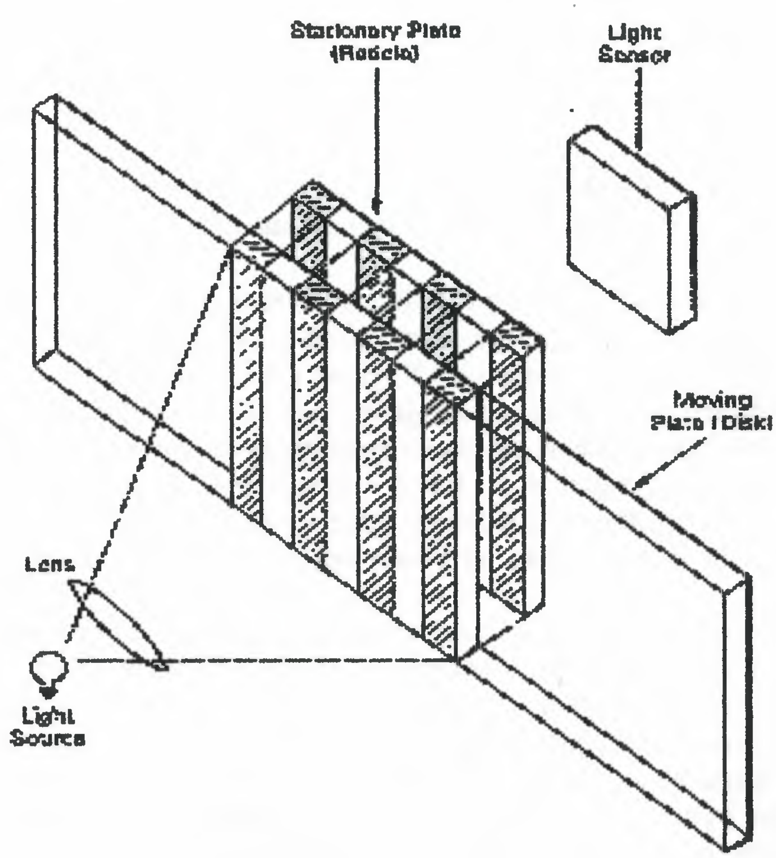


Figure 3.2.8. Section of a single channel encoder.

be fully collimated by the shutter (i.e., there is always some light leakage). This value can be minimized, however, by reducing the clearance between the shutter and the light source (e.g., a 1- to 10-mil gap is typical). It is desirable to do this because the usable component of the sensor output is the peak-to-peak value E1.

If a comparator is used to digitize the sensor output signal, a TTL pulse will be generated each time the voltage passes above the average value Eave. This will theoretically produce a train of pulses with a □0% duty cycle provided that the disk is rotating at a constant velocity (see Figure 5.3.10a). However, if Eave drifts due to LED and/or ambient light intensity variation or photodetector sensitivity changes (caused by elevated temperature or high-frequency operation), the pulses will no longer have a 50% duty cycle, as shown in Figure 5.3.10 b. Although at low speeds this is not a problem, high-speed applications will cause the pulses to be so narrow as to produce sensing errors (i.e., pulses may be missed).

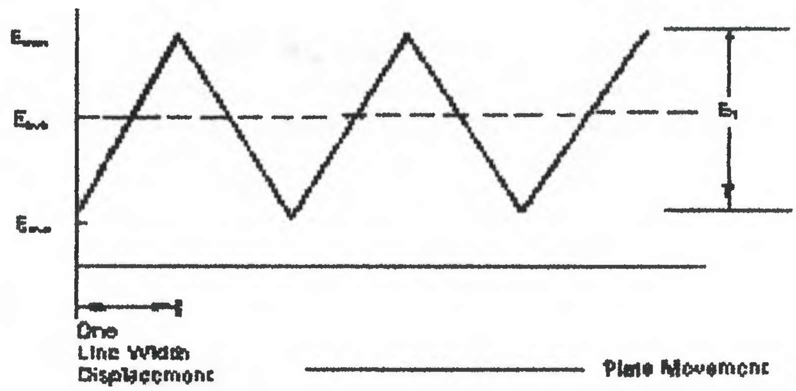


Figure 3.2.9. Output voltage of the sensor in Fig. 5.3.9 as the encoder disk moves relative to the reticle.

This problem can be overcome by employing a second sensor (and reticle) placed 180° out of phase with the first , as shown in figure 3.2.11. Note that the same light source is used to illuminate both sensors. If the outputs of the two photodedectors are connected in "push-pull" so that the two signals are subtracted , a triangular waveform centered zero and having approximately twice the peak-to-peak amplitude of either signal will be generated (see Figure 3.2.12.). In practice , differences in the two sensors cause the average value to differ somewhat zero from. However this is a second-order effect and can easily be offset with a bias voltage applied directly to the difference amplifier.

The push-pull configuration has a number of advantages over a single sensor device. First and most important, the optical encoder is much less sensitive to variations in the average value of the photodetector output since the light sensor will be equally affected. As a direct consequence , the interpulse spacing variation (at constant velocity) is reduced to about one-half that found for a single sensor unit for the same drift in average light intensity. In addition, temperature and/or frequency effects are minimized because, once again, both sensors are affected to the same degree.

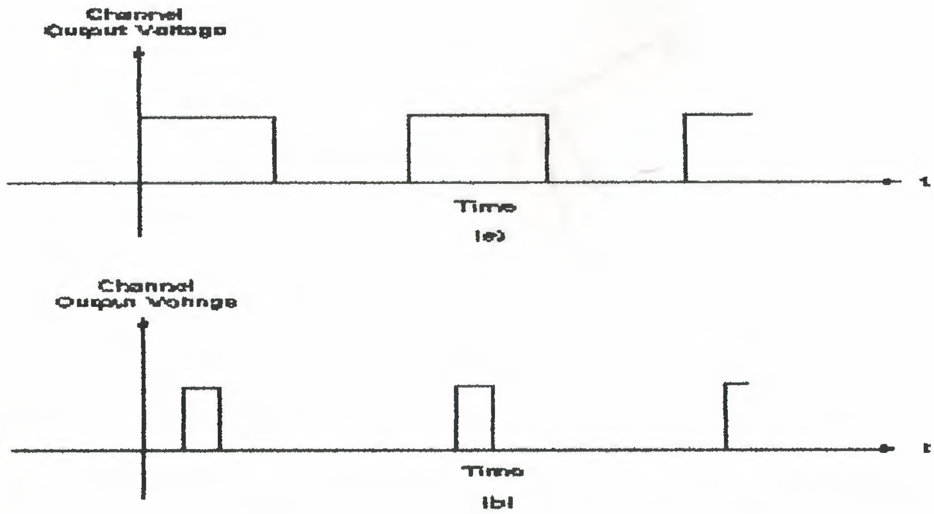


Figure 3.2.10. TTL output of a single encoder channel. The disk is assumed to be rotating with a constant angular velocity: (a) ideal 50% duty cycle; (b) the duty cycle is not 50% due to drift in the average value of the sensor voltage Eave.

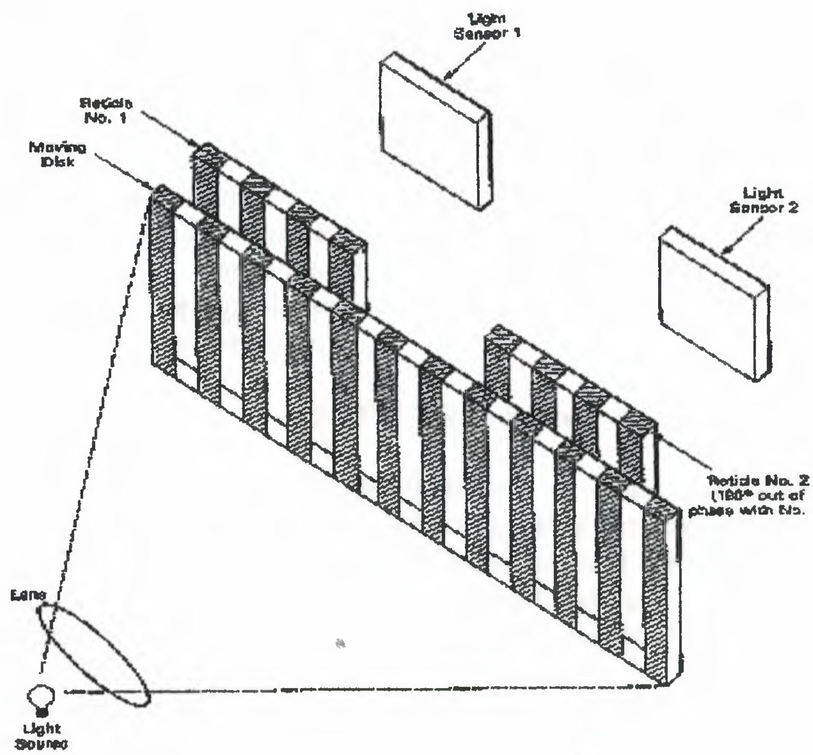


Figure 3.2.11. Use of two sensors with the same light source to reduce the variation in the average value of the encoder's output, Eave.

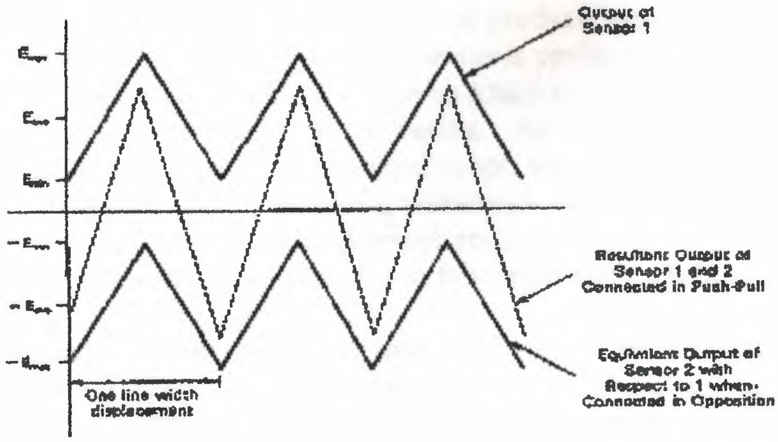


Figure 3.2.12. Push-pult output from two sensors. This arrangement significantly reduces the effect of average value drift on the TTL duty cycle.

3.3 VELOCITY SENSORS

A robotic servo must make use of both position and velocity signals to produce the desired manipulator performance. Up to this point, the monitoring of position has been discussed. The question of how one obtains velocity information is the topic of this section. As we will learn, it is possible to determine the angular velocity of a rotating shaft in several different ways. For example, the dc tachometer has been used extensively for this purpose in many different control applications, including robotics. In addition to this analog device, however, it is possible to utilize an optical encoder and a frequency-to-voltage converter to obtain analog velocity. Alternatively, the optical encoder itself can be made to yield digital velocity information when combined with the appropriate software. We now discuss, in turn, these various techniques for measuring velocity.

3.3.1 DC Tachometers

It is well known that rotating the shaft of a dc motor will produce an analog voltage that increases (or decreases) with increasing (or decreasing) shaft angular velocity. In effect, the motor becomes a dc generator and can therefore be utilized to measure the shaft speed. Although it is possible to use almost any dc motor in this application, dc tachometers are usually specially designed devices. There are a number of reasons why this is so.

The first and perhaps the most important one is that the tachometer ("tach") should produce a dc voltage that not only is proportional to the shaft speed but also has a voltage versus speed characteristic that is ideally linear over the entire operating range. (Some deviation from linearity is usually acceptable at speeds below 100 rpm, however; see Figure 3.3.1.) This permits the tach to be most easily used as a velocity sensor in control applications. Normally, the generated voltage produced by a dc motor will not possess the degree of linearity required in these cases.

A second reason for not using a motor in such an application is that the tach's output voltage should be relatively free of voltage ripple in the operating (i.e., speed) range of the device. Although a certain amount of ripple is permissible and can usually be handled with a low-pass filter, too much may produce unwanted jitter in the device being controlled. This would be particularly offensive in the case of a robotic manipulator. In general, a dc motor will produce too large a ripple for most control applications, so a specially designed device is preferable.

The final reason for not using a dc motor as a tach is that volume and/or weight is often an important system design consideration. As we mentioned before, this is certainly the case for the axes of an industrial robot, where the actuator must often be carried along in the joint itself. Since the tachometer supplies little if any current to the rest of the servo system, the output power requirement of the device is minimal. Thus it hardly makes sense to use a motor in this application, and a smaller device is quite satisfactory.

It is found that a permanent-magnet iron-copper armature tachometer will satisfy the above-mentioned characteristics. The speed-voltage curve of this analog device is quite similar to that shown in Figure 3.3.1. The underlying principle of the tachometer can be understood by recalling that of the tachometer can be understood by recalling that a wire moving in a magnetic field will induce a voltage across the wire that is proportional to its velocity and the sine of the angle between the magnetic field direction and the coil's plane. This angle is 90° when the wire's plane and the field are perpendicular to each other and results in the maximum voltage being developed.

In practice, the armature's copper (or aluminum) coils are wound longitudinally on a cylindrical piece of iron as shown in Figure 3.2.2. It can be seen that the ends of the coil are connected to a commutator, which is a segmented ring. Here only one coil is detailed, but normally there will be many spaced equally around the circular cross section. The corresponding commutator will then have twice as many segments as coils. The sliding electrical contact is usually obtained by a set of two or four carbon brushes which touch the various segments of the commutator.

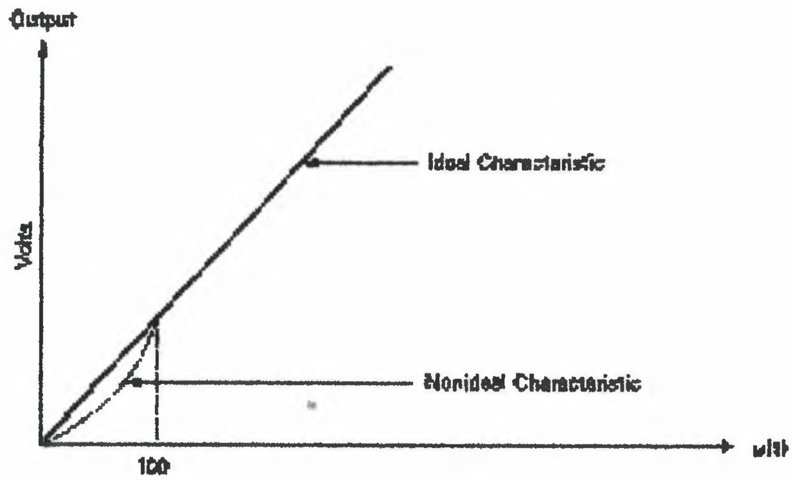


Figure 3.3.1. Output voltage versus speed transfer characteristic of an analog tachometer. At low speeds, the actual device (nonideal characteristic) is not linear.

Based on the above, the operation of the "rotary iron" dc tach can be understood. As any single coil rotates in the field of the permanent magnet, the induced voltage varies sinusoidally with angle. Thus at constant velocity, the voltage will also be sinusoidal in time. The brush commutator assembly will act as a rectifying element by reversing the coil connection for each half of a complete revolution. In this manner, a pulsating dc voltage is produced. All other armature coils will also produce a sinusoidal voltage of differing phase with respect to the first one. Since the coils are evenly distributed around the armature's cross section, the net voltage output by the brushes is very nearly constant (i.e., dc). The small ac component of the voltage that is present is referred to as ripple. Tachometers currently being manufactured usually produce ripples of about 3 to 5% of the dc output.

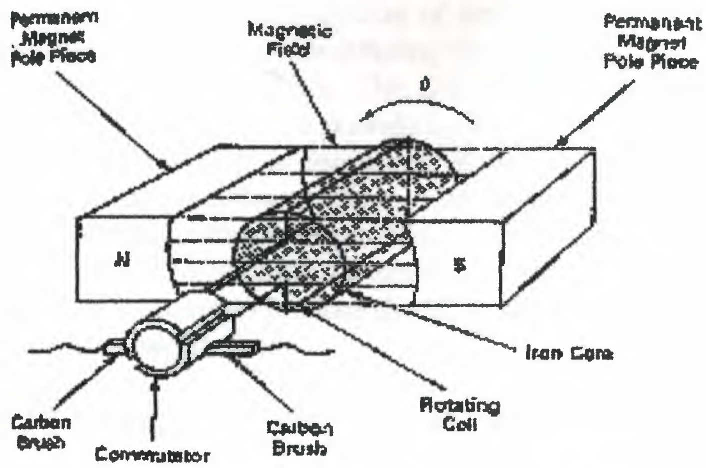


Figure 3.3.2. Analog tachometer showing one coil (of many) mounted on an iron core.

A more costly alternative to the rotary iron design described above is to use a moving coil for the armature. In this instance, a significant reduction in weight is achieved by employing a hollow "cup" whereby most, if not all, of the armature's iron is removed. This is accomplished by fabricating a rigid cylindrical shell out of the copper (or aluminum) coils or skeins using polymer resins and fiberglass. In addition , it is possible to utilize more coils. By eliminating the armature's iron, the inductance of this type of tachometer is reduced, thereby permitting the ripple voltage to be quite a bit smaller than for a rotary iron device. Typical values are in the order of 1% of the dc output. Also, because the moving-coil design allows more coils to be utilized, the low-speed performance of the tachometer is improved over that obtained by the rotary iron version.

It should be clear that if an analog tachometer is used in a robotic application, the moving-coil version is quite probably the more attractive of the two designs because of the reduction in weight. On axes where the actuator is not carried and hence weight is not a consideration, the rotary iron design may be preferable due to the reduced cost. Despite the fact that in this case, the increased ripple can be handled with a low-pass filter, its low-speed performance may still be objectionable, so that the moving-coil device may still be the unit of choice.

As of this writing, the most common class of industrial robot that makes use of an analog tachometer is the SCARA . The primary reason is that the configuration of such a robot does not require the actuator to be lifted against gravity. Recall that the major axes of a SCARA move perpendicular to the gravitational field, thus the added weight of the tach does not present a significant additional burden (i.e., torque load) to either the servomotor or the mechanical structure of the manipulator. However, where the motor must be moved against gravity, it is usually preferable to employ a different technique for obtaining the velocity signals. We now discuss two such methods.

3.3.2 Velocity Measurement Using an Optical Encoder

As mentioned above, the added weight penalty that must be incurred when using a permanent-magnet tachometer is often unacceptable in robotic applications where the actuator must be moved with the particular manipulator link against gravity. In this instance, an alternative to the extra piece of hardware is required. Fortunately, the optical encoder described in an earlier section of this chapter, and already used for position determination, is available for monitoring shaft velocity.

Two techniques exist for doing this. The first utilizes both the encoder and a frequency to voltage converter (FVC) to provide an analog voltage that is proportional to shaft speed. As far as the user is concerned, it behaves very much like the dc tachometer described in the preceding section. The second technique makes use of the encoder and appropriate software to provide a digital representation of the shaft velocity; pure digital servos , would utilize this approach. In fact, most robots today do indeed use the optical encoder to produce digital position and velocity information. We briefly describe these two methods.

3.3.2.1 Encoder and frequency-to-voltage converter

An earlier section of this chapter showed how the TTL pulses produced by an optical incremental encoder could be used to monitor position. The question arises: How can these signals be processed so that velocity information is also obtained the answer is found in the basic definition of velocity; that is, the time rate of change of position. Thus if the number of encoder pulses is observed (and counted) periodically and this number is converted to a dc level, the signal so produced will in fact be proportional to the shaft velocity. Clearly, we are approximating the derivative by Qx/Qt. Here, Qt is the "sampling" interval (or period) and Qx is the number of TTL pulses produced during this time interval.

A device that accomplishes the above is referred to as a frequency-to-voltage converter or FVC. This product of advanced integrated-circuit technology accepts both channels of the TTL encoder pulses and, using its own internally generated clock, counts these pulses during each clock cycle. The binary count is then output to an internal DAC which produces the desired dc voltage that is proportional to the encoder disk speed and hence the motor shaft speed. An example of an FVC is the Analog Devices AD 451 shown in Figure 3.3.3 in block diagram form. This unit will produce a 0- to 5-V output for pulse repetition rates of dc to 10 kHz.
(The AD 453 will go to 100 kHz.)

How does the velocity signal produced by this device compare to that of an analog tachometer? First, the output of the FVC has less ripple than that of the tach, and in fact the nature of this ripple is totally different. The internal DAC produces a piecewise constant output which, depending on its conversion rate will have a period (i.e., an update rate) which is so small that it will cause the FVC's output to appear to be continuous in most applications. Thus, unlike the analog tach, no low-pass filter is needed when using the FVC. Second, the FVC will exhibit more time delay than the tach, the exact amount depending on the internal clock rate.

In high-performance systems, such as semiconductor wire bonders that require servos having case of the servos used to control robot joints, the extra phase lag created by the large bandwidths, this delay can create stability problems which must then be dealt with using additional and

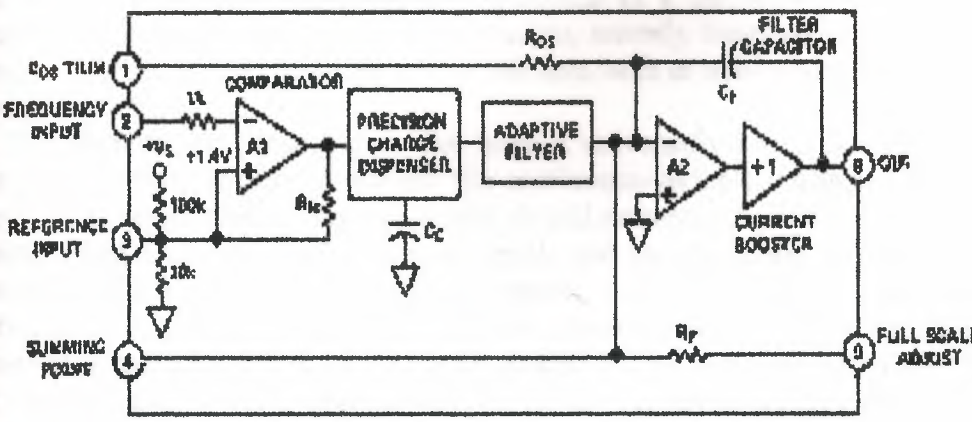


Figure 3.3.3. Block diagram of an Analog Devices 451/453 frequency-to-voltage converter. The 451 has a frequency range of dc to 10 kHz, whereas the model 4S3 can handle pulse repetition rates up to 100 kHz.

compensation. However, in the case of the servos used to control robot joints, the extra phase lag created by the delay is usually not of any consequence due to the much smaller bandwidth requirements.

Why is it that we do not find the FVC being used extensively in robotic systems? One reason is that current prices for the devices are on the order of $50, making the FVC almost as costly as an analog tachometer. As we will see, there is a considerably less expensive way to use the encoder signals to obtain velocity information. Another reason for not using the FVC is that hybrid servos, involving digital position and analog velocity, are not as frequently utilized. More common in robot systems are pure digital servos.

3.4 TOUCH AND SLIP SENSORS

Of all the senses that human beings possess, the one that is probably the most likely to be taken for granted is that of touch. It is only when a hand or arm is amputated that the ability to recognize objects and/or adaptively control the grasping force that comes from the human tactile sensory apparatus is truly appreciated. It is therefore not too surprising that in the attempt to imbue robots with some of the attributes of human beings, developments in robotic vision have outshadowed those in the area of touch and slip sensing.

In the last few years, however, as new and more sophisticated applications for robots have been conceived, tactile sensing has been recognized as an extremely important machine sense. In the area of parts handling, for example, it has become increasingly important to be able to detect any misalignment (i.e., the actual orientation) of the parts as they are presented to the robot. In addition, it is often necessary to know where a part is being grasped by the robotic gripper and whether or not it is slipping. Although vision has been used (or proposed to be used) in this respect, it appears that tactile sensing may be a less costly and faster (computationally) solution to the problem. Also, a major advantage of tactile sensing over vision is that it can yield the desired information about part position and orientation within the jaws of the gripper. Moreover, there are many applications where the limited resolution/pattern recognition capabilities of a tactile device is more than adequate for the desired task. For these reasons, recently there has been a significant increase in research and development in this area both at universities and in industry (robotic and otherwise).

The term tactile sensing does not have a universally accepted meaning. For example, Harmon has defined it to be "the continuous-variable sensing of forces in an array." The implication is that the sensor should possess skin-like properties and be capable of detecting differing levels of signals and parallel patterns of touching. In contrast to this, Harmon refers to simple contact or force sensing, whether binary or continuously variable at one, or at most a few, specific points as simple touch sensing. However, other researchers have chosen to describe as tactile sensors those devices that do produce a binary signal from the active elements in the array. That is, thresholding of the signal from an array element is used to determine whether contact has been made. It should be noted that just as in the case of binary machine vision it is possible to obtain a good deal of object recognition information in this instance. Also, because there are usually significantly fewer data, processing time is reduced significantly. In this section we use this modification of Harmon's definition.

In Harmon's report, he presented the results of a survey of 47 researchers and manufacturers in which they were asked to give the desired attributes of an "ideal" tactile sensor. In summary, he found that what these people wanted was that:

1. Tactile sensors should be compliant and rugged (i.e., durable in the manufacturing environment).
2. Sensors (i.e., sensory arrays) should be "smart," meaning that they should process most of the information before communicating with the robot.
3. Sensor resolution should be on the order of 100 mils, although some applications could require a larger or smaller value.
4. The sensors should be able to detect forces as low as 5 to 10 gr, with a dynamic range of about 1000 to 1.
5. Sensor response should be stable (with time), monotonic (and preferably linear, although some nonlinearity would be acceptable), and most important, not exhibit hysteresis (i.e., be repeatable).

The prototype experimental and the (few) commercially available tactile sensors meet some but not all of these requirements. In fact, much research and development remains to be done before a device exists that will permit a robot to lift an egg without crushing it or a block of brass without dropping it when the objects are presented in random sequence.

In the first part of this section, we present a number of tactile sensing techniques that have been used as either simple touch or tactile sensors with varying degrees of success. The second portion of the section discusses several methods of slip sensing.

3.4.1 Tactile Sensors

A variety of techniques and materials have been used in an attempt to produce a tactile sensor that is sensitive, rugged, and reliable (i.e., meets the requirements listed above). As of this writing, none do this, although a few satisfy some of items on the list. We will briefly describe a number of devices that utilize different sensing principles. In particular, we discuss an extension of the simple contact rod proximity sensor to produce a three-dimensional tactile sensor. Other devices covered make use of photodetectors, air pressure, conductive elastomers, or polymers as their sensing elements. The section concludes with a description of several tactile arc welding seam trackers.

3.4.1.1 Proximity rod tactile sensors

As mentioned earlier in this chapter, certain simple proximity-sensing techniques can be extended to produce a robotic tactile sensor. An example of this is shown in Figure 3.4.1. where the single-contact rod proximity sensor has been replaced by an array of such sensors (i.e., 4 x 4 = 16). A possible mode of operation requires that the robot wrist on which the device is mounted be moved down toward and parallel to the table or other surface on which an object is resting (Figure 3.4.1.c). Descent continues until the base of the sensor is at a distance approximately equal to the length of the sensing rods above the tabletop (Figure 3.4.l.e). At this point, mechanical or electrical switches connected to each of the sensor rods are checked for closure (i.e., contact). In this manner, a two-dimensional or binary pattern of the object is obtained. Image processing techniques similar to those employed with binary vision systems can be used to provide object type, shape, and orientation information. An appropriate set of actions can then be performed by the robot, [e.g., reorientation of the gripper (if necessary) and closing of its jaws].

A major difficulty with this technique is that the robot must know exactly how far to descend toward the table surface. If it does not go far enough it is possible that not all of the sensing rods will come in contact with the object. If it goes too far, the table will appear as part of the object. One method of overcoming this problem is to replace the (binary) switches with elements that measure actual distance (i.e., provide gray-level information). With such a modification as the sensor moves toward the object, the rods are once again pushed back into the body of the device (Figure 3.4.ld). However, in this case, the robot stops its descent when all rods have moved a minimum (or threshold) distance, thereby indicating that the sensor s elements have come in contact with either the object or the tabletop (Figure 3.4.le). Measuring the distance moved by each of these rods (relative to their starting position) yields a three-dimensional image of the object being "scanned." Gray-level image processing techniques similar to those employed with vision systems can be used for this purpose.

This procedure has a problem also. Since the rods must be able to move quite freely, it is possible that false deflections may be obtained. Spring-loading of the rods is possible, but a better solution suggested by the authors is to vibrate the tabletop. The robot will then continue to move toward the object until all rods are vibrating. At this point, the robot is commanded to stop, the relative rod deflections measured, and the object recognition algorithms used to process these data.

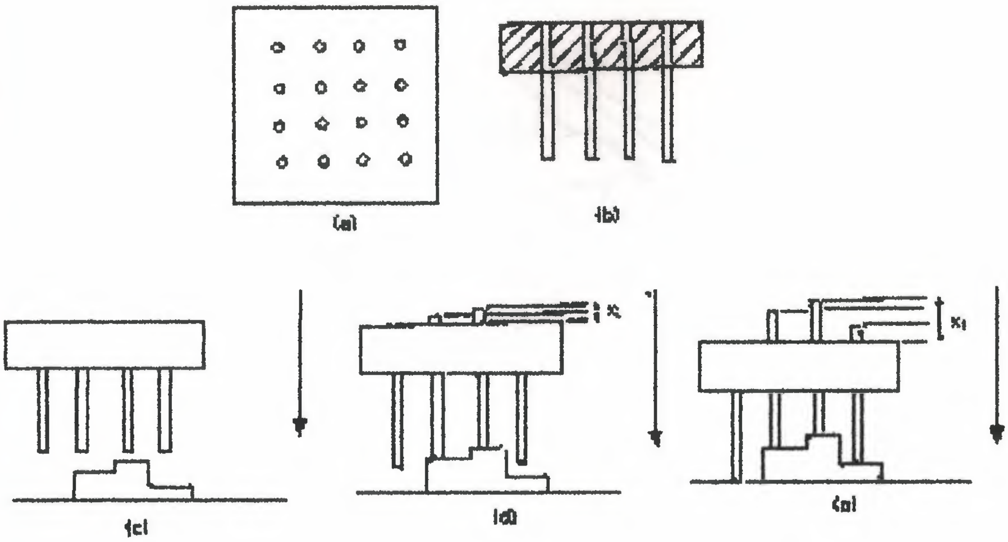


Figure 3.4.1. Three-dimensional proximity rod tactile sensor: (a) top view; (b) side view; (c) sensor descending; (d) sensor in partial contact with object(e) sensor in fullg contact with object.

Besides the originally proposed switch sensors, a variety of linear measuring techniques can be used to obtain the relative rod deflections. For example, the authors used rods made of ferrous material. Magnetic detection methods were then used to sense distance (see Figure 3.4.2). This was accomplished by causing the robot to move vertically (using stepper motors) and looking for a rod to move the ferrite cylinder into or out of the sensing coil. (Such an action produced a significant change in voltage across a coil.) The travel distance of each rod could be deduced from the instant each one caused a switch. The state of all the matrix of switches was continually scanned to determine the appropriate switching pattern and length of rod travel. In this manner, the part contour was sensed. In a later version of the tactile sensor, it was suggested that each rod be connected to a pot. All the comments of Section 3.2.1 concerning the problems of using a pot are pertinent in this respect. Obviously, many of the other position-sensing methods discussed in earlier sections of the chapter could also be used. However, the more expensive ones would not be practical since each rod would require a separate position sensor.

3.4.1.2 Pneumatic switch sensors

IBM has been interested in manufacturing advances for a number of years. As early as 1973, they announced the development of an experimental end effector upped with 100 pneumatic switches on each of the gripper's "fingers" (see Figure Figure 3.4.2).

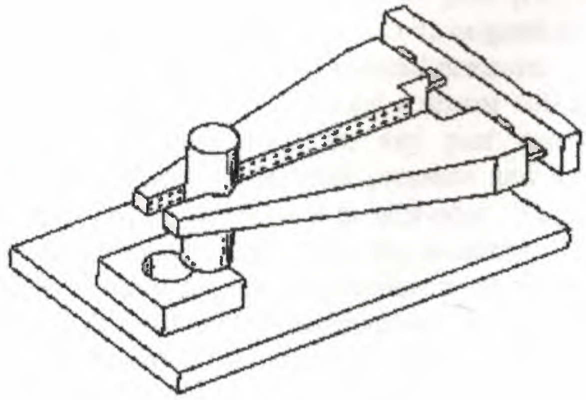


Figure 3.4.2. IBM pneumatic switch sensor. Each of the fingers of this experimental end effector has 100 pneumatic switches.

An exploded view of two of these switches is shown in Figure 3.4.3. The entire sensor is covered with a flexible "skin" whose purpose is to increase the friction between the walls of the gripper and the object being grasped (e.g., a peg). In addition, it is important that this skin (or covering) electrically insulate the part and the gripper. Some form of nonconducting flexible rubber or foam is suitable for the application. Underneath the covering, a thin metal sheet (or diaphragm) is secured to the sensor's solid body. With nothing inside the gripper's jaws, a source of pneumatic (or liquid) pressure keeps the diaphragm pressed against the skin. However, the gripping of the peg, for example, will cause a force (F1 or FZ) to be applied to a number of switch locations. If the force at any of the sites is large enough to overcome the applied pressure, the diaphragm will snap to the inward position as indicated by the dashed lines in Figure 3.3.2 .

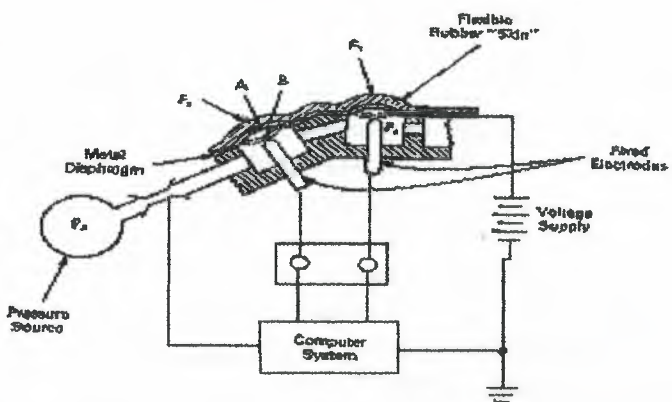


Figure 3.4.3. Exploded view of two elements of the pneumatic switch sensor shown in Fig. 3.4.2. The thin metal diaphragm is in position A if the force F2. is below some threshold whereas it is in position B if the threshold is exceeded.

This will, in turn, allow the diaphragm to come into contact with the fixed electrode , thereby completing the electrical path (i.e., closing the switch at that particular site). The metal sheet will return to its original position (switch open) when the external force is removed due to the internal pressure.

It is clear from this discussion that such a sensor will be strictly binary in nature. However, the actual force exerted on any part can be varied somewhat by preprogramming the amount of internal pressure and stopping the motion of the gripper's jaws when any single switch is activated. Sheet dimensions and material type also affects the force required to close the switches.

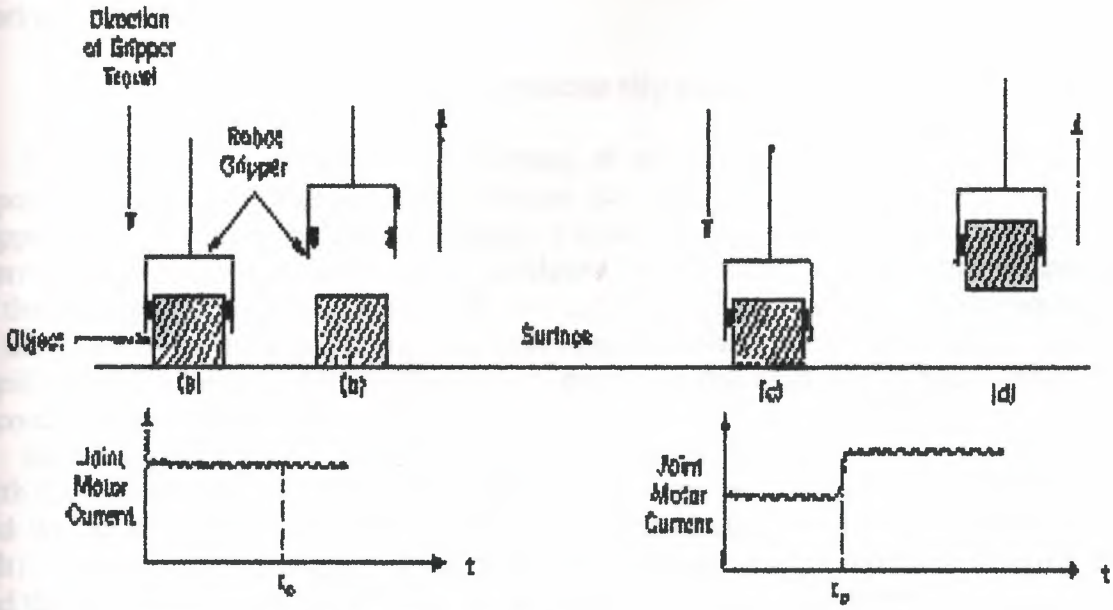
Although some binary pattern recognition is also possible with the sensor, the experiment that was actually performed involved the attepted insertion of a peg into a hole. The peg was placed within the gripper's jaws and moved to the approximate hole position. The air pressure was set high enough so that the act of grasping the peg caused no switches to be closed. The (binary) distribution of forces (switch closures) caused by any misalignment of the peg in the hole was sensed, processed by the computer, and this information used to modify the gripper's orientation so as to facilitate insertion. It is important to understand that since the manipulator has little or no compliance, any misalignment of the peg with the hole would normally prevent insertion.

Despite the promising nature of the pneumatic sensor, IBM did not continue its development. The most likely reason was its rather limited dynamic range (i.e., 0 to 50 gr).

3.4.2. Slip Sensors

One of the capabilities of the human hand certainly taken for granted is its ability to determine when an object that is being grasped is slipping. The biological control system associated with the hand utilizes the inputs from the appropriate slip receptors and causes the gripping force to be increased or decreased, as the case may be. Machine determination of slippage of a part or object when in the grasp of a robot or other electromechanical "hand" is still in the experimental stage. Of all the "external" robotic senses, slip detection is perhaps the least developed, and in fact, much of the research in the field has been oriented toward prosthetic applications. It is the firm belief of the authors, however, that this situation will change. Most certainly, in the next few years, as more complex and sophisticated assembly applications become commonplace, it will be necessary to detect slip rapidly and to adjust the gripping force "on the fly" to prevent the part from being damaged in a fall.

Perhaps the simplest way to determine if a part is slipping (or has not been properly grasped) is to use what is often termed the lift-and-try technique (see Figure 3.4.8). This entails using the motor current of a particular joint or set of joints on a robot as a measure of whether or not a part is slipping. In this respect, current monitoring can be performed either digitally or in an analog manner. Regardless of which technique is employed, the gripper is first oriented correctly, next placed over the particular part, and then a certain minimum grasping force applied. As the manipulator attempts to lift the object in question from the surface (e.g., a pallet, table, or conveyor) the motor current in one or more joints should increase due to the added load torque. If no increase is detected, the manipulator is commanded to return to the starting point. The force is then incremented by some predetermined amount and the robot "tries again." The procedure is repeated until the monitored joint current does



(a) Minimum gripper force applied ® t =to
(b) Part is not lifted since force is not large enough. Joint motor current does not change as gripper is raised above surface.
(c) Gripper descends again and force is incremented.
(d) Part is successfully lifted [force is high enough]. Joint motor current is seen to increase [® t=to indicating that gripper is loaded.

Figure 3.4.8. Lift and trv technique for slip detection.

applied. As the manipulator attempts to lift the object in question from the surface (e.g., a pallet, table, or conveyor) the motor current in one or more joints should increase due to the added load torque. If no increase is detected, the manipulator is commanded to return to the starting point. The force is then incremented by some predetermined amount and the robot "tries again." The procedure is repeated until the monitored joint current does increase, at which time it is assumed that the part is not slipping and is properly grasped.

There are obvious difficulties with this technique. The first is that even if the part is successfully raised above the resting surface, there is no guarantee that it will not slip out of the gripper as the manipulator moves. If, in fact this occurs, the procedure outlined above will not detect the slippage while the robot is in motion. A second problem is that if a fragile part is to be lifted, the minimum applied gripping force should be small, to avoid crushing. On the other hand, a heavy, more robust part could easily handle a larger initial force. If the mix and order of parts to be lifted were not known a priori, it is possible that either damage could be done to some or that it would take far too long to acquire others. The last difficulty with the technique is that monitoring motor current is not always error free. Care would have to be taken to prevent spikes due to brush noise from being mistaken as a current above the "lifting threshold." Obviously, the use of brushless motors would reduce this problem but would increase the cost, due to the need for electronic commutation.

In addition to the lift-and-try procedure, a number of experimental devices based on optical, magnetic, or conductive sensing techniques have been developed. We now describe briefly several of these slip detectors, which were proposed by groups working in Japan and Yugoslavia.

3.4.2.1. Forced oscillation slip sensors

In 1972 a group of researchers working at the Nagoya University in Japan reported on a number of experimental devices that were developed for detecting the slippage of an object being held by a robotic gripper. A variety of sensing techniques were employed. For example, "forced oscillation" was used whereby any translation of the part in a direction tangential to the surface of the gripper jaws (i.e., orthogonal to the direction of the applied gripping force) caused a short burst of voltage (i.e., a "spike") to be generated. One method is very much like that used to play back signals recorded on a (analog) phonograph record .

Here a sapphire needle protrudes from the surface of the sensor and is in contact with the object being grasped. If the part begins to slip, the needle will be displaced and will produce mechanical deformation in a piezoelectric crystal (e.g., Rochelle salt). The resultant generated voltage spike can be sensed using a threshold detector and the gripping force increased incrementally until the part stops slipping.

There are a number of difficulties with this type of slip sensor. The major one is that, although the rubber damper shown in the figure makes the device less sensitive to non-slip-related motions, the unit still tends to respond to mechanical vibrations of the robot manipulator itself. It is important to note that such a detector must be able to sense slippage accurately while a part is being moved. Thus it is necessary to reduce as much as possible this sensitivity to non-slip-related. Thus it is necessary to reduce as much as possible this sensitivity to non-slip-related motions. Another obvious disadvantage of the sensor described above is that the needle must move against the surface of a part in order to detect slip. This will cause eventual wear, meaning that periodic replacement of the needle will be required (just as in the case of the phonograph stylus). Finally, the entire device tends to be easily damaged if it is dropped or rapidly decelerated (also like a phonograph pickup cartridge).

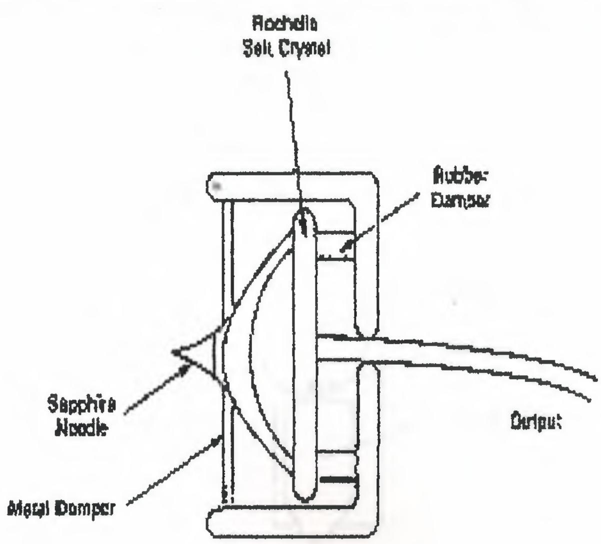


Figure 3.4.9. Prototype slip detector using a sapphire needle and piezoelectric crystal to produce a forced oscillation.

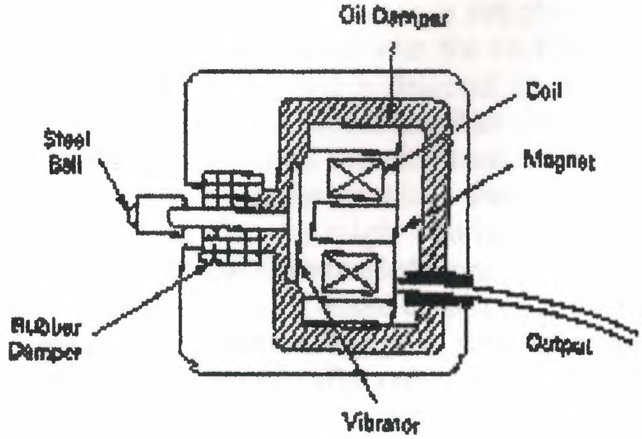


Figure 3.4.10. Improved forced oscillation slip sensor.

The unit shown in Figure 3.4.10 incorporates a number of improvements, which eliminate most of the difficulties described above. In particular, the sapphire needle has been replaced by a more robust steel ball (0.5 mm in diameter). The fragile crystal transducer has also been eliminated and in its place a permanent magnet-coil arrangement is used. Any motion of a part along the gripper's jaws will cause a mechanical displacement of the coil, which will, in turn, produce a voltage output. Again, the use of dampers (both rubber and oil) and thresholding circuitry will reduce the sensitivity of the sensor to manipulator vibrations.

3.4.2.2 Interrupter-type slip sensors

The same Japanese research group also developed additional prototype slip sensors using two techniques borrowed from other types of sensors [e.g., optical or magnetic) encoders or interrupters. One such detector is shown in Figure 3.4.11. This device consists of a rubber roller that protrudes above the sensor surface and contains a small permanent magnet. Any slip of the part in the gripper s jaws will cause roller rotation. As the permanent magnet passes over the magnetic sense head (e.g., one used in a tape recorder can be employed a pulse is generated and the gripping force can be increased accordingly.

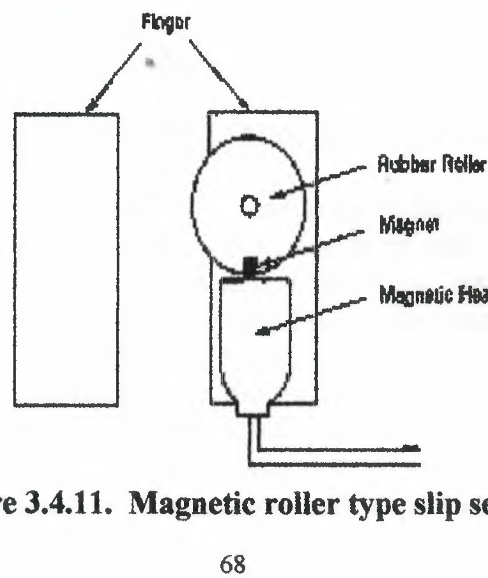


Figure 3.4.11. Magnetic roller type slip sensor.

A major disadvantage of this technique is that the slip "resolution" is not good. It can be seen from Figure 3.4.11 that if the sensor is not placed initially in a "reset" position (e.g., magnetic section of the roller over the read head), it will take some fraction of an entire rotation before any slip is detected. Even if the reset position is used to begin with, so that slip will then be detected as soon as almost any rotation occurs, further slip will not be sensed until one additional roller revolution occurs.

A possible solution to the resolution problem would be to use a number of magnets rather than one. These elements might also be replaced by small pieces of ferromagnetic material symmetrically embedded around the circumference of the rubber roller. In addition, the "read" head would consist of a simple dc energized coil. In this case, part slippage would cause a change in reluctance (and a corresponding voltage pulse at the terminals of the coil) after only a small fraction of a full rotation. Continued slip would also be sensed in this manner without having to wait for a complete roller revolution.

A second interrupter-like device uses optical rather than magnetic detection (see Figure 3.4.12). In this instance the rubber roller of the previous sensor has a small slit placed in it. A photoemitter-detector assembly is employed to detect any part motion. Clearly the device has the

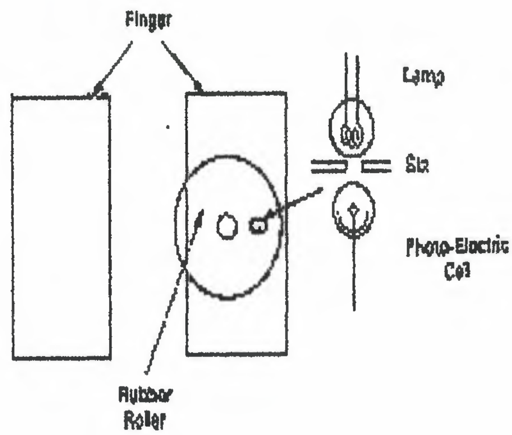


Figure 3.4.12 Optical interrupter type slip sensor. The photoemitter-detecto assembly consisting of the lamp, slit, and photoelectric cell is shown in detail at the right.

same resolution difficulty as described above. Again, an obvious way to increase the resolution would be to place more slits in the roller.

3.5. FORCE AND TORQUE SENSORS

In many manufacturing applications involving industrial robots, it is extremely important to be able to adjust and/or monitor the force and torque being applied to a part. A number of the tactile sensors described in the preceding section have the ability to provide information about how much force the jaws of a robotic gripper are exerting on an object. For example, the one produced by the Lord Corporation can be used as a force sensor in addition to its limited pattern-recognition capabilities. The reader is referred to Section 3.4 for further information.

There are, however, applications where only simple force/torque detection is required and where the inclusion of object recognition provided by an array of sensors may actually slow down the particular manufacturing process. In this section, we describe a number of techniques/devices that have been developed for monitoring these quantities direction (usually parallel to the longitudinal axis of the LVDT) limits its usefulness in robotic applications, where one cannot guarantee that forces will always be orthogonal to the fingers of a gripper. Thus it is not surprising that these transducers have not been generally utilized on robots. Clearly, some other type of device is required.

3.5.1. Strain Gage Force Sensors

One of the simplest methods of sensing a force (or pressure) exerted on an object is to detect the deflection of the fingers of the robotic gripper in response to such an applied force. The strain gage provides a convenient and accurate means of doing this. We now describe briefly the operation of such a device. Readers interested in learning more are referred to an article by Mounteer and Perrin.

The principle underlying the operation of a strain gage is that a mechanical deformation produces a change in resistance of the gage, which can then be related to the applied force. To better understand the operation of this device, consider a simple strain gage consisting of a plastic body (or some other flexible, nonconducting base) whose top surface is coated with a thin layer of a conducting material (e.g., aluminum or copper. If the conductive coating is assumed to have a uniform cross sectional area A, then the resistance of the device is given by

$$Rg = L/\sigma A \qquad (3.5.1.)$$

where σ is the conductivity of the conducting material and L is the length of the gage. When glued to an object, any deformation will cause the gage to bend either concave up or down. Since the conducting material thickness is assumed to be small, this bending produces little change in A. however, the gage length is either reduced or lengthened. It is seen from Eq. (3.5.1) that such action causes a corresponding decrease or increase in the gage resistance.
There are four basic types of strain gages:

1. Unbonded wire
2. Bonded metal foil
3. Thin film
4. Semiconductor

The first does not utilize a nonconducting substrate. Here the sensing element usually consists only of an extremely fine platinum tungsten wire. In the second class, a thin copper or aluminum alloy foil is glued to either a nonconducting base (bonded) or directly to the object under study (nonbonded). With the thin-film gages, vacuum or sputter techniques are used to deposit resistors onto heat-treated steel substrates. As such, these devices represent the "state of the art" in strain sensors. Probably the most sensitive of the four types, however, is the semiconductor gage, which is actually divided into two subclasses. One is made of silicon elements bonded to a cantilever beam or diaphragm and is called a "bonded bar semiconductor strain gage." The other borrows photolithographic and diffusion techniques from integrated-circuit technology to produce a "diffused-type semiconductor strain gage. '

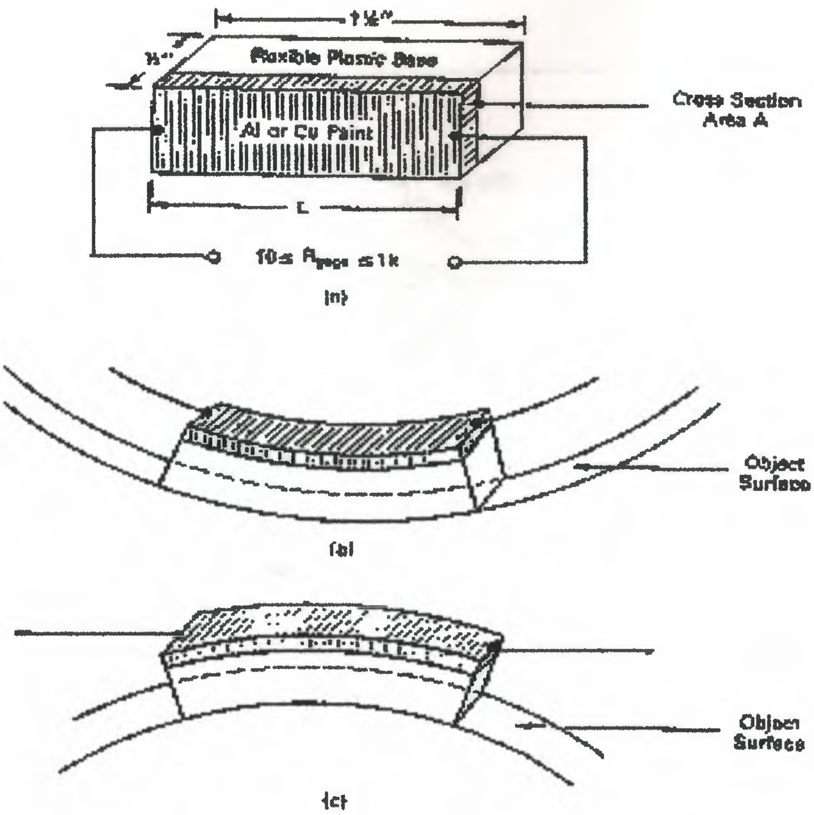


Figure 3.5.1. A simple strain gage: (a) undeformed device-gage (length = 1.5 in.; (b) concave up deflection of an object-L < 1.5 in. and Rg. decreases; (c) concave down deflection of an object-L > 1.5 in. and Rg increases.

Regardless of what type is employed, some electronic circuit must be used to sense the change in gage resistance and produce a voltage output as a result. Such a circuit is shown in Figure 3.5.2. Here the gage is one arm of a Wheatstone bridge (Rg). Another arm is adjusted so that its resistance R1 is approximately equal to that of the unstressed gage. A 100 kΩ fixed resistor R2 and a 1 MΩ balance pot Rb complete the circuit. If the bridge is initially balanced, so that

$$RgRb = R1R2 \qquad (3.5.1.)$$

the output voltage Vab will be zero. Any change in Rg caused by a deformation of the gage will unbalance the bridge and cause Vab to have a nonzero value. By applying a known force to the object onto which this sensing device has been placed, the system can be calibrated (i.e., the value of Vab can be related directly to force).

An obvious difficulty with this type of force sensor is that variations in ambient temperature tend to change the gage resistance, thereby causing the bridge to become unbalanced even when no force is applied. It is possible to overcome such a problem by either (automatically) rebalancing the bridge periodically or by utilizing two gages (and two bridges) and using the difference of their outputs as the actual sensing signal.

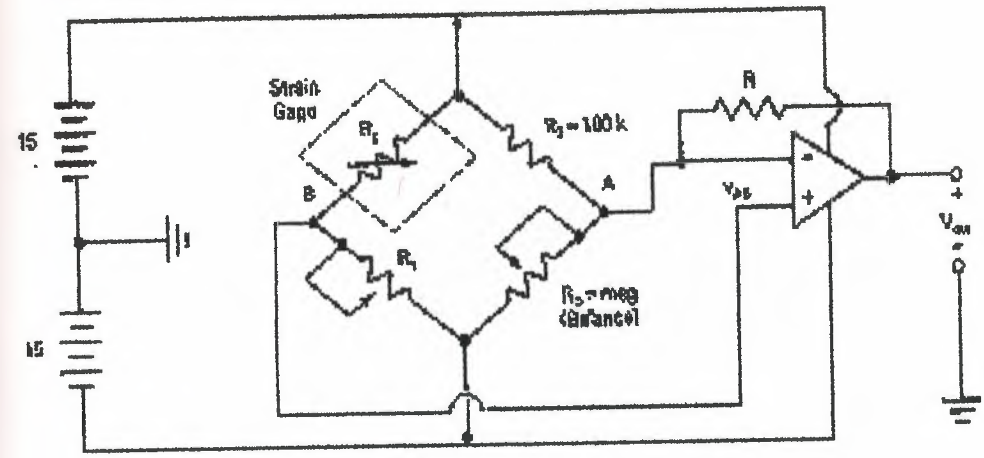


Figure 3.5.2. Strain gage hridge circuit. R, is approximately equal to Rg when the gage is not deformed. Rb is used to halance the bridge before any deformation occurs. Vab or Vout , can be related to the applied force.

The latter technique requires more circuitry but makes temperature drift a second-order effect.

Strain gages can be used to produce a robotic force sensing element. For example, as shown in Figure 3.5.3, these devices can be placed on the back of the "fingers" of a parallel jaws gripper. Then, as the fingers begin to grasp an object, the resultant deflection will be monitored by the gages. If the deflection versus force characteristic for the gripper material and structure is known, the grasping force can be related to the output voltage. For this technique to be successful, the objects being grasped must be made of a solid material that is significantly less deformable than the gripper's fingers. Obviously, there will be problems with using such a sensor to control the output force while lifting a fragile object (e.g., an egg or one made from Styrofoam), and some other force-sensing device will be necessary.

Another robotic force sensor making use of strain gages was developed in the mid-1970s at Stanford University by V. Scheinman as part of his graduate work. A modified version was produced at SRI-NASA Ames for the Ames Anthropomorphic robot (see Figure 3.5.3). In reality, this device was a six-dimensional force and torque detector that made use of clever machining of a piece of aluminum tubing to create a series of elastic beams onto which were bonded pairs (for temperature compensation) of foil gages. The unit was 3.2-in. long and had a 3-in. O.D. and a 0.18-in. wall thickness.

As can be seen from the figure, there are eight narrow beams, with four oriented so that their long axes are in the z-direction (denoted by PX+, Py+, PX-,and Py-), and with the remaining four perpendicular to the z-direction (denoted by QX+, Qy+, QX-, and Qy-). The gage pairs are indicated as R1 and RZ and are oriented so that a vector from the center of the latter passes through the center of the former along the positive x, y, or z direction: for example, the gages on beams PX+ and PX- are perpendicular to the y direction. The neck at one end of any beam has the effect of "amplifying" he strain at the gage positions while transmitting negligible bending torque.

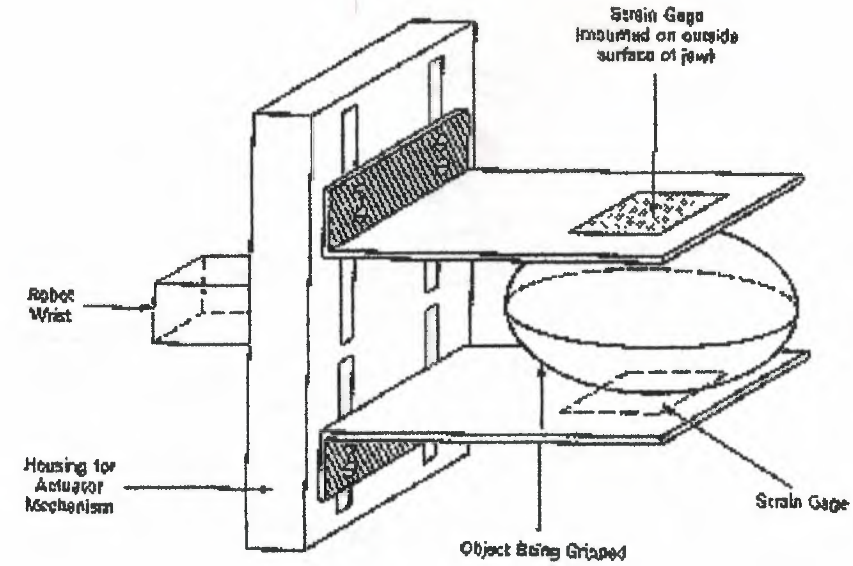


Figure 3.5.3. A robotic force sensor can be made by placing strain gages on the outer surfaces of the fingers of a parallel jaws gripper. These gages monitor the finger deflection when an object is grasped.

If the output voltage from any pair of gages is given the same name as the beam, e.g., Px+ is the voltage due to R1 and R2 located on beam Px+, it can be shown that the three forces FX, Fy, and Fz and the three torques MX, My, and Mz are proportional to

Fx ~ Px+ + Py-
Fy ~ Px+ + Px-
Fz ~ Qx+ + Qx- + Qy+ + Qy-
Mx ~Qy+ - Qy-
My ~ Qx- Qx+
Mz ~ Px + - Px - - Py+ + + Py -

Rather than use individual bridges for each gage, the potentiometric arrangement was utilized. Although not as sensitive as a Wheatstone bridge, this voltage-divider circuit obviously requires far fewer components to produce the desired temperature compensation effect. It should be noted that since aluminum is a good thermal conductor, any variation in the ambient temperature would cause the two gage resistances to change by the same amount. Thus the circuit output will not change.

To prevent failure due to excessive loading, the shear pins were included. These permitted the wrist to handle safely maximum x, y, and z forces of 70, 70, and 108 lb, respectively. The maximum safe x, y, and z torque loads were 72, 72, and 144 in.-lb, respectively.

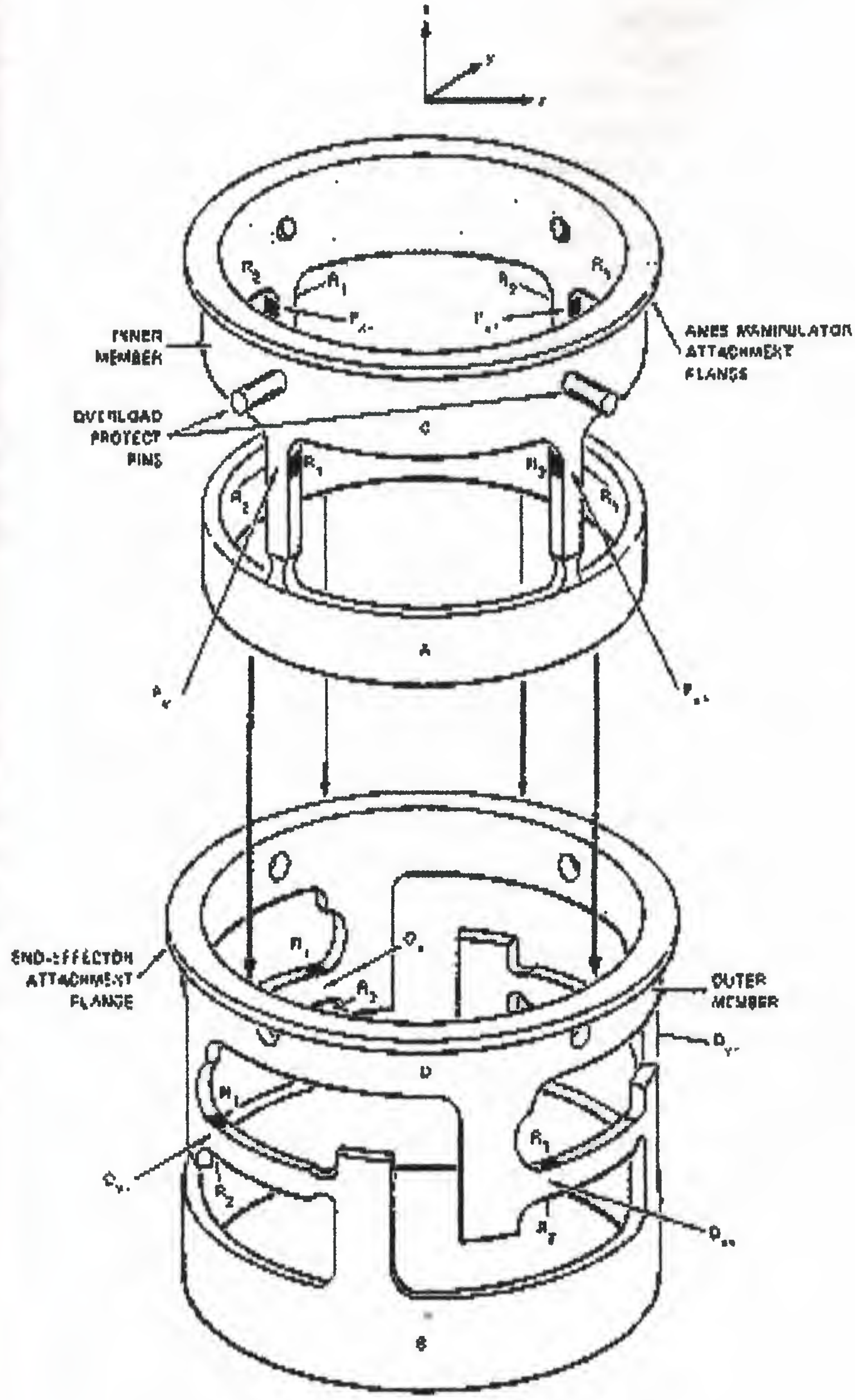


Figure 3.5.4. The SRI-NASA Ames force-torque sensor , a modification of the work of V. Scheinman , was developed for the Ames Anthropomorphic robot.

A refined version of the foregoing sensor was subsequently built for the Jet Propulsion Laboratory under a contract with Scheinman's company, Vicarm, Inc. (later acquired by Unimation). Semiconductor strain gages, mounted on each of the four sides of the deflection bars of the cross, were used in place of the foil types to provide increased sensitivity. Once again, a potentiometric circuit was employed and produced the eight outputs W1 to Ws, as indicated A 6 x 8 transformation matrix was then used to convert these data into actual x, y, and z components of force and torque as indicated in this figure. The information derived from this sensor permitted the adjustment of the grasping force applied to several test objects. Bejczy reported that the results were "highly repeatable."

A commercial robotic force sensor that utilizes strain gages is currently available (B and B Machine and Engineering, Allston, Massachusetts). This device places the gages on the fingers of a gripper and can yield x, y, z force information. (Three twising moments can also be obtained from the force data and gripper opening.) Its sensitivity is 0.5 V/Ib; it has a linear range of ±20 Ib with overloads of up to ±40 lb permissible, and weighs about 2 oz.

3.6 SUMMARY

In this chapter we have treated extensively the topic of non-vision-based robotic sensors. These have been divided into two classes, those that provide internal information and those that provide external information. The former group of devices is generally used to keep track of the individual joint's instantaneous position, velocity, and/or acceleration. With the data from these sensors, the joints can then be controlled properly. Of all the sensors considered, the optical incremental encoder has been presented in great detail and many of the practical considerations necessary for its successful application to robots discussed.

The second group of sensors introduced in the chapter provides the robot with the information about its (external) environment. As discussed, most of these devices are still quite experimental in nature, with only a few commercial units available. In the future, it will be absolutely essential that robots performing, complex manufacturing tasks possess the ability to apply just the right amount of force/pressure to an object. In addition, it will be important that these manipulators be able to determine what the object is from a tactile "image" provided by ; an array of sensors located in the gripper. One group of external sensors that are = well developed are those used on welding robots. These units are currently often utilized to assist the manipulators in producing welds that are both accurately placed and of high quantity.

CHAPTER 4

CONTROL

In this chapter, we will discuss the basic principles of feedback control theory. We will limit the discussion to the control of one joint. Extensions to the coordinated control of several joints. Initially, the discussion will focus on the continuous form of control, and differential equations will be formulated that describe the response in time. Then, software control and discrete time will be introduced.

We will assume that we are controlling a rotary joint, since these are very common in robots. The same principles are directly applicable to linear (prismatic) joints, such as pistons.

4.1 PROPORTIONAL ERROR CONTROL

The basic principle of control is very straightforward; move the system in the direction that minimizes some error function. An example error function might be E = θd - θ, where θd is the desired angular position and θ is the actual angular position of the joint. (We will in future sections refer to θd as the set point.) When E = 0, the joint is at the desired position. If E is negative, then the joint has moved too far and must reverse its motion. Thus, always moving in the direction which makes E approach zero will provide a type of control.

In addition to the drive direction, we should also be concerned with its magnitude. That is, not only must we ask, "In which way should I move the motor?" but also, "How much power (torque) should

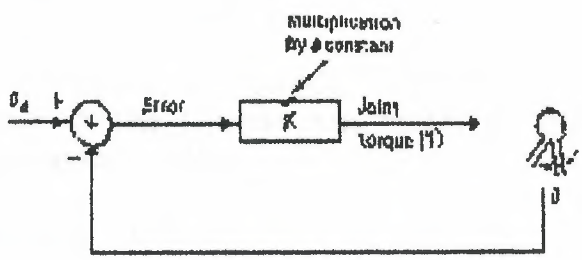


Figure 4.1 A PE control system, in which the control torque, T, is in proportion to the difference between the desired joint angle, Bd, and its actual angle (B).

I apply to the motor?" Again, the error signal E = θd - θ provides an answer. Let us apply a drive signal (a control) which is proportional to E. This rule defines a feedback control system as shown in Figure 4.1. Such a system is called a PE (proportional error) control system. To analyze the performance of PE controllers more carefully, we must have a model of the load being moved. In Chapter 3, we discussed DC motors and determined that

$$Tm = Jeq\ \theta'' + Feq\theta' \qquad (4.1)$$

where Tm, is motor torque, Jeq = J1 + Jm is the inertia of the load (reflected through the gears) plus the inertia of the motor, Feq is friction, defined similarly, and θ is angular position. Henceforth, to simplify notation, we will use J and F to represent Jeq and Feq , respectively.

We can also determine that, to a reasonable approximation,

$$Tm = KmI \qquad (4.2)$$

where I is the motor current. Equating Eqs. 4.1 and 4.2, we find

$$KmI=J\theta'' + F\theta'$$

Now our choice of a control law comes into play. First, let us assume PE control; that is, we apply a torque proportional to the error signal. This is accomplished by applying a current proportional to the error. This assumes, of course, that we have a controllable current source such as the amplifier. We will absorb the proportionality constants together to get

$$Ke(\theta d - \theta) = J\theta'' + F\theta' \qquad (4.3)$$

Without loss of generality, let $\theta d = 0$. (The choice of an origin is somewhat arbitrary.)

Now

$$-Ke\,\theta=J\theta''+F\,\theta' \qquad (4.4)$$

To see the performance of such a controller, first, suppose that it operates on a frictionless load (F = 0).

$$-ke\theta = JB'' \qquad (4.5)$$

This differential equation has as its solution a function that is equal to (minus) its own second derivative. A sine function does nicely, and we predict that a PE controller with no friction will oscillate.

Example 4.1 Oscillation of a Controller

Equation 4.5 can be rewritten as $\theta'' = -\frac{Ke}{J}\theta$. Assume that= 100.

Determine the solution to Eq. 4.5, and, from that solution, determine the frequency of oscillation.

Solution: Equation 4.5 describes a function proportional to its own second derivative. A sine is one such function, as is a cosine. Without information about initial conditions, we cannot determine which, or what the phase will be. Thus, we choose

$$\theta = \sin 10t.$$

We can verify that this satisfies Eq. 4.5 by differentiation.

$$\theta' = 10 \cos 10t$$
$$\theta'' = -100 \sin 10\,t = -100\theta$$

The motor so controlled will oscillate at a 10-Hz rate.

In the presence of friction, Eq. 4.4 describes the behavior and has a solution of

$\theta = \exp(-F/2J\ t)\ [C1 \exp(2\ wt) + CZ \exp(-\ z\ wt)]$

where

$$w = \sqrt{(F2/J2)-(4Ke/J)} \qquad (4.6)$$

The damping term, exp (-F/2J t) guarantees that, with increasing time, the joint will get closer and closer to its goal of B = 0. Further-more, we can see that if F2/4 Ke > J, then the term under the radical will be positive, resulting in even more damping. Such a solution is referred to as overdamped, and is demonstrated in Figure 4.2.

If F2/4 Ke >G J, then the exponent is complex and the solution is a damped sinusoid, shown in Figure 4.3. Intuitively, these solutions make sense, for high friction means "hard to start, easy to stop"; and high inertia means "hard to start, hard to stop."

If F2/4Ke =J, the solution is critically damped. That is, it gets to the goal as quickly as possible without overshoot.

Clearly, by choosing Ke, we can affect the performance of the system. We will discuss this at a greater length later.

There are some problems with the proportional error control system when we must hold a load against gravity. To do so requires a torque, so we cannot hold against gravity without an error, since no error would imply any torque. This is known as the steady-state error problem.

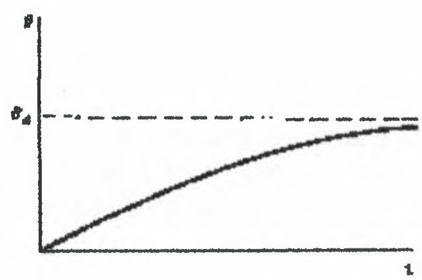


Figure 4.2 Overdamped.

A second problem with proportional error control is overshoot. That is, a manipulator operating under control of a proportional servo has only friction to slow it down. To see this, suppose that the arm is close to its destination. Then

$T = Ke(\theta d - \theta)$

is quite small, but not negative. If there is much friction, the arm may stop short of θd due to lack of drive, but if friction is small and inertia is large (relatively), then the arm may move on pastθ d. Now the error signal is negative, torque is backward, and the arm will be driven back to θd. In the meanwhile, however, it has overshot its goal.

If the task to be performed requires critical positioning, as in moving television picture tubes, for example, the occurrence of overshoot can be disastrous.

Let us now consider some techniques for dealing with steady-state error and overshoot.

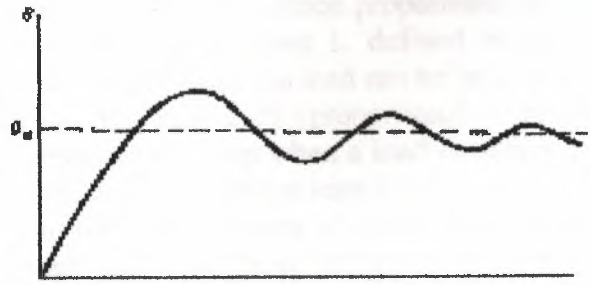


Figure 4.3 Underdamped.

4.2 THE STEADY-STATE ERROR PROBLEM

To hold a load against gravity, a controller must exert a torque (or force). To do so with a PE controller requires an error, as the following example illustrates.

Example 4.2 Effect of Loading on a PE Controller

A robot with one prismatic joint is oriented so that the joint is aligned vertically. The actuator is controlled by a controller that relates force (measured in kilograms) to position (measured in centimeters) by

$$f = Ke(Yd - Y)$$

where

$$Ke = 0.6 \text{ kg/cm}$$

Assume that the load seen by the actuator (including the mass of the joint itself) is 30 kg. If the desired vertical position is y = 300 cm, determine the true position in the steady state (that is, no acceleration and no velocity).

Solution:

$$Yd - y = \frac{f}{Ke} = \frac{30 \text{ kg}}{0.6 \text{kg/cm}} = 50 \text{ cm}$$

Since yd = 300 cm, y must be 250 cm.

Thus, to hold a load of 30 kg against gravity, this controller will be in error by 50 cm, in the steady state.

One approach to dealing with steady-state error is to produce as output a torque

$$T = L + Ke(\theta d - \theta)$$

where L is a constant sufficient to hold the load when $\theta d - \theta$ is zero.

Use of this approach requires that the load be known precisely. In the case of robots, this knowledge is difficult to achieve since the load on a particular joint is usually a function of the positions and motions of the other joints.

An alternative is to make the drive signal equal to the integral of the error with respect to time. That is, allow the output of the servo (the motor torque) to accumulate with time and make the rate of accumulation proportional to the error signal. Such a controller essentially finds the constant L defined earlier by the experimental technique of increasing L slowly until the load can be held stationary with no error.

A robot operation with such a PI (proportional integrating) controller on its vertical axis can be observed to droop when a load is suddenly applied and then rise back to the desired position as the integral term builds up.

Of course, the improved performance of an integrating controller does not come for free. We have been discussing how the integrator helps the steady-state error problem. That is, integration is of assistance when the arm is stationary or moving slowly. When an integrating controller is used to achieve fast motion, however, it tends to increase the overshoot. In fact, under certain loading conditions, such controllers can be unstable and oscillate about the desired point. Thus, while reducing one problem, steady-state error, we have made another problem, overshoot, even worse. Such are the joys of engineering (or economics, for that matter)!

4.3 THE OVERSHOOT PROBLEM

Overshoot occurs because the controller has an insufficient mechanism for "applying the brakes." A PE or PI controller in fact has nothing to stop the arm other than friction. If there is any positive error at all, $\theta d - \theta$ is positive, and the motor will have positive (although small) drive applied right up to the point where the error goes to zero. If friction is small and inertia large relative to the friction, a joint driven by such a controller will overshoot.

To provide a degree of active braking, we can use the following concept:

1. If error is large (we are a long way from the goal) and the velocity is small, apply a large drive.
2. If error is small (we are close to the goal) and the velocity is high, apply a negative drive.

The simplest way to achieve this is to make the drive torque proportional to the derivative (rate of change) of position with respect to time:

$$T = Ke(\theta d - \theta) - Ka\theta' \qquad (4.7a)$$

where θ' is the angular velocity.

This equation defines the operation of a PD (proportional derivative) controller.

The ability of this controller to handle overshoot then depends on the gains of the controller, Ke and Kd, and the inertia and friction of the load. One cannot always guarantee zero overshoot unless something is known about maximal values for inertia. Choice of optimal Ke and Kd is then possible. However, these constants are most often determined experimentally. Increasing Kd is equivalent (for purposes of control) to increasing the friction of the system.

To see this, we once again equate Eqs. 4.1 and 4.2 and substittite the new control law

$$Ke(\theta d-\theta)- Kd\theta'=J\theta''+F\theta' \quad (4.7b)$$

again assuming without loss of generality that $\theta d = 0$

$$-Ke\theta-Kd\theta=J\theta''+F\theta' \quad (4.8)$$

Rearranging terms yields

$$-Ke\theta =J\theta'' +(F+Kd)\theta' \quad (4.9)$$

which is exactly Eq. 4.4 with Kd added to the friction. Thus, a PD controller has exactly the same behavior as a PE controller, but the designer now has another parameter to adjust for best performance, a parameter functions exactly like friction.

A controller with derivative feedback can be combined with the concept of integration to yield a PID (proportional integral derivative) controller. There are several ways in which one could configure such a controller. One such configuration is shown in Figure 4.4.

The torque provided by a PID controller satisfies

$$Tm (t) = Ke(\theta d - \theta(t) +Ki \int_{to}^{t} (\theta d - \theta(r)dr - Kd\theta'(t) \quad (4.10)$$

As we will discuss in the next section, this equation could be written in discrete form as

$$Tm, (i) = Ke(\theta d - \theta(i) + \frac{K}{i} i \sum_{l=0}^{i} (\theta d - \theta(l)) - Kd\theta'(i)$$

at time t = i, which explicitly indicates the use of discrete time computations.

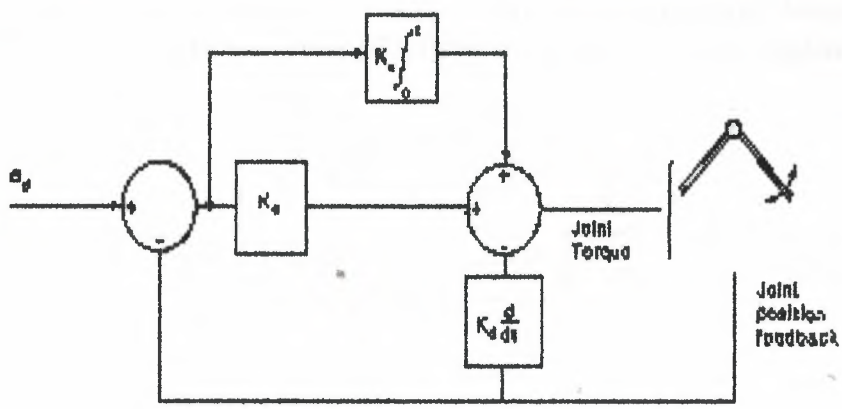


Figure 4.4 PID control.

Increasing Kd tends to slow the arm down since it increases the negative contribution to torque due to velocity. Decreasing Kd decreases the damping of the system, thus increasing the likelihood of overshoot.

Because of their relatively simple implementation and robustness, PID controllers are probably the most commonly used controllers today, even though their performance is not necessarily optimal.

4.4 THE SAMPLED-DATA CONTROLLER

The control system of Figure 4.4 could easily be implemented with operational amplifiers, as shown in Figure 4.5. Proper placement of capacitors provides differentiation and integration, and use of potentiometers provides control of the gains Ke, Kd, and K;. Such analog circuits were the standard means for realizing servo controllers until very recently. It should be noted that the differentiator using op-amps shown in Figure 4.5 tends to be extremely sensitive to noise. A superior practice is to sense 8 directly, with a tachometer, for example, rather than to differentiate 8. However , silicon technology now provides means for performing the same operations digitally and thus avoids the difficulties of accuracy, drift, and temperature compensation that plague analog circuits. Furthermore, the continuing decrease in the cost of digital circuitry makes the digital approach increasingly attractive.

Probably the most obvious way in which to implement a PID controller digitally is to replace each block of Figure 4.4 with its digital

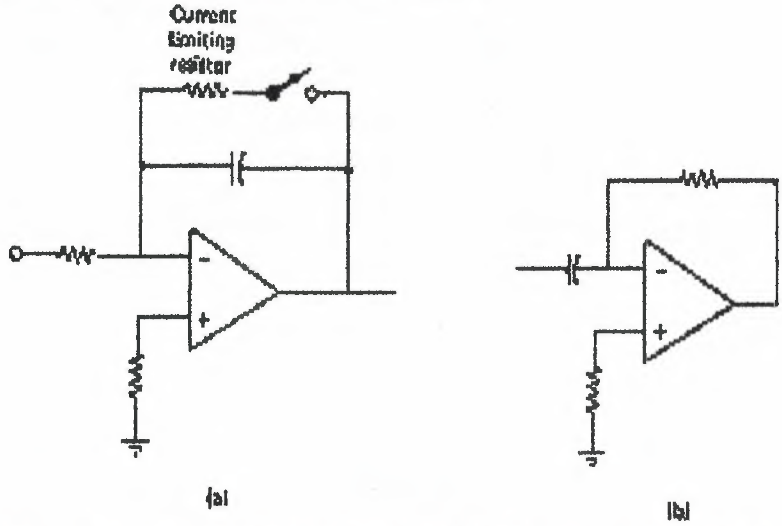


Figure 4.5 (a) Integrator using operational amplifier. The switch is used to set initial conditions to zero.(b) Differentiator using operation amplifier.

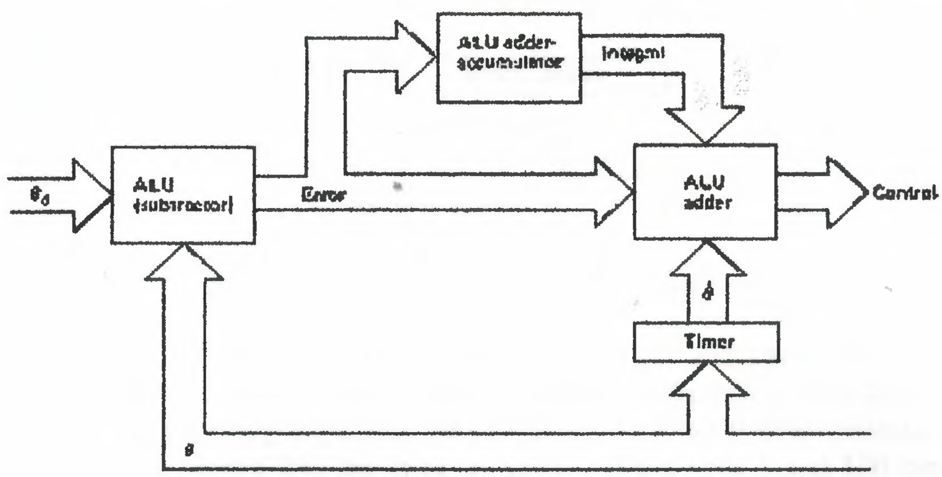


Figure 4.6 Digital implementation of PID control.

equivalent, as shown in Figure 4.6. There, an ALU (arithmetic-logic unit) is used to compute the error signal; an ALU plus accumulator provides the integration; and a counter, with timer, provides differentiation. We will refer to this as a parallel digital implementation.

Such an approach is far from cost effective when compared with performing same operations in a computer. The cost of microprocessors is now so low as to make a software approach far more economical, as well as providing the flexibility of easy modification. Although the parallel approach might be necessary if the variable to be controlled changes very rapidly (e.g., a new and significantly different value every 10 microseconds), the joint variables of a robot change much more slowly than this. Later in this section, we will examine the speed requirements for a computer controller.

Figure 4.7 shows a flow chart representing software implementation of a PE controller. The unique difference between a software controller such as this and a continuous time controller is the effect of loop cycle time. Loop cycle time refers to the amount of time required from when the input is read until the input is read again. The functioning of such a controller is analogous to riding a bicycle with one's eyes closed. The rider takes a quick look, determines that he still has far to ride, closes his eyes, and pedals hard. Sometime later, he looks again, determines that he is now closer, closes his eyes, and does not pedal quite as hard. Obviously, the performance of the system depends strongly on how often the rider takes a look, or on how often control returns to block 1 of the flow chart.

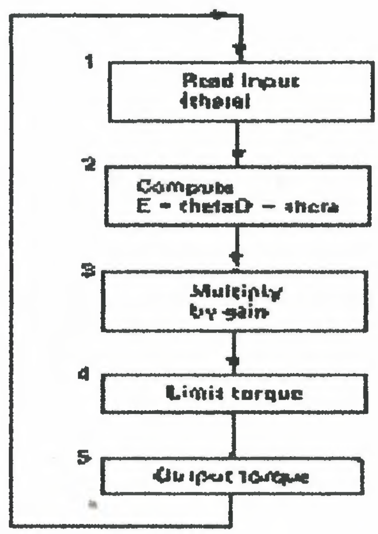


Figure 4.7 Flow chart of a PE controller.

The minimum loop cycle time (or sampling interval) will vary from one robot to another. It depends on the mechanical time constant of the physical system, which is related to the inertia. Larger systems have a longer mechanical time constant. Studies (Paul, 1982) have shown that the minimal value lies between 5 and 100 ms. Many robot systems use 16 ms (1 /60 sec) since the line provides such convenient source of timing signals.

4.5 THE VOLTAGE-CONTROLLED DC MOTOR

If a permanent magnet DC motor (pmDC) is driven by a controllable current source, such as the circuit shown in Figure 4.7, then we can use Eq. 4.3 to describe the system's performance. It is not always possible, however, to supply a controlled current. If we instead control the voltage to the motor, we must take a closer look to analyze the system.

Figure 4.8 represents a pm DC motor in the steady state, that is, turning at a constant velocity. In this model, we ignore the motor's inductance. This simplification is made since transient effects due to inductance are, in general, much faster than the mechanical actions we are controlling. Thus, we assume that the physical motor cannot respond to the inductive transients.

If we happen to be controlling voltage, with a PD controller, we have

$$Vd = Ke(\theta d - \theta) - Kd\,\theta' \qquad (4.11)$$

Then, the current through the motor is

$$I = \frac{Vd - Eb}{R} \qquad (4.12)$$

$$= \frac{1}{R}(Ke(\theta d - \theta) - Kd\,\theta' - Eb) \qquad (4.13)$$

Torque is related to current by

$$Tm = Km\,Ia \qquad (4.14)$$

$$= \frac{Km}{R}(Ke(\theta d - \theta) - Kd\theta' - Eb) \qquad (4.15)$$

The back EMF of the motor results from its acting as a generator and is proportional to rotational velocity:

$$Eb = Kb\theta' \qquad (4.16)$$

So Eq. 4.15 becomes

$$Tm = \frac{Km}{R}(Ke(\theta d - \theta) - Kd\,\theta' - Kb\theta'$$

$$= \frac{Km}{R}(Ke(\theta d - \theta) - (Kd + Kb)\,\theta') \qquad (4.17)$$

$$Tm = K1(\theta d - \theta) - K2\,\theta'$$

We see that this equation has exactly the same form as Eq. 4.7a and this will have the same type of solutions. However, the damping constant has been increased by Kb

. Hence, the effect of the back EMF of the motor on a voltage-driven controller is to increase the damping constant of the controller.

4.6 CHOOSING SERVO GAINS

As we saw in Eq. 4.10, the torque from a PID controller is described by

$$T = Ke\,(\theta d - \theta) + Ki \int(\theta d - \theta)\, dt - Kd\,\theta' \qquad (4.10)$$

As before, we could equate this torque to the mechanical torque, T = JB + F8, and develop a differential equation to characterize the performance. In this case, the differential equation would be third order. Nonetheless, conditions could be developed for best performance and formatted as equations involving the gains Ke, Ki, and Kd and the loads J and F. We will do just that and discover that since J and H□, J especially, change radically with arm con iguration, such simple optimization techniques are doomed to failure.

The θ-r Manipulator

To see the difficulties inherent in choice of gains, we will consider the θ -r manipulator. This simple robot, shown in Figure 4.8, has only two actuators, one driving a rotary joint and one driving a prismatic joint. If we are using PD control on the rotary joint, then the torque from the controller is

$$T = -Ke\,\theta - Kd\,\theta'$$

and the torque due to the motion of the load is

$$T = J\theta'' + F\theta'$$

Equating these, as earlier, we once again find Eq. 4.9, with time solution

$$\theta = \exp - \frac{Kd + F}{2J}\, t\, [C1 \exp(wt/2) + C2 \exp(-wt/2)] \qquad (4.18)$$

where

$$w = \sqrt{\,[(F+Kd)2/J2] - (4Ke/J)} \qquad (4.19)$$

The critically damped solution gives the most speed without overshoot can be formed by setting the term under radical equal to zero.

$$(F + Kd\,)2 - 4\,KeJ = 0 \qquad (4.20)$$

Any choice of Ke and Ka satisfying this condition will be critically damped.

Two observations are in order. First, the critically damped solution is in fact many solutions. Any of an infinite number of choices of Ke and Kd can satisfy Eq. 4.20 as is true any time we have two variables and one equation. To find a unique choice for Ke and Kd requires adding another constraint. One such constraint which is popular is to require that some function be minimized, such as

$$\min \int u2\, dt$$

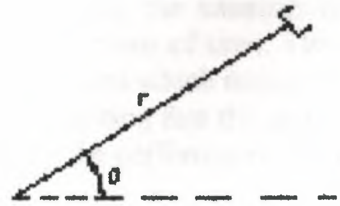


Figure 4.8 θ -r manipulator.

where u is the output of the controller.

$$u = Ke\,\theta - Kd\,\theta'$$

This is referred to as a minimum energy controller. Use of this condition can result in a unique choice of Ke and Kd. Details of how this is done are beyond the scope of this book. The reader is referred to basic texts in optimal control (Bryson and Ho,1969).

The second observation that needs to be made in this context is the dependence of the solution on J and F, as r varies.

The inertial load seen by a rotating actuator is

$$J=mr2 \qquad (4.21)$$

where m is the effective mass and r the effective radius. That is, m is the mass at the end of a massless rod of length r.

Substituting J = mr2 into Eq. 4.9, we find

$$mr2\,\theta'' + (F + Kd)\theta' + Ke\theta = 0 \qquad (4.22)$$

This equation still has the solution given by Eq. 4.18 and can be critically damped if

$$(F + Kd)2 - 4mKer2 = 0 \qquad (4.23)$$

If r is known and constant, we have no problem. However, now let us consider two more difficult (and more common) cases: constant gains but different r and coordinated motion.

Constant Gains

If we choose a Ke and Kd to satisfy Eq. 4.20, we have a solution that is good (critically damped) at only one point. We see from Eq.4.23 that if r takes on a different value, our previous solution will be in-correct. In fact, if r becomes smaller, the solution will be overdamped and if r becomes larger, it will be underdamped. (The proof is left as an exercise.) So with constant gains, the system is virtually always performing poorly. Of course, for any constant value of r, Eq. 4.23 can be used to find good values for the gains, and this technique is often used.

Coordinated Motion

If both joints move simultaneously, the situation is much more complex, for in that case, r and, therefore, J are functions of time. The system must be described by more sophisticated modeling techniques which incorporate the interaction of forces.

For now, let us conclude by suggesting that the gains be chosen experimentally by making adjustments and observing the performance. Two rules of thumb are

1. Increasing Kd leads to more sluggish, overdamped response.
2. Increasing Ki significantly increases the likelihood of overshoot and oscillations.

4.7 CHOOSING THE CONTROLLED VARIABLE

Up to this point, we have discussed only control of angular position , θ. However, there is no reason that other parameters could not be controlled. In general, we must be concerned with two variables, the variable that is implicitly controlled and that which is explicitly controlled. For example, in a DC motor, we implicitly control torque, and in a hydraulic system, we implicitly control velocity. The variable explicitly controlled is the variable that creates the error signal. For example, we may feed back true velocity, subtract it from desired velocity, and use that error signal to provide the drive signal, thus implementing a velocity servo. In this context, we will use the term controlled variable to mean that variable that is fed back and subtracted.

As system designers , we have the freedom to choose the controlled variable according to the application at hand. For example, we may need to avoid abrupt starts and stops in a very-high-gain hydraulic system and thus may program in an acceleration algorithm, as shown in Figure 4.9. One convenient means for implementing acceleration is to provide input to a velocity servo.

Although we have the freedom to choose the controlled variable

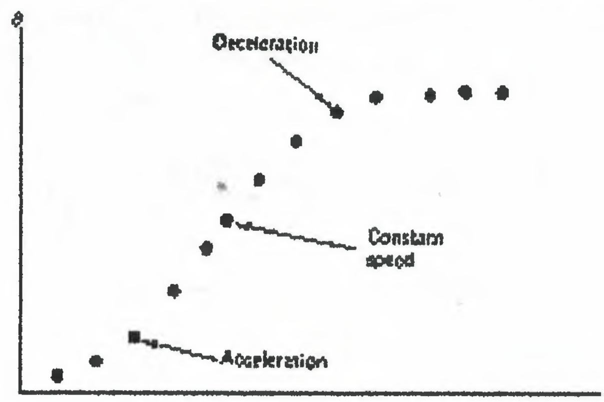


Figure 4.9 Set points for a position servo. By changing the set point as a function of time, acceleration may be controlled.

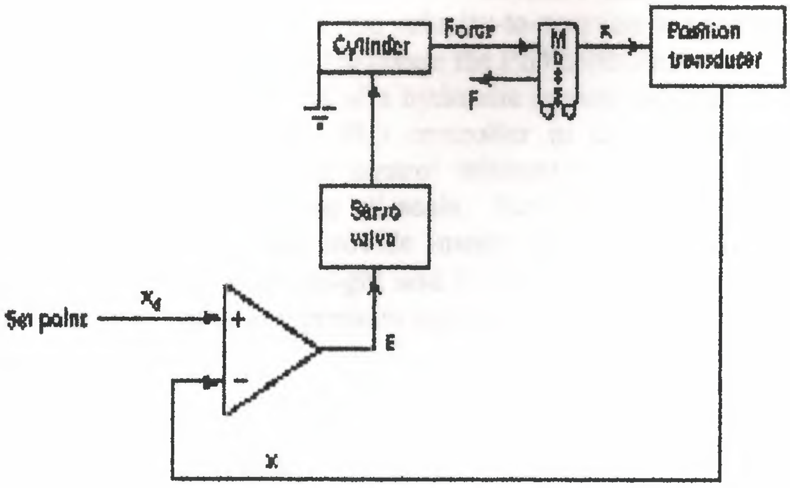


Figure 4.10 A hydraulic position control system.

the physical nature of the system may influence our choice of which variable to choose. For example, in a flow-controlled hydraulic system, the servo valve controls flow and, therefore, velocity. We can make use of this fact to implement simple acceleration algorithms. Furthermore, if velocity is not the variable we intend to control, the fact that it is the implicitly controlled variable may lead to unexpected results. To see what these may be, let us consider position control of a hydraulic system.

Figure 4.10 shows the structure of a position controller. Figure 4.11 is derived from Figure 4.10, neglecting dynamic characteristics of both the servo valve and the actuator. The model does include, however, the frictional (damping) effects within the actuator and load friction. From Figure 4.11, we see that the servo valve produces an output flow or, if we neglect damping (F = 0), a flow proportional to the error in position. Hence, we have a PE controller driving a system with damping. But unlike the PE controller we studied earlier, this error signal results in a velocity, not a torque.

In Figure 4.1 I, the output of the actuator is seen to be a velocity. The load is then modeled as a simple integrator that converts velocity into position. However , for sufficiently powerful actuators , this may be an

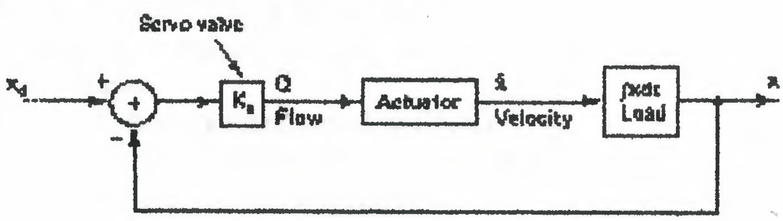


Figure4.11 Representation of Figure 4.10 ignoring dynamics.

acceptable model. (See Clark , 1969, for a serious introduction to design of electrohydraulic control systems.)

The fact that it is velocity that is ultimately controlled leads to a surprising result: This PE controller does not suffer from the steady- state error problem. To see this, recall that any nonzero error will leave the servo valve at least partially open, resulting in flow. That flow will continue, with resultant displacement of the actuator, until true zero error is reached. In fact, the velocity-to-position integration acts exactly like the integrator we explicitly added to create the PID controller.

Thus, we see that simple analysis of a hydraulic system indicates that PE control results in performance similar to a PID controller in an electrical system. The integration results from the velocity control inherent in the servo valve and the damping from the high friction in the oil seals. Similar nonquantitative analysis of other control systems can likewise provide insight into perforamce. The reader is referred to Clark (1969) for both insight and first-level quantitative analysis on not only position, but also velocity and pressure controllers.

4.8 Notation

Symbol	Meaning
θ	Angular position, joint variable for a typical rotary joint
θ'	Angular velocity
E	Position error
θ d	Desired position
Jeq	Inertia seen by the actuator (including its own inertia)
Feq	Friction seen by the actuator (including friction due to the actuator itself)
J	Same as Je a
F	Same as Feu
Tm	Torque of a DC motor
T	An arbitrary torque
I	Current into a DC motor
Km	Proportionality constant, relating current and torque
Ke	A servo gain (a constant multiplying the error term)
K	A servo gain (a constant multiplying the derivative term) .i ;
Ki	A servo gain (a constant multiplying the integral term)
va	Voltage applied to a DC motor
R	Resistance of the armature of a DC motor
Eb	Back EMF generated by a DC motor
Kb	Proportionality constant, relating back EMF to angular velocity
K1	A proportionality constant, equal to K", Ke /R
K2	A proportionality constant, equal to Km (Ka + Kb)/R
r	The second joint of the 8-r manipulator
θ (alter.)	The first joint of the 8-r manipulator
u	The output of an arbitrary controller
m	A mass
Q	Flow, quantity controlled by a hydraulic servo valve

CHAPTER 5

COMPUTER HARDWARE AND SOFTWARE FOR ROBOTS

This chapter discusses the computers and computer components used to operate and control robotic systems. Digital computers are discussed that use the binary number system or the related octal and hexadecimal systems. Those not familiar with these number systems before proceeding with this chapter.

Some of the functions to be supplied by computer hardware in robotic systems are discussed first. The essential functions are supervisory control, trajectory calculation and control, vision and sensory information processing, several types of monitoring functions, communication with related equipment through input/output inter-faces peripheral equipment functions to provide for data storage, printing, and system support.

Fundamental concepts of logic circuits and their use for control and calculation in digital computers are covered briefly in this chapter. Then the overall organization of computer central processing units is described. Peripheral equipment used in computers-disks, tapes, printers, and similar data processing equipment-is described briefly. Then the more esoteric and advanced equipment used as part of robotic systems is discussed: vision systems, sensors, and robot servos. The important area of input/output equipment is described as it relates to robotic systems, along with consideration of the problems of interfacing, interrupts, and interrupts handling. Then overall system organization of computers, robots, and associated equipment is described. Examples of present and proposed robot systems are described to complete the overview of computer application to robotics.

5.1 HARDWARE NEEDS FOR ROBOT SYSTEM

Some of the hardware needed by robotic systems is essentially the same as that of the usual data processing systems. Many of the functions required can be performed by microprocessors and minicomputers used for other commercial and industrial applications. Some vision and some of the more complex control calculations require high-speed computation capability, which is only available in parallel processors and large mainframe computers. These advanced systems are drscussed in Section 5.8.

5.1.1 System Supervision

Digital computers provide a unique capability to supervise the many complex and interacting operations being performed, often simultaneously, in a robotic system. Operation of the robot manipulator arms must be controlled and synchronized with the operation of associated vision systems, sensors, and auxiliary equipment. At the same time, the supervisory function must gather operational data to record the status and activity of the system. In addition, it must communicate with other associated systems and perhaps with a higher-level control system.

5.1.2 Trajectory Calculation and Control

Movement of the robot arm through a controlled trajectory may be relatively simple or may challenge the total computing capability of the computer system. In many robot systems, one or more separate computers are used for each degree of freedom of the manipulator. Calculations performed include matrix multiplication, matrix inversion, and other complex mathematical functions. Advanced systems must incorporate obstacle avoidance in the trajectory calculation.

5.1.3 Hardware Vision

Typically, the vision function is provided by a separate computer system that communicates its results to the supervisory control computer upon request. Some robotic languages have commands to specify to the vision system the location to be examined. The vision system then does the necessary scanning and analysis, and reports the result of this work to the supervisory or control computer in the form of tables placed in a specified location in memory accessible to both computers. This communication requires that both computers use an agreed-upon sequence of signals in a specified code.

5.1.4 Sensor Monitoring

As robotic systems become adaptive and intelligent, more and more sensors will be used .Some of these sensors will have their own associated microcomputers or logic functions. Other sensors will require periodic scanning by the control or supervisory computer to gather the necessary information. Sensory information may cause high-priority interrupts of the main processing and control function. These interrupts must be handled quickly, in most cases, to ensure that the robotic system does not damage itself or its environment. Sensors monitoring force, torque, and position will provide a constant stream of input to be acted upon by the control computer, and perhaps analyzed and recorded by a computer program assigned to this task.

5.1.5 Safety Monitoring

Safety monitoring depends on specialized sensors in addition to the many sensors used for motion control and position monitoring. Some systems now have a separate microcomputer assigned to this task. It has the ability to override the control computer and stop or reverse motion and other functions if necessary, much like the reflex actions in a human arm, hand, or finger. Light beams and acoustic sensors are especially useful for surveillance of the volume surrounding the robot, while temperature, pressure, and other sensors may warn of other types of danger. It is necessary to plan for safety and design in the required safety control systems as part of the basic system design in order to obtain a reliable, safe system.

5.1.6 Input/Output

Communication of data and control between the components of the robotic system requires that suitable interfaces and communication standards be established between these parts of the system. Data storage is commonly done on magnetic disks. Output data is sent out to printers, display consoles, or remote monitors. Status information is received from auxiliary equipment. Commands are sent to this auxiliary equipment over the input/output lines.

There may be multiple levels of control in the system. Disk files usually have their own controller, which controls several disk drives. This controller accepts data from the central processor or control computer, converts it appropriately, and stores it on the disk file. On request, it locates stored information and sends it back to the central processor. Other equipment has similar input/output needs. These topics are discussed in more detail in Sections 5.2 and 5.3.

1. Storage registers

Storage registers are used to hold information temporarily while it is awaiting transfer or being used in an operation or a control function. Information from memory is transferred into a Memory Data Register (MDR), held there for a clock cycle, and transferred to another register, perhaps the accumulator register, which holds information for input to the Arithmetic and Logic Unit or ALU. These registers must have the ability to read in information from one source and read it out to another register, memory, or the ALU. They are therefore provided with input and output switching circuits of many types depending on the use to be made of the register.

2. Shift Registers

Many computer registers are shift registers; they are capable of accepting information at one end of the register and shifting it toward the other end, 1 bit at a time. Suppose an 8-bit register is set to 00000000 initially. Then if we apply a pattern such as the binary number 00110101 to the left end and shift it 1 bit at a time, we will see the following pattern of bits as the new number is shifted in. These steps assume that the right end of the new number is entered into the left end of the register.

Initial condition	0	0	0	0	0	0	0	0
Step 1	1	0	0	0	0	0	0	0
Step 2	0	1	0	0	0	0	0	0
Step 3	1	0	1	0	0	0	0	0
Step 4	0	1	0	1	0	0	0	0
Step 5	1	0	1	0	1	0	0	0
Step 6	1	1	0	1	0	1	0	0
Step 7	0	1	1	0	1	0	1	0
Step 8	0	0	1	1	0	1	0	1

After eight pulse times, the complete new number has been shifted into the register. It could have come from another register, it could have been a serial input set of pulses from an external device. Shift registers usually have the ability to read in or out from either end. In addition, they can read in or out all binary digits at once, in parallel.

Some shift registers can do end-around shifting. The condition of the right flip-flop of the register can be moved to the flip-flop at the left end and all others shifted one step to the right. End-around shifting can be done in either direction. To provide for the carry operation in arithmetic, an additional separate flip-flop is used to hold the high-order (left end) bit temporarily. This bit can then be tested to see if a carry operation occurred.

Instruction operations operate directly on registers for many of their functions. Control of shifting direction and amount is provided by the control registers of the computer.

5.2 PERIPHERAL EQUIPMENT

Around the edges or periphery of the computer for the robot system are the machines and devices that provide input to the computer and receive output from it. Note that input and output are referenced to the computer.

Computer input is stored information, vision input, sensory input, or status input from other devices. Output from the computer is data for storage, printing, or communication, robot servo control signals, and control signals for auxiliary equipment. All of the input/output devices are classified as peripheral equipment and make possible the useful operation of the robot system.

Figure 5.2.1 is an example of a complete system controlled by a main computer called the host computer. The LSI-11 Controller is used to control the robot arm itself.

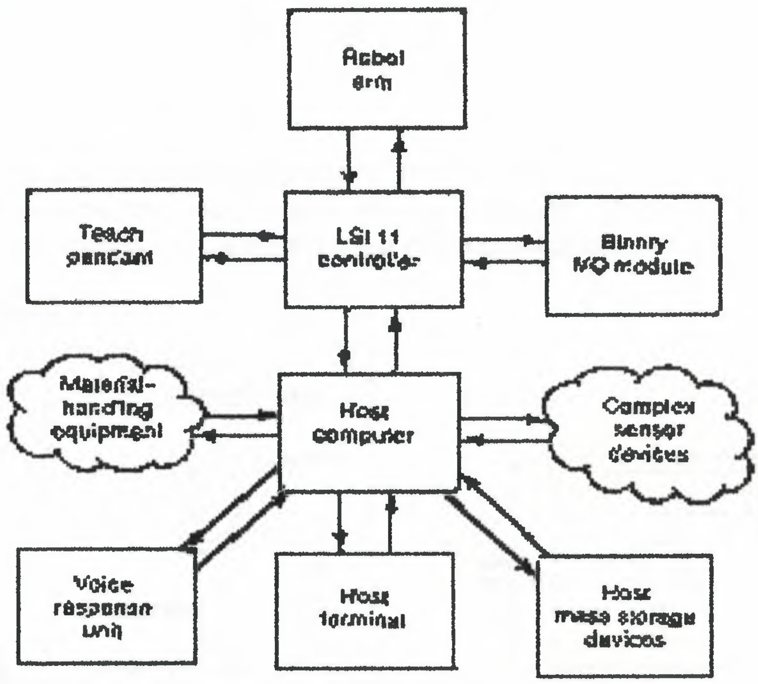


Figure 5.2.1 An integrated robot system with peripherals.

5.2.1 Data Storage Devices

Magnetic disks and magnetic tapes are the two chief storage media for robot systems. There are two general types of disks: hard disks and flexible or floppy disks. Tapes are available in large reels 24 inches in diameter and in two or three sizes of tape cassettes similar to the common tape cassettes used for audio recording. The characteristics of these disks and tapes will be discussed.

Older systems used magnetic tapes to store control programs, but magnetic disks are being used extensively in newer systems. Tapes are still useful for backup storage and for small systems in which the low cost of tape cassette recorders is important.

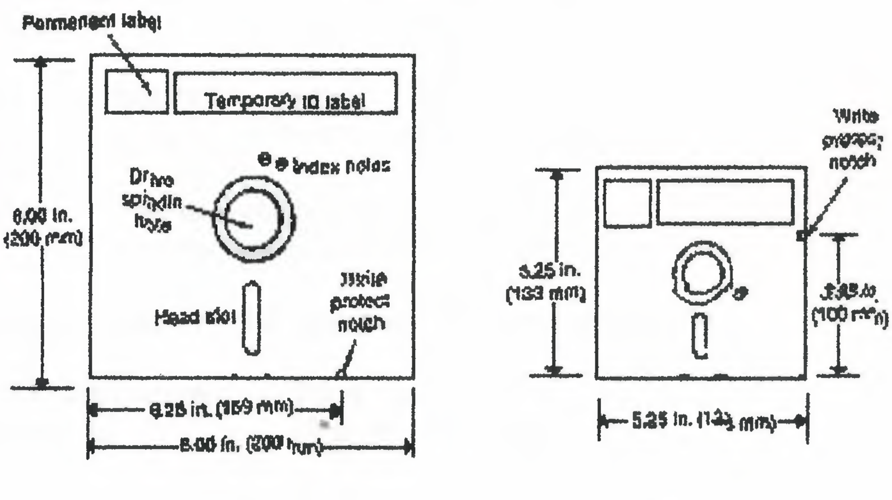
As prices decrease, the use of magnetic disks for control programs is expected to increase. Tape cassette storage now costs less than \$80 per tape unit , but floppy magnetic disk systems are available for as little as \$300 , so that the previous cost advantage of tape cassette storage is rapidly being eroded. Either method is capable of storing several hundred thousand bytes of information, and thus is more than adequate for many simpler applications.

Disks have the advantage of direct access; it is possible to locate data or a control program by mechanically moving a magnetic read head directly to the disk track where it is stored. Access time is in the range of 25 to 150 milliseconds depending on the type of disk and drive mechanism used. Access to information on tapes requires reading the tape serially until the desired information is found. Serial scanning of the tape may require several seconds or. even minutes depending on the length of the tape and the location of the information along the tape.

Disks are more reliable than tapes and usually easier to handle. Another advantage of disks is higher data transfer rates. Floppy disks transfer data at 180,000 to 360,000 bits per second, hard disks at more than 1 million bits per second. The reason is that the rotational speed of a disk can be higher than the linear speed of tape. When very high transfer rates are required, multiple heads can be used on disks, so that all tracks on a disk can be read at once.

1. Magnetic Disks

Large computers such as mainframes use 14-inch disks capable of storing as many as 600,000 bytes per disk. These disks have transfer rates of several megabytes per second. They are too expensive for most robot applications.



(a) *(b)*

Figure 5.2.2 Floppy disks in a protective envelope. (a) Standard 8-inch disk. (b) Standard 5.9 5-inch disk.

Robot systems typically use floppy disks. Floppy disks are made of a circular piece of magnetic tape with a center hole to accommodate the drive mechanism. They are flattened out by the centrifugal force due to the rotational speed of the drive mechanism. These disks are enclosed in a thin, square envelope of stiff paper lined on the inside with a soft material to protect the surface and keep it clean. In use, the floppy disk, in its envelope, is inserted into a drive mechanism that clamps the disk through a concentric drive hole in the envelope and rotates the disk inside the envelope. There are access holes in the envelope to allow disk heads to contact the magnetic surfaces and read information from the rotating disk. The standard form of the 8 inch and 5.25 inch diameter disks is shown in Figure 5.2.2. A smaller disk, 3.50 inches in diameter, is also available.

Floppy disks are identified as single-sided, single-lensity (SSSD), single-sided, double-density (SSDD), double-sided, single-density (DSSD), and double-sided, doubled-ensity (DSDD). Single-sided disks are recorded on only one side, double-sided disks on both sides. Similarly, single-density disks are recorded at 3,200 bits per lineal inch of track, double-sided disks at 6,400 bits per lineal inch. There are formatted and unformatted specifications because some of the recording area is required to put in control information. Also, there are hard-sectored and soft-sectored disks.

Hard-sectored disks have holes punched in a track around the disk to identify the beginning and end of a sector. Soft-sectored disks use a special binary code to identify the start of a sector. It is important, in purchasing disks, to buy the hard-sectored or soft-sectored disks if your system requires them. One type of disk cannot be run on a drive of the other type. Most blank disks can be run on double-sided drives, but there may be a reliability problem. There is a four-to-one difference in storage capacity between the SSSD and the DSDD disk formats, but the single-density disks are more reliable and allow more flexibility.

A disk is divided into sectors like a piece of pie. Each sector contains an identification mark and several tracks of information. Its characteristics are as follows:

1. Size: 8-inch diameter — Track density: 48 tracks per inch (tpi)
   Total tracks: 77 — 26 sectors per track

Format:	SSSD	SSDD	DSSD	DSDD
Capacity: (in bytes)	250K	500K	500K	1000K

Reliability: less than one soft error in 109 bits
less than one hard error in 10'2 bits

Soft errors are detectable by software; hard errors are undetectable.

2. Size: 5.25-inch diameter — Track density: 48 tpi
   Total tracks: 23 — 18 sectors per track

Format:	SSSD	SSDD	DSDD	DSDD
Capacity:	89K	180K	180K	360K

Hard disks are also available for use with microprocessors in 8-, 5.25, and 3.50-inch diameters. These disks are rigid, usually made of aluminum, and mounted on a rigid shaft. As a result, the accuracy of bit placement is greatly enhanced. Many more tracks per inch (tpi) of disk radius are possible (192 tpi, for example), and bit densities up to 10,000 bits per lineal inch of track are used. They are modeled after the IBM design of hard disks called the Winchester disk after the San Jose, California, location of the original development laboratory. Standard capacities of 10, 15, and 25 megabytes of storage are available on these disk systems.

Costs are in the $1000 to $3,000 range for complete disk drives, or roughly $100 per megabyte of storage. Access times can be as low as 25 milliseconds.

2. Magnetic Tapes

Magnetic tapes used in robotics are primarily one-eighth inch wide in standard tape cassettes. These tapes are identical to those used for video recording except that, in some cases, the magnetic material has been optimized for storing digital pulse information. Standard video tapes will work satisfactorily in most digital recording applications, however.

Data is recorded serially in magnetic tape cassettes. There are two tracks per tape, and each is normally recorded separately. Data rates are 185 bytes per second, so that storing a large program may take several seconds. Some recorders are capable of storing data at higher speeds.

Tape data is read by serially searching the tape until the specified program is found, as identified by an identification code written on the tape when the data was recorded.

An early example of tape cassette use, in 1974, was the Type 501 Unimate Cassette Program Storage Unit. The Type 501 was a rugged device weighing 22 pounds and encased in the heavy-gauge aluminum case. This device could store 400 program steps at the rate of 1 program step per second. It was used to copy programs from a magnetic drum in the Unimate o robot for storage and later use. vlore recently, Unimate o has converted to the use of floppy disks for program storage.

5.2.2 Printers

Printers are used to provide information output from computers. Typical output is system status information, data processing reports, the results of computations, or a copy of stored information for archiving and backup use.

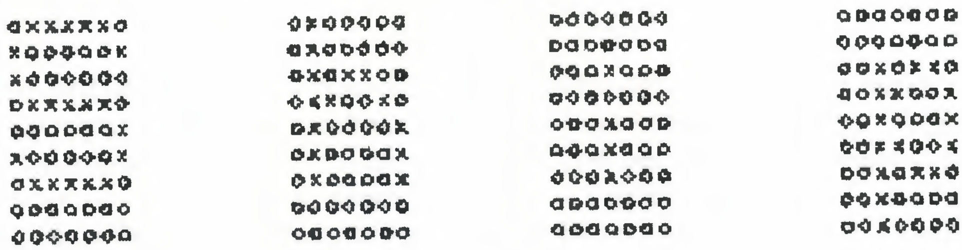
Major categories of printers are serial (or character) printers and line printers. These may each be subdivided into printers that form characters by mechanical pressure (impact printers) and nonimpact printers that form characters by the use of heat, light, sprayed ink droplets, or other non-mechanical means. Any of these printer categories can be further divided into dot matrix and typeface printers.

As an example, the familiar typewriter uses a ball or type bar with a typeface engraved on it to form images. Typewriters have a ribbon that is pressed against the paper by the type bar to form an image in the shape of the character on the type bar or ball. It prints one character at a time, serially, one after another. We would then classify the typewriter as a serial, impact printer using a typeface to form images.

1. Dot Matrix Serial Printers

Dot matrix printers form characters from an array or matrix of small dots. Typically, the dot matrix array is seven dots wide by nine dots high. In high quality dot matrix printers, the array may be 14 dots wide by 21 dots high to allow for maximum flexibility in forming characters such as M or W that are wider than the average character. Some dot matrix characters are shown in Figure 5.2.3.

In Figure 5.2.3(a), the dot matrix example has xs where the matrix is to print and Os to indicate a wire or hammer that will not print. In Figure 5.2.3(b), the appearance of the resulting four letters is shown, for the word Ship, in greatly enlarged form. Note the descender in the letter p that forms the portion of the character that descends below the print line.



(b)

(a)

Figure 5.2.3 (a) Dot matrix layout. (b) Enlarged view of the resulting printed characters.

Dot matrix printers may have only one column set of hammers or print wires. This one set is used repeatedly to form the complete character. At each step, the dots to be printed in that column are selected. To print a matrix that is seven dots wide by nine dots high, the set of hammers would be seven high and be used nine times per character. Since these printers can operate at rates of 50 to 150 characters per second, each set of hammers must step across the paper and print at 450 to 1,350 times per second, less than a millisecond per step in some cases.

2. Daisy Wheel Serial Printers

The most common type of serial printer to provide high-quality typeface quality is the daisy wheel printer. There is a radial element, or petal for each character to be printed. These petals are arranged in a circle about 3 inches in diameter on a central rotating shaft. Each petal has a character molded on it and is flexible in a direction perpendicular to the plane of the wheel. Daisy wheels are rotated at constant speed, and printing is done "on the fly." A single hammer is controlled to hit the selected petal at the exact instant when it is in position in front of the paper.

Control of printers is necessarily complex. The Diablo HYTYPE II printer showing the logic control blocks, the servos for the print wheel and carriage, the power amplifiers for the drive motors and controls, and the transducer for monitoring print wheel position precisely. An Intel 8080 microprocessor is used to monitor and control the printer's activities. It is of interest to note that the control capability provided is sufficient to control one joint of a robot arm.

This printer can move 48 increments per inch vertically and 120 increments per inch horizontally to provide for subscript and superscript printing and to print in the high-resolution graphics mode. It is also possible to control spacing between characters and lines in a very flexible way. Proportional spacing between characters can be obtained with this type of printer under the control of a word processing program such as Wordstar.

3. Nonimpact Serial Printers

Instead of requiring mechanical pressure and a ribbon to form a character image, the nonimpact printer may use heat, ink jets, or laser beams to form the image. Some fast, quiet printers use a dot matrix of wires that are selectively heated to form a character on heat-sensitive paper, typically a coated white paper with a black background. Ink jet printers form the character by spraying ink through an orifice and between plates that electrostatically deflect the ink jet vertically to form the character. Horizontal motion of the print head allows the complete character to be formed. This technique is illustrated in Figure 5.2.4. Ink not needed to form a particular character is recycled back to the ink reservoir.

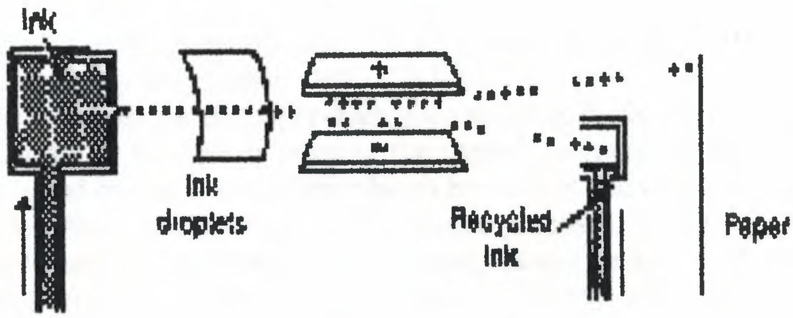


Figure 5.2.4 Ink jet print head .

4. Line Printers--Impact Type

High-speed printed output from mainframe computers and mini-computers is provided by line printers. Impact printers are reasonable in cost and provide high-quality output. In addition, they can make up to six carbon copies of each page of output due to the force of the impact. Impact line printers are made in two general configurations: rotating drum and moving horizontal chain.

Drum Printers

Rotating drum printers have a set of raised characters around the periphery of the drum for every character position; there are usually 132 positions. The drum is behind the paper. In front of the paper are a horizontal ribbon and a set of print hammers. As the selected character rotates into position behind the paper, the hammer hits the ribbon and presses it against the paper and the raised character on the drum. If a line of As is to be printed, for example, all 132 print hammers would strike the paper at once as the As in each character position came into position. In some printers the hammers are movable to one of six or more horizontal positions, thus requiring fewer hammers and control circuits but slowing the print speed by a factor of six. Timing errors in operating the hammers cause the characters to be displaced vertically, producing an uneven line. Drum printers print at 500 to 2,000 lines per minute, one line per complete rotation of the drum.

Chain Printers

Horizontally moving chains carry several sets of raised characters on the individual chain links, typically five sets of 48 characters. The continuously moving chain is in front of the paper is a set of hammers, one for each character position. The horizontal ribbon moves between the chain and the paper. When the selected character for each position is aligned properly, the hammer presses the paper against the ribbon and the chain to form the character. In this printer, an error in hammer timing causes a horizontal error that is less observable than the vertical position error found in the drum printer. For this reason and others, the chain printer has become the dominant impact line printer.

Line Printers-Nonimpact Type

Nonimpact line printers primarily use ink jet printing or laser printing. Ink jet printing. A high-intensity laser beam that is deflected by mirrors or other optical means to draw the character on the paper does laser printing. Printing can be done on light-sensitive material or by directly burning the paper. Computer output microfilm (COM) is often created by the use of a laser beam because of the high printing speed possible. This microfilm can then be used for storage purposes, or the image can be enlarged and duplicated by an electrostatic printing process such as that used in copiers made by Xerox and others. Printing rates of 20,000 lines per minute are obtained. These high-speed laser printers are expensive, in the range of $100,000 or more each.

Another nonimpact printing process forms the image by de positing an electric charge on the paper from a high-density line of electrodes across the paper. Several thousand electrodes are formed in a horizontal line across the paper and selectively supplied with a high voltage to charge the paper at a particular point. As the paper moves upward, the characters are formed line by line by controlling the charge pattern on the paper. Applying a fine resinous powder and heating, as in other electrostatic printing processes does development of the charge pattern. This method produces high-speed character printing and is also excellent for forming diagrams, pictures, and graphic output. It is also quite expensive, but is especially useful for duplicating drawings and diagrams for engineering and scientific use due to the high resolution of the resulting output.

5.2.3 Visual Display Terminals

Visual display consists of forming characters and figures on a screen to present information from the computer to human users. Such displays are also called monitors and are used wherever a human operator is involved in giving direction to or monitoring the operation.

Two types of visual display terminals are in general use: round and flat screens. Round screen Cathode Ray Tubes (CRT's) have been in use for many years. They are the same type of tube as the television tubes that are ubiquitous in our society. Other technologies are now being used to form a flat screen display that is more compacts than the CRT displays. These flat screen devices use liquid crystals electroluminescence, or gas plasmas to generate or control the image formed on the display. All of these displays depend on the control of electrical patterns on a grid or mosaic of individual elements. They operate like the dot matrix printers, except that the arrays are larger and require separate control lines for multiple X and Y positions, just as memories do.

Liquid crystal displays are similar to those used on many watches. An electric field is used to rotate tiny crystals so that they will pass or reflect light. This phenomenon is used to form flat screen images that are suitable for displaying characters and slowly varying phenomena. Typical screens are 60 to 80 characters wide and show up to 40 lines at a time. They are more expensive than CRTs but are compact and require only a small amount of power to operate.

Electroluminescent displays form a lighted image, usually in red or green, by selectively applying a relatively high voltage to a grid of cells and exciting them to produce light in a particular pattern. They are useful in making small displays but are more expensive than CRTs for large displays.

Gas plasma displays require a carefully controlled mixture of gases to be enclosed in a sealed glass container and excited by high voltages (100 to 500 volts) to form an image. They are capable of producing high-intensity light at rapidly varying rates in a flat, compact screen display unit. Control is expensive, but the resulting product is of sufficiently high quality to be useful in some display applications.

5.2.4 Peripheral Equipment Vision

Vision is an increasingly valuable sensory input for robots. It enables the robot to orient itself in space, locate and identify objects, and make measurements on parts in its field of view. Software in the robot control program can be used to send requests to a vision system that is a separate entity and receive back from the vision system the desired information on location, orientation, and the identity of objects. This input can then be analyzed and manipulated so that the robot control system can perform its assigned task.

Hardware and software for vision are described in detail and will not be discussed here. Interfacing of the video system with the robot control system has many of the same requirements as other input-output systems.

5.2.5 Sensors

Position, temperature, pressure, magnetic field, proximity, force, velocity, torque, stress, and many other types of sensory elements are required for robot system applications. These elements may be part of the positioning control of the robot itself or may be associated with auxiliary equipment used with the robot system.

Sensors and their operation are discussed in considerable. Sensory elements are especially likely to produce computer system interrupts to signal a particular situation. Therefore, their control systems must be carefully designed to handle the interrupt conditions that may occur.

5.2.6 Robot Servos

Servomotors may be controlled by either digital or analog servos. Each servo receives a command from a main control computer for the robot arm and carries out the command independently. Simple servos may perform their task with fixed or hard-wired electronic controls. However, the advanced robot systems use a microprocessor on each robot servo. These controllers are peripherals the main robot control computer.

A good example of a well-designed robot controller. Separate digital servo boards are used, as shown, in the PUMA 550/560 controller chassis. Each board is plugged into a connector on the motherboard or back plane of the controller and connects to the bus supplying power and signals to all of the control boards. A microprocessor is used on each digital servo board to perform the necessary calculations and comparisons to control one joint of the robot in response to commands from the main control computer, the Digital Equipment Corporation (DEC) LSI-11/2. The CPU, RAM memory, EPROM memory, and other control boards are plugged into the same bus and communicate as required with each other and with the digital control boards.

Position commands from the robot controller, the LSI-11/2, are sent to the digital control boards, where they are compared with the digital information from the encoders on each joint axis. The difference between the commanded position and the measured position is a digital number that is converted to an analog voltage (error signal) and sent to the analog servo amplifiers that directly control the drive motors. Motors rotate in such a way that the digital output of the encoders becomes the same as the digital input command, the error signal goes to 0, and the joint stops at the commanded position. Current feedback from the motors is used to stabilize the analog servos since it is a measure of motor velocity. This input is part of the Proportional, Integral, and Derivative (PID) control in the motor control servo loop.

Output from the controller is sent to the CRT, floppy disks, manual control panel, and the auxiliary vision system through the Quad Serial Board, which converts the digital signals from the LSI-11/2 to the correct serial or parallel form, at the correct voltage levels, to control these peripheral devices.

Hydraulic servos work in much the same way, with either analog or digital signals to control the electrohydraulic valves used to control oil flow in the hydraulic system.

Clock pulses are generated by the Clock/Terminator board at a 1 megahertz (MHz) rate for input to the servo interface board. This device also supplies an event timing pulse to the CPU to synchronize the sampling of events by the CPU. This pulse is fast enough to allow the CPU to monitor other equipment properly and have time to perform the computations required preparing commands for the next cycle of events.

5.2.7 Auxiliary Equipment

Either the host computer or the robot controller may provide control of auxiliary equipment. Auxiliary analog inputs from other devices are converted, in the Analog-to-Digital Converters, to digital signals for input to the controllers. Auxiliary devices may be conveyors, numerically controlled machine tools, furnaces, forging hammers, or other equipment working with the robot. In this way, the robot controller can insure that all devices operate in a controlled , synchronous manner to attain for example , a fairly complex sequence of signals must be transmitted between the robot control systems to ensure that they cooperate and do not collide with each other. Semaphores are protected signals used for this type of signaling, and provide increased security and speed of concurrent operation. Although they can be simulated by software, system efficiency is decreased. Interrupts, described in Section 5.2.3,captraps are similar. They are necessary for efficient handling of external operations and sensory information. Procedure calls are useful in efficient handling of subroutines because they provide automatic handling of the addressing required. Supervisor calls allow the program to address the operating system efficiently and are a valuable feature in robotic systems.

5.2.8 Advanced System Organization

Humanity never seems to be satisfied. As we get faster computers and more machines to serve us, the need for more and better machines seems to expand. This is also the situation with robots. The present robots are simple compared to the complex machines we can expect to see in a few years. In this section, we will look briefly at some of the needs for improved systems and possible ways to approach them. It is clearly beyond the scope of this book to describe how these machines might work, but it is of interest to discuss existing needs and the developments in process to fill them.

The computation required to control one robot arm could quickly exceed the capacity of a powerful microcomputer, a minicomputer, or even a mainframe computer. When we consider the computation required for the interaction of several robots, each with a vision system and tactile sensors, we see that it may become necessary to have more computing capability than is readily available in one computer. Even more computing power becomes necessary if we consider artificial intelligence, speech recognition, or other new areas of robot application.

Two basic approaches are used to provide more computing power. We can make faster individual computers or use more computers together. Faster individual computers are certainly feasible. New semiconductor circuits have been made that operate in 5 picoseconds, which is 1,000 times faster than circuits in common use today. However, it appears that using several computers together may be a quicker and less expensive way to solve some problems.

Multiple computers can be used in parallel, each doing part of a computation; this is called multiprocessing. Alternatively, they can work in serial and process information in assembly-line fashion, this is called pipelining.

Some computations can be broken down into pieces and done all at once. Matrix multiplication is an example. We can multiply a row of one matrix by a column of another matrix without interference between the operations. In principle, the matrix multiplication of can be done by four microprocessors working together if we figure out how to coordinate their operation, a problem we will not discuss here. Figure 5.3.1 illustrates one way of doing matrix multiplication with four processors and two data streams. A systolic array is made up of processors connected to do a certain class of problems as efficiently as possible.

A systolic array can be designed to do pipelining efficiently. In Figure 5.3.2. One processor element (PE) can do 5 million operations per second (MOPS). With six PEs in series, it may be possible to do 30 MOPS with the same type of processor.

In general, there is no linear increase in speed with multiple processors on the same problem; that is, six processors may not speed

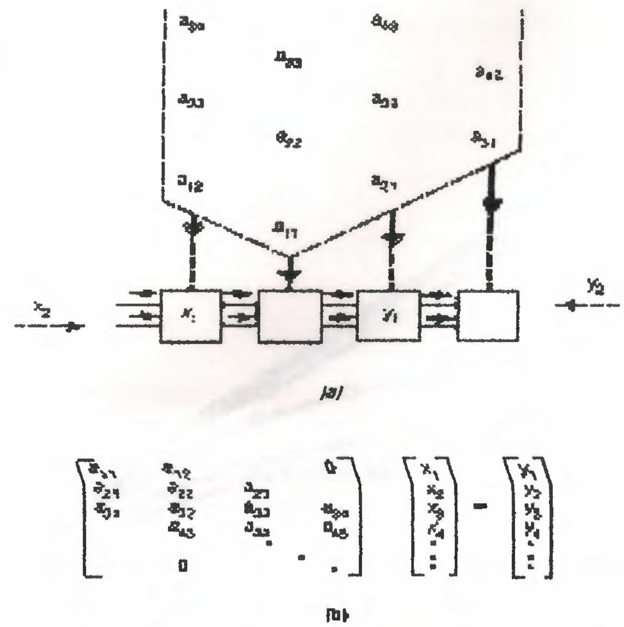


Figure 5.3.1 Linearly connected systolic array for computing the product of matrix a by vector x. (a) Linear systolic hardware; (b) Computation

Up the operation by six times. Problems of synchronization, overhead processing required for control, development of a suitable program (or algorithm), contention between processors for access to memory, or input/output problems may reduce the speedup to half as much as would be expected. If overhead is a big problem, there may be a limit on the total number of processors that can be used.

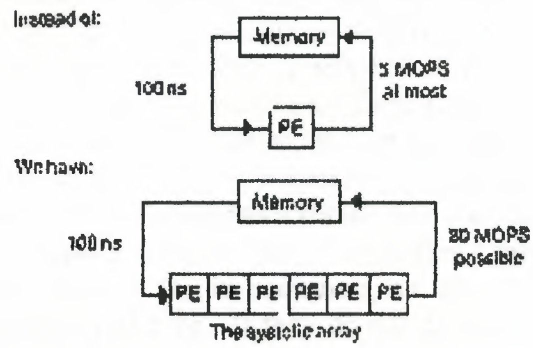


Figure 5.3.2 Basic principle of a systolic system.

Figure 5.3.3 is a graph of the speedup obtained for some actual multiple processor systems. Harpy is a speech recognition system developed at Carnegie Mellon University. Quicksort is an algorithm or procedure widely used for sorting large files. Both of these flatten out at a speedup of about four, even though six to eight processors are used together.

It is possible, for some types of problems, to get speedups of thousands of times. The Massively Parallel Processor (MPP) has 16,384 processors working in parallel. It has a speed of about 6 billion operations per second, a speedup on the order of 5,000 times. This high speed makes certain types of vision processing feasible.

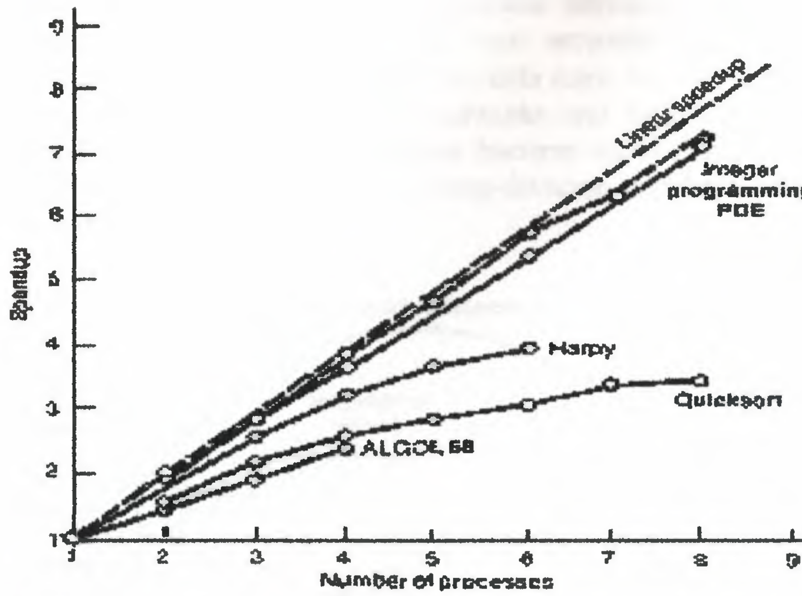


Figure 5.3.3 Performance of some sample programs using multiple processors.

5.3 NETWORKS

There are two basic types of networks: local area and long-distance networks. A local area network (LAN) is a data communications system that provides communication capability between a number of independent local devices. It might be used in one building, in several buildings on a site, or within a local area usually limited to a few thousand feet. LANs are usually owned by one organization. A LAN for a robotic system might allow communication between all of the robots, machines, and control units on a factory floor, for example. Data rates in a LAN are in the moderate to high rang up to as much a 10 million bits per second.

One popular LAN is the Ethernet. It is a passive system of stations located at specific devices or computers. Each station has a transmitter and receiver (transceiver) hooked to a common coaxial cable. They use a system called Carrier-Sense Multiple Access/Collision Detection (CSMA/CD). In operation, all the receivers listen on the coaxial line for a specified period of time. If they do not detect a carrier signal from some other device's transmitter, they start to send. Since there is a finite transmission time on the cable, it is possible for two stations to transmit at the same time. In that case, there is a "collision". Each detects the collision and waits a random length of time, and then they try again. This system has been found to be simple to implement, relatively inexpensive, and reliable in operation.

There are other LANs using various operating rules. Another method is a master-slave arrangement where one processor controls the network and determines which stations will transmit.

Long-distance networks may be provided by a common carrier such as Telnet or by one of the telephone companies such as American Bell. In this case, they operate like any other long-distance network, and customers pay tariffs for their use. Switching, message routing, flow control and other functions are provided by the long-speed distance network.

Local networks may be connected to long-distance networks, as shown in Figure 5.3.4. Interfaces are required between the local networks and the long distance networks to provide a way to match capabilities, data rates, signals, and so on.

A comprehensive discussion of local networks and long-distance networks is given in Schneidewind. As robotic systems become more complex, they can be expected to use both local networks and long-distance networks to provide data processing and control functions.

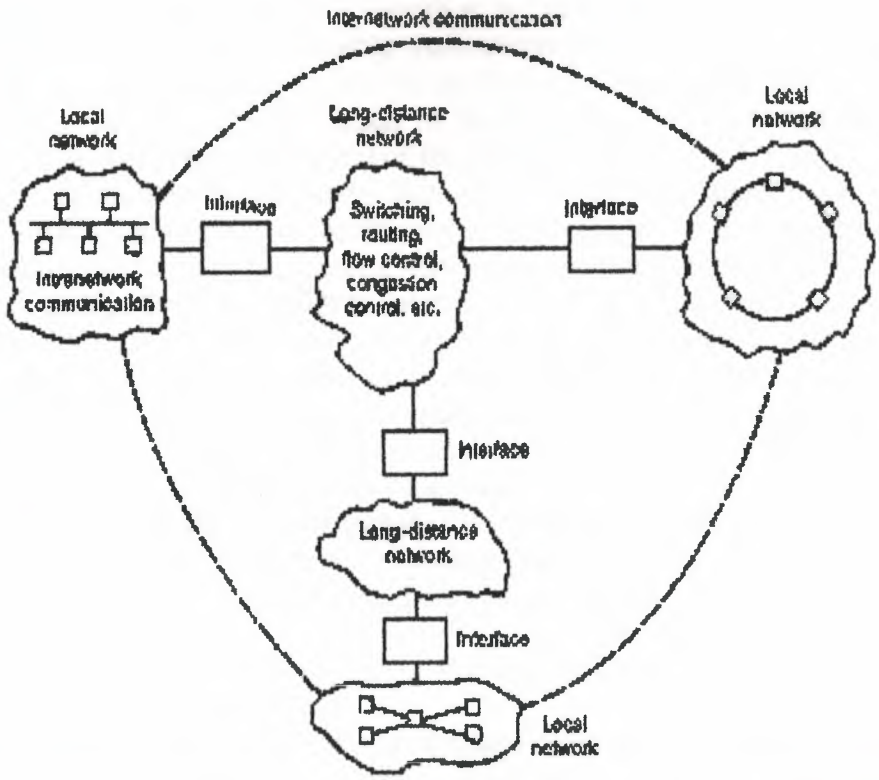


Figure 5.3.4 Connection of local networks and long-distance computer networks.

5.4 ROBOT PROGRAMMING LANGUAGES AND SYSTEMS

5.4.1 Introduction

In this chapter we begin to consider the interface between the human user and an industrial robot. It is by means of this interface that a user takes advantage of all the underlying mechanics and control algorithms that we have studied in previous chapters.

The sophistication of the user interface is becoming extremely important as manipulators and other programmable automation are applied to more and more demanding industrial applications. It turns out that the nature of the user interface is a very important concern. In fact, much of the challenge of the design and use of industrial robots focuses on this aspect of the problem.

Robot manipulators differentiate themselves from fixed automation by being "flexible," which means programmable. Not only are the movements of manipulators programmable, but through the use of sensors and communications with other factory automation, manipulators can adapt to variations as the task proceeds.

In considering the programming of manipulators, it is important to remember that they are typically only a minor part of an automated process. The term workcell is used to describe a local collection of equipment, which may include one or more manipulators, conveyor systems, parts feeders, and fixtures. At the next higher level, workcells might be interconnected in factorywide networks so that a central control computer can control the overall factory flow. Hence, the programming of manipulators is often considered within the broader problem of programming a variety of interconnected machines in an automated factory workcell.

5.4.2 The three levels of robot programming

There have been many styles of user interface developed for programming robots. Before the rapid proliferation of microcomputers in industry, robot controllers resembled the simple sequencers often used to control fixed automation. Modern approaches focus on computer programming, and issues in programming robots include all the issues faced in general computer programming, and more.

Teach by showing

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

Explicit robot programming languages

With the arrival of inexpensive and powerful computers, the trend has been increasingly toward programming robots via programs written in computer programming languages. Usually these computer-programming languages have special features, which apply to the problems of programming manipulators, and so are called robot-programming languages (RPLs). Most of the systems, which come equipped with a robot programming language, have also retained a teach-pendant style interface as well. Robot programming languages have taken on many forms as well. We will split them into three categories as follows:

1. Specialized manipulation languages. These robot-programming languages have been built by developing a completely new language which, while addressing robot-specific areas, may or may not be considered a general computer programming language. An example is the VAL language developed to control the industrial robots of Unimation, Inc □1□. VAL was developed especially as a manipulator control language, and as a general computer language it was quite weak. For example, it did not support floating-point numbers or character strings, and subroutines could not pass arguments. A more recent version, VAL II, now provides these features. Another example of a specialized manipulation language is AL, developed at Stanford University.

2. Robot library for an existing computer language. Starting with a popular computer language (e.g., Pascal) and adding a library of robot specific has developed these robot-ramming languages Subroutines. The user then writes a Pascal program making use of frequent calls to the predefined subroutine package for robot-specific needs. Examples include AR-BASIC from American Cimflex and Robot-BASIC from Intelledex, both of which are essentially subroutine libraries for a standard BASIC implementation. JARS, developed by NASA's Jet Propulsion Laboratory, is an example of such a robot programming language based on Pascal.

3. Robot library for a new general-purpose language. First creating a new general-purpose language as a programming base, and then supplying a library of predefined robot-specific subroutines have developed these robot-programming languages. An example of such a robot programming language is AML developed by IBM (7□. The Robot programming language KAREL, developed by GMF Robotics (8□, is also in this category, although the language is quite similar to Pascal.

Studies of actual application programs for robotic workcells have shown that a large percentage of the language statements are not robot-specific. Instead, a great deal of robot programming has to do with initialization, logic testing and branching, communication, etc. For this reason, a trend may develop to move away from developing special languages for robot programming, and toward developing extensions to general languages, as in categories 2 and 3 above.

Task-level programming languages

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

The border between explicit robot programming languages and task-level programming languages is quite distinct. Incremental advances are being made to explicit robot programming languages which help to ease programming, but these enhancements cannot be counted as components of a task-level programming system. 'In the task-level programming of manipulators does not exist yet but is an active topic of research.

5.4.3 A sample application

Figure 5.4.1 shows an automated workcell, which completes a small sub-assembly in a hypothetical manufacturing process. The workcell consists of a conveyor under computer control, which delivers a workpiece. A camera connected to a vision system is used to locate the workpiece on the conveyor. There is an industrial robot equipped with a force-sensing wrist. A small feeder located on the work surface supplies another part to the manipulator. A computer-controlled press may be loaded and unloaded by the robot, and finished assemblies are placed in a pallet.

The manipulator's controller controls the entire process in a sequence as follows:

1. The conveyor is signaled to start, and is stopped when the vision system reports that a bracket has been detected on the conveyor.
2. The vision system determines the bracket's position and orientation on the conveyor and inspects the bracket for defects such as the wrong number of holes.
3. Using the output of the vision system, the manipulator grasps the bracket with a specified force. The distance between the fingertips is checked to ensure that the bracket has been properly. Grasped. If it has not, the robot moves out of the way and the vision task is repeated.
4. The bracket is placed in the fixture on the work surface. At this point, the conveyor can be signaled to start again for the next bracket. That is. Steps 1 and 2 can begin in parallel with the following steps.
5. A pin is picked from the feeder and inserted partway into a tapered hole in the bracket. Force control is used to perform this insertion and to perform simple checks on its completion. If the pin feeder was empty, an operator is notified and the manipulator waits until commanded to resume by the operator.
6. The bracket-pin assembly is grasped by the robot and placed in the press.
7. The press is commanded to actuate, and presses the pin the rest of the way into the bracket. The press signals that it has completed, and the bracket is placed back into the fixture for a final inspection.

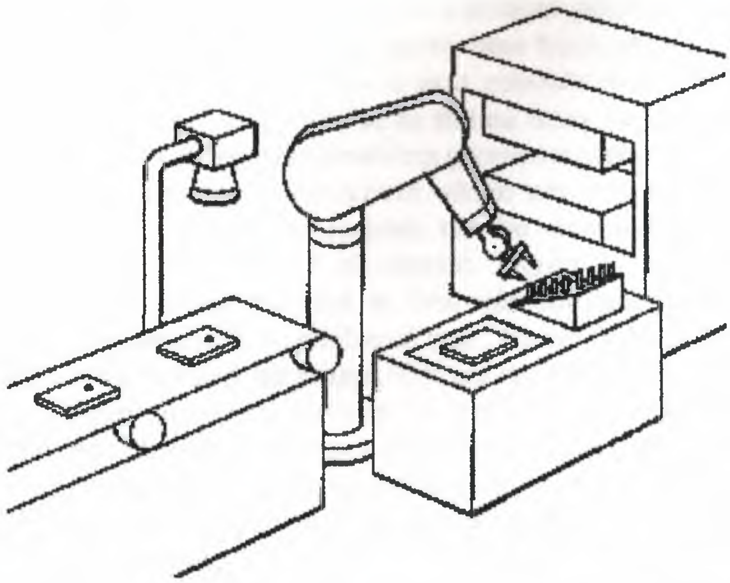


Figure 5.4.1 an automated workcell containing an industrial robot.

8. Using force sensing the assembly is checked for proper insertion of the pin. The manipulator senses the reaction force when it presses sideways on the pin, and can do several checks to determine how far
the pin protrudes from the bracket.
9. If the assembly is judged to be good, the robot places the finished part into the next available pallet location. If the pallet is full, the operator is signaled. If the assembly is bad, it is dropped into the trash bin.
10. Wait for step 2 (started earlier in parallel) to complete, then go to step 3.

This is an example of a task that is possible (though slightly challenging) for today's industrial robot .It should be clear that the definition of such a process through "teach by showing" techniques are probably not feasible. For example, in dealing with pallets, it is laborious to have to teach all the pallet compartment locations; it is much preferable to teach only the corner location and then compute the others making use of the dimensions of the pallet. Further, specifying interprocess signaling and setting up parallelism using a typical teach pendant or a menu-style.

Interface is usually not possible at all. This kind of application ne-cessitates a robot programming language approaches to process description. On the other hand, this application is too complex for any existing task-level languages to deal with directly. It is typical of the great many applications which must be addressed with an explicit robot programming approach. We will keep this sample application in mind as we discuss features of robot programming languages.

5.4.4 Requirements of a robot programming language

World modeling

Since manipulation programs must by definition involve moving objects in three-dimensional space, it is clear that any robot programming language needs a means of describing such actions. The most common element of robot programming languages is the existence of special geometric types. For example, types are introduced which are used to represent joint angle sets, as well as Cartesian positions, orientations, and frames. Predefined operators who can manipulate these types often are available. The "standard frames" introduced in might serve as a possible model of the world: All motions are described as tool frame relative to station frame, with goal frames being constructed from arbitrary expressions involving geometric types.

Given a robot-programming environment which supports geometric types, the robot and other machines, parts, and fixtures can be modeled by defining named variables associated with each object of interest. Figure 5.4.2 shows part of our example workcell with frames attached in task-relevant locations. Each of these frames would be represented with a variable of type "frame" in the robot program.

In many robot-programming languages, this ability to define named variables of various geometric types and refers to them in the program forms the basis of the world model. Note that the physical shapes of the objects are not part of such a world model, and neither are surfaces, volumes, masses, or other properties. The extent to which objects in the world are modeled is one of the basic design decisions made when designing a robot programming system. Most present-day systems support only the style just described.

Some world-modeling systems allow the notion of affixments between named objects. That is, the system can be notified that two or more named objects have become "affixed" and from then on, if one

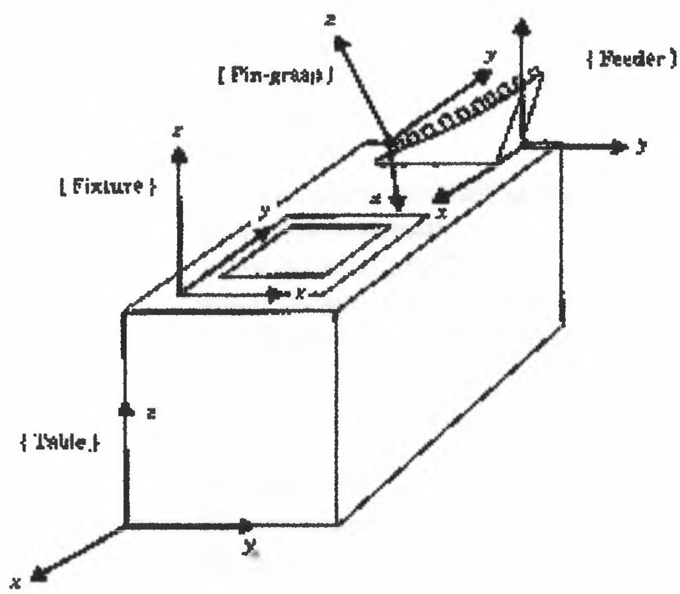


Figure 5.4.2 Often a workcell is modeled only by a set of frames which Are attached to relevant objects.

Object is explicitly moved with a language statement, any objects affixed to it are moved as well. Thus, in our application, once the pin has been inserted into the hole in the bracket, the system would be notified (via a language statement) that these two objects have become affixed. Subsequent motions of the bracket (that is, changes to the value of the frame variable "bracket") would cause the value stored for variable "pin" to be updated as well.

Ideally, a world-modeling system would include much more information about the objects with which the manipulator has to deal, and about the manipulator itself. For example, consider a system in which objects are described with CAD-style models, which represent the spatial shape of an object by giving definitions of its edges, surfaces, or volume. With such data available to the system, it begins to become possible to implement many of the features of a task-level programming system.

Motion specification

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

To illustrate various syntax's for motion primitives, we will consider the following example manipulator motions: 1) move to position "goal," then 2) move in a straight line to position "goal2," then 3) move without stopping through "via1" and come to rest at "goal3." Assuming all of these path points had already been taught or described textually, this program segment would be written as follows.

In VAL II:

move goal
moves goal
move via1
move goal3

In AL (here controlling the manipulator "garm"):

move garm to goal;
move garm to goal2 linearly;
move garm to goal3 via via1;

In Intelledex Robot-BASIC:

10 move goal
20 move straight goal2
30 cpon
40 move via1
50 move goal3
60 cpoff

Most languages have similar syntax for simple motion statements like these. Differences in the basic motion primitives from one robot programming language to another become more apparent if we consider features such as

1. The ability to do math on structured types like frames, vectors, and rotation matrices
2. The ability to describe geometric entities likes frames in several different convenient representations-with the ability to convert between representations
3. The ability to give constraints on the duration or velocity of a particular move. For example, many systems allow the user to set the speed to a fraction of maximum.

Fewer allow the user to specify a desired duration or a desired maximum joint velocity directly.

4. The ability to specify goals relative to various frames, including frames defined by the user and frames in motion (on a conveyor for example)

Flow of execution

As in more conventional computer programming languages, a robot programming system allows the user to specify the flow of execution. That is, concepts such as testing and branching, looping, calls to subroutines, and even interrupts are generally found in robot program-ming languages.

More so than in many computer applications, parallel processing is generally important in automated workcell applications. First of all, very often two or more robots are used in a single workcell and work simultaneously to reduce the cycle time of the process. But even in single-robot applications such as the one shown in Fig. 6.1. There is other workcell equipment, which must be controlled by the robot controller in a parallel fashion. Hence signal and wait primitives are often found in robot programming languages, and occasionally more sophisticated parallel execution constructs are provided.

Another frequent occurrence is the need to monitor various processes with some kind of sensor. Then, either by interrupts or through polling, the robot system must be able to respond to certain events, which are detected by the sensors. The ability easily to specify such event monitors is afforded by some robot programming languages.

Programming environment

As with any computer languages, a good programming environment helps to increase programmers' productivity. Manipulator programming is difficult and tends to be very interactive, with a lot of trial and error. If the user were forced to continually repeat the "edit-compile-run" cycle of compiled languages, productivity would be low. Therefore, most robot programming languages are now interpreted so that individual language statements can be run one at a time during program development and debugging. Typical programming support such as text editors debuggers. and a file system are also required.

Sensor integration

An extremely important part of robot. Programming has to do with interaction with sensors. The system should have the minimum capability
To query- touch and force sensors and use the response in if-then-else constructs. The ability to specify event monitors to watch for transitions on such sensors in a background mode is also very useful.

Integration with a vision system to send the manipulator system the coordinates of an object of interest. For example, in our sample application, a vision system locates the brackets on the conveyor belt and returns to the manipulator controller their position and orientation relative to the camera. Since the camera frame is known relative to the station frame a desired goal frame for the manipulator can be computed from this information.

Some sensors may be part of other equipment in the workcell. For example, some robot controllers can use input from a sensor attached to a conveyor belt so that the manipulator can track the belt's motion and acquire objects from the belt as it moves.

The interface to force control capabilities comes through special language statements, which allow the user to specify force strategies. Such force control strategies are by necessity an integrated part of the manipulator control system the robot programming language simply serves as an interface to those capabilities. Programming robots, which make use of active force control, may require other special features, such as the ability to display force data collected during a constrained motion.

In systems, which support active force control, the description of the desired force application may become part. Of the motion specification. The AL language describes active force control in the motion primitives by specifying six components of stiffness (three translational and three rotational) and a bias force. In this way the manipulator's apparent stiffness is programmable. To apply a force, usually' the stiffness is set to zero in that direction, and a bias force is specified. For example:

move garm to goal
with stiffness (80,80,0,100,100,100)
with force = 20 *ounces along that;

5.4.5 An example application coded in three RPLs

Perhaps the only way to gain an appreciation of the current state of the art in robot programming languages is to read some examples of robot programs in various languages. In this section we have chosen a palletizing example from and show the actual code to accomplish this task as expressed in three different robot-programming languages. Each of these programs solves the same scenario: pick a part from a pallet with rl rows and cL columns and put it into a pallet of r2 rows and c2 columns; signal or wait for presentation and removal of full or empty pallets. These programs are documented or self documenting so that you should be able to follow them with a careful reading.

Palletizing application written in AL

Below is an AL program that will accomplish the palletizing application.

```
BEGIN "Palletizing sample program"
  FAAME in-pallet, out-pallet, part;
  COMMENT
    The C1, 1) positions of the pallets and
    grasping position of the parts;
  VECTOR delsl, del-cl;
  VECTOA dels2, del-c2;
  COMMENT Relative displacements along the rows and columns;
  SCALAR rl, cl, irl, icl;
  SCALAA r2, c2, ir2, ic2;
  COMMENT counters;
  EVENT in-pallet-empty, in-pallet s eplaced;
```

```
EVENT out-pallet-full, out-pallets eplaced;
COMMENT
   Here insert the frame definitions for IN-ALLET
   and OUT-ALLET and the vector value for displacements
   along the rows and columns. These would be taught and
   recorded using the robot. FRAME definitions are typically
   unreadable by humans;
COMMENT
   Now define. the procedure PICK and PLACE called in the
   main program later on;
PROCEDUAE PICK;
BEGIN
   FRAME pick-rame;
   Irl: = irl + 1;
   IF irl GT rl
   THEN
   BEGIN
          irl := 1;
          icl := icl + 1;
          IF icl GT cl
          THEN
          BEGIN
             SIGNAL in-pallet-empty;
             WAIT in-pallet -seplaced;
             icl := 1;
        END;
     END;
     pick-frame := in-pallet+(irl-1)*del sl+(icl-1)*del-cl;
     MOVE BHAND TO pick-frame;
     CENTER BAAM;
     AFFIX part TO BARM;
   END;
   PROCEDURE PLACE;
   BEGIN
     FRAME place-frame;
     ir2:= ir2 + 1;
     IF ir2 GT r2
     THEN
     BEGIN
        ir2 := 1;
        ic2 := ic2 + 1;
        IF ic2 GT c2
        THEN
        BEGIN
          SIGNAL out-pallet-empty;
        WAIT out-pallet seplaced;
         ic2 := 1;
      END;
    END;
     place-frame := out-pallet+(ir2-1)*del-r2+(ic2-1)*del-c2;
```

```
    MOVE part TO place-frame;
    OPEN BHAND TO 3.0*IN;
    UNFIX part FROM BARM;
  END;
  COMMENT The main program;
  OPEN BHAND TO 3.0*IN;
  WHILE TRUE DO
  BEGIN
    PICK;
    PLACE;
  END;
END;
```

Palletizing application written in KAREL

Below is a KAREL program that will accomplish the palletizing aplica-tion.
Program PALLET

```
-- Transfers workpieces from one pallet to another.
var

-- Variables for the input pallet:
  BASE1 : position       -- (1,1) position on pallet
  IAl,IC1 : integer      -- counters for rows & cols
  NRl,NC1 : integer      -- limits for rows & cols
  DRl,DC1 : vector       -- delta between rows & cols
  ISIGl,OSIG1 : integer -- signals for pallet changing
-- Variables for the output pallet:
  BASE2 : position       -- (1,1) position on pallet
  IR2,IC2 : integer      - counters for rows 6 cols
  NR2,NC2 : integer      -- limits for rows ,6 cols
  DR2,DC2 : vector       -- delta between rows & cols
  ISIG2,OSIG2 : integer -- signals for pallet changing
---------------------------------------------------
routine PICK
-- Pick a workpiece from the input pallet.
var
  TARGET : position     - - target pose
begin
  IR1 = IR1 + 1
  if IR1 > NR1
  then
    IAl = 1
    ICl = ICl + 1
    if IC1 > NC1
    then
      IC1 = 1
          -- get a new pallet
      dout[OSIG1) = true
          - notify pallet-changer
```

```
    wait for din[ISIG1]+
        -- wait for input line to go high,
        -- meaning pallet has been changed
    dout [OSIG1] =false
        -- turn off our output signal
        -- compute target pose
   endif
  endif
  TARGET = BASE1
        - start with (1,1) pose
  shift(TARGET,CIA1-1)*DR1+(IC1-1)*DC1)
        -- shift for row and col offset
        -- get the part
  move near TARGET by 50   - move to 50 mm away from TARGET
  move to TARGET
  close hand 1
  move away 50   -- back away from TARGET by 50 mm
end PICK
routine PLACE
-- Place a workpiece on the output pallet.
```

Palletizing application written in VAL II

In the VAL II version of the palletizing application, the program transfers parts between two pallets using the external binary I/O signals to request additional pallets. It communicates with the user via the system terminal, asking questions and providing information on the system operation. A pallet location is taught by instructing the operator to move the robot to the corners of the pallet, using the manual teach pendant. The program then computes all locations in the pallet. Once both pallets have been taught, the robot transfers parts until manually stopped by the operator.

```
 PROGRAM main()

,
; ABSTRACT: This is the top-level program to move parts
; between two pallets. It allows the operator to teach
; the pallet locations if desired, and then moves parts
; from one pallet to the next.

; DATA STRUCTURES:

,
;  in.pallet[] = An array of locations for items on the
;                pallet to be unloaded.
,  in.height     = Approach/depart height for input pallet.
;  in.max        = The number of items on a full input pallet.
,  in.count      = The number of items left on this input
;                pallet.
,  out.pallet[]= An array of locations for items on the
;                pallet to be loaded.
,  out.height  = Approach/depart height for output pallet.
,  out.max        = The number of items on a full output pallet.
```

```
,  out.count        = The number of items left on this output
;                   pallet.
,  #Safe            = Safe robot location reachable from both
                    pallets.

   LOCAL $ans, in.count, out.count
; Define binary signal numbers used to control pallets
   transfer  = 1001  ;Input signal TRUE when transfers permitted
   in.ready  = 1002  ;Input signal TRUE when input pallet ready
   out.ready = 1003  ;Input signal TAUE when output pallet ready
   in.change = 4         ;Output signal requests new input pallet
   out.change= 5         ;Output signal requests new output pallet

; Ask operator about setup and teach neu pallets if desired
   PAOMPT "Do you want to define the pallet (Y/N):", $ans
   IF $ans == "Y" THEN
     DETACH()            ;Detach robot from program control
     TYPE "Use the PENDANT to teach the input pallet location"
     CALL setup.pallet(in.count, in.pallet[], in.height)
     TYPE "Use the PENDANT to teach the output pallet location"
     CALL setup.pallet(out.count, out.pallet[], pout.height)
     TYPE "Press the COMP button on the PENDANT to continue"
       ATTACH()          ;Attach robot (wait for COMP button)
     END
   , Initialize transfer data
     transfer.count = 0       ;Count of parts transferred
     in.count = 0            ;Assume empty input pallet at start
     out.count = 0           ;Assume full output pallet at start
   ; Wait for transfer signal, then start the pallet transfer
     MOVES #safe             ;Move robot to a safe place
     TYPE "Waiting for trasfer signal ...", /S
     WAIT SIG(transfer)        ;Wait until transfer signal seen
     TYPE "starting transfer", /C2
   ; Main loop transferring from one pallet to the other, requesting
   ; new pallets as necessary.  Quit when transfer signal becomes FALSE
     WHILE SIG(transfer) DO          ;while transfer is still ok...
         IF in.count <= 0 THEN; If out of input parts, ask for new
           SIGNAL in.change        ;Aequest pallet change
           WAIT SIG(-in.ready) ;Wait for input not ready
           WAIT SIG(in.ready)     ;Followed by input ready
           in.count = in.max      ;Indicate full pallet
       END
         IF out.count <= 0 THEN; If output pallet full, ask for new
           SIGNAL out.change       ;Request pallet change
           WAIT SIG (-out.ready);Wait for output not ready
           WAIT SIG (out.ready) ;Followed by output ready
           out.count = out.max ;Indicate empty pallet
       END
         ; Acquire input part
         OPEN                            ;Open gripper
```

```
        APPROS in.pallet[in.count], in.height     ;Move over part
        SPEED 20                            ;Move at 20% speed
        MOVES in.pallet[in.count]              ;Move to part
        CLOSEI                             ;Close immediately
        DEPARTS in.height                    ;Move up again
        in.count = in.count - 1               ;Count down
        ; Place output part
        APPROS out.pallet[out.count], out.height ;Move over output
        SPEED 20                            ;Move at 20%, speed
        MOVES out.pallet[out.count]              ;Move to empty place


        OPENI                              ;Open immediately
        DEPARTS out.height                    ;Move up again
        out.count = out.count - 1              ;Count down
        ; Count transfer and display it
        transfer.count = transfer.count + 1
        TYPE /U1, "Number of parts transferred:", /I8, transfer.count
    END   ;End of while loop
    ; All done transferring parts, move robot to safe place and quit
    MOVES #safe
  .END
  .PROGRAM setp.pallet(count, array[], approach)
  ; ABSTRACT: Routine to compute an array of locations given locations
  ; which represent the upper left, lower left, and lower right point
  ; of a pallet. All output locations have the orientation of the
  ; upper left part location.
;, INPUT PARM: None
; OUTpUT pARM: count   = Number of items on this pallet.
,       array[] = Array containing the.pallet locations.
,       approach= The approach height for this pallet.

LOCAL ul, 11, lr, AP, t[], ncol, nrow
LOCAL row, col, cs, rs, i, frame
; Ask operator to teach pallet locations
CALL teach.point ("upper left pallet position", ul)
CALL teach.point ("lower left pallet position", 11)
CALL teach.point ("lower right pallet position", lr)
CALL teach.point ("approach height above the pallet", ap)
PROMPT "Enter the number of columns (left to right): ",ncol
PROMPT "Enter the number of rows (top to bottom): ", nrow
count = ncol*nrow               ;Compute count of items
, Setup to compute pallet locations
cs = 0                     ;Assume 1 column, zero spacing
IF ncol > 1 THEN
  cs = DISTANCE(ll,lr)/Cncol-1) ;Compute spacing of columns
END
rs = 0                     ;Assume 1 row, zero spacing
IF nrow > 1 THEN
  rs = -DISTANCE Cul,ll)/(nrow-1);Compute spacing of rows
```

```
END
; Compute frame values
SET frame = FRAME (ll,lr,ul,ul)  ;Compute frame for pallet
approach = DZCINVERSE(frame):ap);Compute approach height
                                   ; wrt frame plane
_DECOMPOSE t [1]  = ul
; Loop to compute array values
i = 1
FOR row = 0 to nrow-1
   FOR col = 0 to ncol-1
      SET array [i7  = frame: TRANS (row*rs, col*cs,0,t[4],t[5],t[6]))
      i = i + 1
   END
END
RETURN
.END
```

5.4.6 Problems peculiar to robot programming languages

While advances in recent years have helped, programming robots is still difficult. Robot programming shares all the problems of conventional computer programming, plus some additional difficulties caused by effects of the physical world.

CHAPTER 6

INDUSTRIAL APPLICATION EXAMPLES

6.1 MACHINE TOOLS LOADING APPLICATION

There is already a classical layout for using robots in machine tool applications. Figure 6.1 is a fairly typical example. The robot stands stage center surrounded by the machines it tends.

Metal cutting, especially when the parts are large and the volume is low to medium, is a lengthy process, which means that the part spends a long time in station. If the cutting cycle exceeds about 20 seconds and particularly if the time of dwell runs into minutes, it does not make economic sense to have the robot unoccupied while the machining cycle is completed. Secondary and tertiary assignments for the robot should, in such cases, always are sought.

This particular installation has been producing completely machined valve bodies on a two-shift basis for over four years with high reliability.

Without stretching the analogies between human operators and robots too far, it is a fact that when a single operator tends several machines, each with long cycle times, he must walk between them and keep them all going. When machining cycles are particularly long, the robot handling workpieces can be made to travel among more machine tools than can conveniently be placed around a stationary robot. The robot depicted handles no less than eleven different machine tools and even carries along with it a buffer station for parts in intermediate stages of completion.

It can be noted in passing that the robot of today likes some room to swing its arm, and therefore to adapt an existing transfer line to robots may be difficult. Transfer line machines are usually put as close together as possible; the moving robot may therefore be the only solution if the factory layout does not permit of sufficient room for wider spacing to accommodate stationary robots. However the final design is usually on economic grounds, and if one moving robot can do the job of three or four stationary ones the technique speaks for itself.

All of the installations described thus far are to be found in Japan. According to recent estimates, the Japanese have more robots in action than any other nation. Of these, they most definitely lead the way in machine tool applications which lends support to the suggestion voiced in the press cutting - that soon Japan will be operating completely robotized automatic factories with the ability to change products by changing computer and robot programs.

In the USA, Xerox Corporation has a high volume line using three robots, a transfer conveyor, and two center-driven C.N.C. lathes with double-end capability, and a supporting cast of brazing, grinding, broaching and turning machines. The layout is as shown in Figure 6.2. The three robots have a 10 feet reach and are used to transfer the parts between the conveyor and the machine tools.

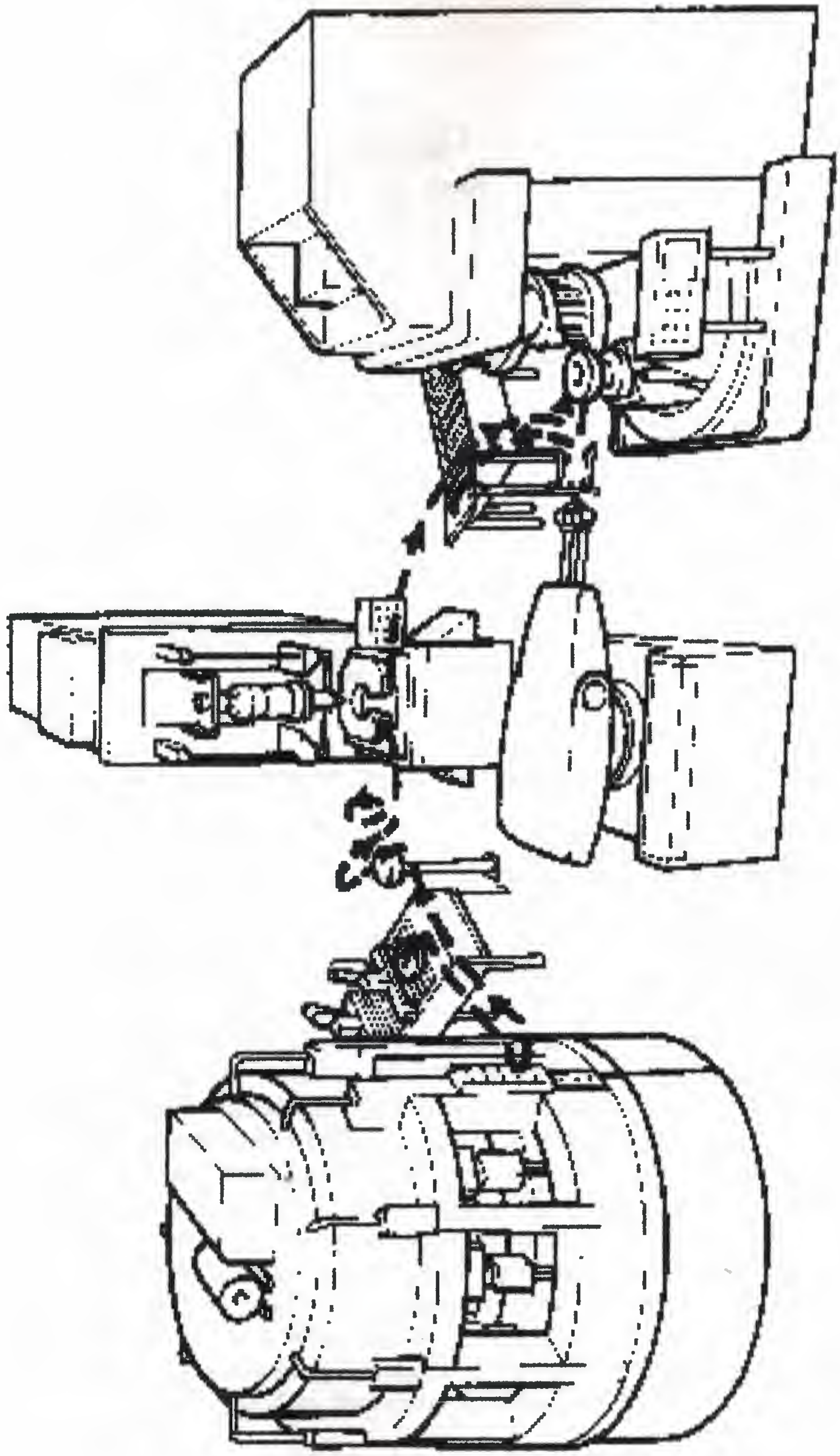


Figure 6.1 Typical layout for applying robots to machining applications

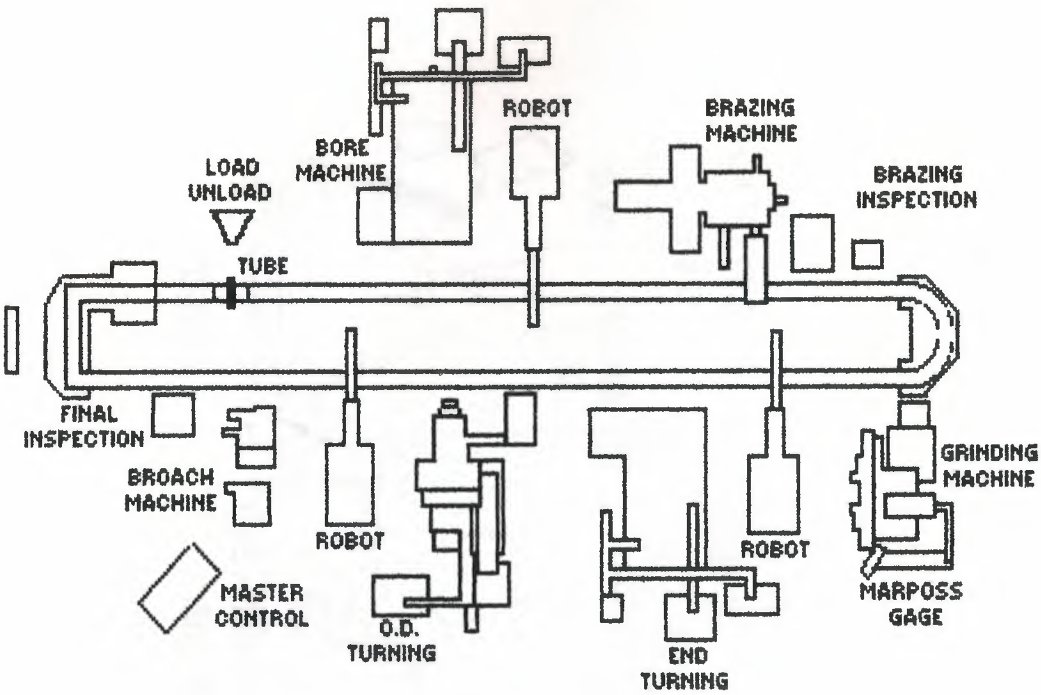


Figure 6.2. Layout of three-robot line in machine shop at Xerox Corporation

Nine basic programs will accommodate the entire family of parts made, while alternate bypass programs come into operation during any machine tool downtime. These programs are on cassettes to permit rapid product changes to be made. To oversee the whole operation, a supervisory programmable controller takes care of automatic program selection, the synchronizing of robot movements with other line operations, cycle initiation, fault monitoring and emergency shutdown. Each of the three robots has two hands, which can be used independently to clamp and unclamp the parts. This speeds up the handling of finished and unfinished parts at each operation, keeping the load and unloads cycle time to an absolute minimum.

Massey Ferguson produces four different sizes of planetary gears in a schedule in which the volume can vary from low to high. The company was contemplating a transfer line but decided instead to opt for the greater flexibility of a robotized line. The installation, which resulted, is shown in schematic form in Figure 6.3. Each of the three robots has its own workstation and each performs a different operation. Conveyor transfer and inter-station buffer storage are conventional.

Some idea of the flexibility achieved can be gained from the fact that each of the three robots can search for availability among its own machine tool complement to keep the line working during random downtime or scheduled tool changes. Willing workers indeed!

A robot transfer line has a considerable edge over a hard automation line and compared with manual operation, production is 25% faster. The complete system Fayback in this case is estimated at 21/a years. This is good news for the large scale, big company manufacturer. But what about the small operator who has to contend with batch manufacturing? Well, he is not out of the running, even at this early stage of robotized machine tools. A recent development, aimed just at this problem, is to integrate a robot into a system of conventional NC equipment as shown in Figure 6.4. With the concept comes a parts classification system, which will be very useful in work-loading the system.

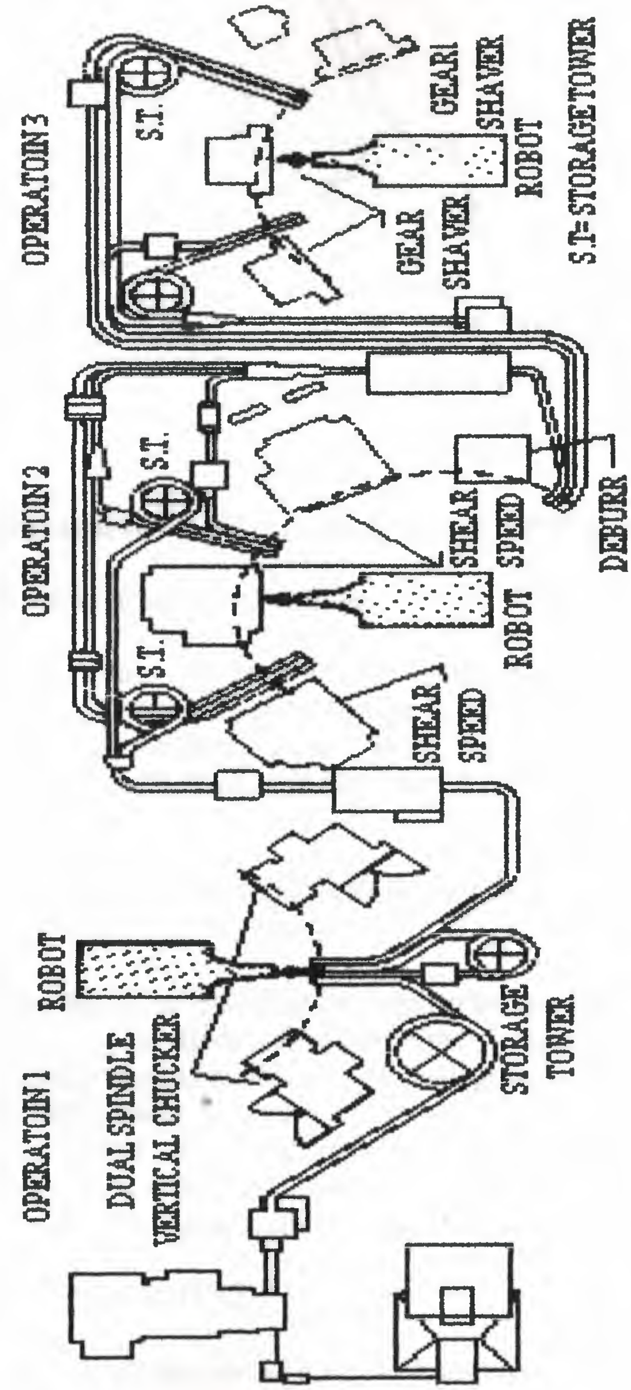


Figure 6.3 Layout of three robots on machine line at Massey Ferguson

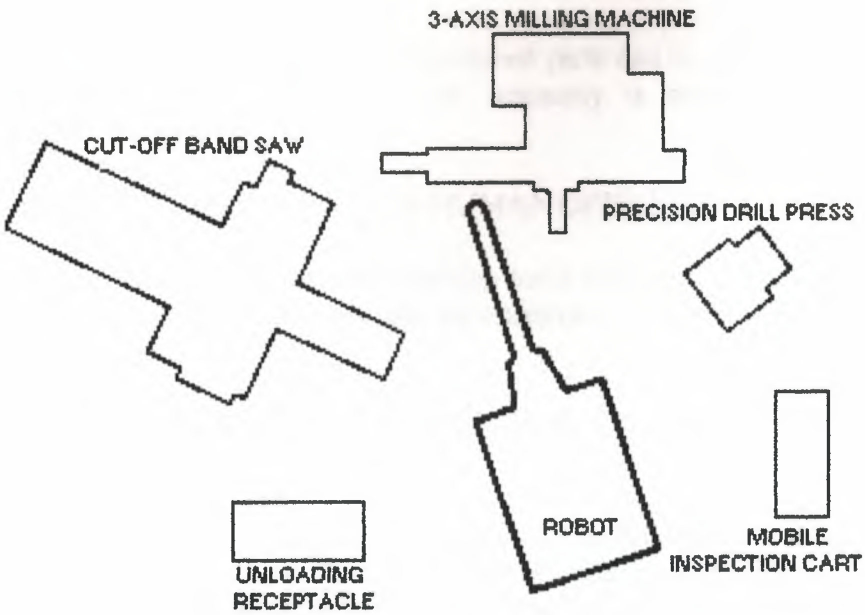


Figure 6.4 integrated robot N.C system for small batch manufacture

Robot attributes for machine tool applications

In machine tool applications (just as in any other role) it goes without saying that the robot must have a long enough reach to service all work stations and be able to carry the heaviest parts being handled by the system. The following checklist spells out other desirable features, some being essential while others would simply make the job a great deal easier.

1: UP TO SIX INFINTTELY CONTROLLABLE ARTICULATIONS BETWEEN ROBOT BASE AND GRIPPER APPENDAGE

Sometimes it seems that three articulations ought to be enough to load a machine tool chuck, but the real world finds otherwise. Often manipulation patterns are complex to avoid machine structure. The use of dual grippers compounds motion requirements and the requirement to lend a variety of machine tools may demand great variation in load-unload paths. And palletizing speaks of a variety of arm destinations. As to making the articulations infinitely controllable, one can only note that the `limited sequence' robot rapidly runs out of useful arm destinations.

2: FAST `HANDS-ON' INSTINCTIVE

In a machining system, it's nice to do a layout knowing that anyplace reachable by the robot can be quickly programmed on the job. If a range of parts must be handled, all programs should be easily generated and stored for use as needed. This is most easily done using `record playback' instinctive teaching with actual parts at hand.

3: REPEATABILITY TO 0.3 mm.

Quite a few raw castings and rough-machined parts can be given final orientation by ingenious centering grippers, but still, accuracy is necessary to center the workpiece into the machine tool checks.

4: SPEED EQUIVALENT TO THAT OF HUMAN OPERATOR

If a robot is slow on the job, the machine tools will not be utilized at optimum capacity. A loss in productivity can eat up economic benefit derived from human operator replacement.

5: PROGRAM SELECTION CAPABILITY FOR CUTOUT OR ALTERNATE ACTION

Alternate action is very commonly needed in a robotized machining center. A machine being serviced will demand that arriving parts be buffered. Perhaps any of groups of identical machines must be bypassed on occasion. If an inspection station is included, workpiece destination will depend upon commands from this station.

6: COMPATIBILITY WTTH WIDER NC SYSTEMS

There are sophisticated systems that control machine tools in the DNC or CNC mode and they are under continual command changes. The system s robot should be able to respond with equal alacrity. Sometimes, it is possible, at cost advantage, for a robot to share a NC controller that has spare capacity and a compatible interface.

7: PALLETIZING AND DEPALLETIZING CAPABILITY

In many applications, it is not practical to present workpieces one at a time in a single location. To avoid human attendance, parts may be delivered on a pallet to the workstation, and the robot may also palletize the output. Palletizing and depalletizing capability provides for inter machine buffer storage so that a system can stay on stream. When machine elements are temporarily down.

8: LOCAL AND LIBRARY PROGRAM ACCEPTANCE

As long as a robotized machining center is designed to handle a range of parts, then a range of programs will be necessary for the robot or robots. Some of these programs must be on hand for instantaneous recall, others may be extracted for external storage and later convenient introduction into the local memory.

9: HIGH RELIABILTTY - NOT WORSE THAN 400 HOURS MTBF

A robot tending a machine tool - or worse still, a group of machine tools, threatens the system with a `domino effect'. If the robot goes down, the system goes down and production is lost. Human operators could stand in, but that defeats the labor saving purpose.

The answer is high reliability, better than 400 hour MTBF, mean-time-between-failure. If the robot also has a low MTTR, mean-time-to-repair, then it can be more dependable than human labor. Two- percent downtime is a reasonable demand, outstripping human downtime, which now runs about 3.5% in USA metalworking.

10: DUAL GRIPPER CAPABILITY

The dual gripper capability is related to point 4, speed. With dual grippers lost motion is avoided, since the robot arm can strip a completed part and load a new part without leaving the machine tool bed. Naturally, point 1 figures in, because manipulative power is essential in handling two hands mounted at the end of one arm.

11: MOBILITY IN WORK AREA

As we have seen, there are applications where a single robot can tend to conveniently arrange around a stationary robot. Then, the robot should be mobile. Human flexibility is not necessary; only the ability to move linearly, as on tracks, among workstations.

12: AUTOMATIC SENSING FOR CHUCK ALIGNMENT

Often, parts do not have symmetry around the chuck centerline and the parts must be rotated to random locations to permit engagement. A special robot gripper sensing mechanism performs this task. Better still, newer machine tools provide for fixed chuck destinations so that the robot need not seek engagement.

13: ADAPTIVITY - INCLUDING RUDIMENTARY VISION

Adaptivity can cover all of the capabilities that a human brings to the task. Use of instrumentation can help the robot to react similarly. So it is with tooling wear and breakage and finished part inspection. Scrap accumulation, where chip breaking is inadequate, may require rudimentary vision. That's on the way, and a robot worth its salt will accept vision module inputs.

Robotics and NC are obviously complementary in control technology, but they are also part of a larger manufacturing technology scene that is inexorably driving industry toward the goal of unmanned manufacturing. Impetus also comes from workers who rightly rebel against debilitating factory jobs and exact ever increasing `hazard pay' to push up manufacturing costs still further.

6.2 PLASTIC MOLDING APPLICATION

From the above brief descriptions of the molding process, it will be obvious that the operation has much in common with die-casting. In injection molding, for example, the charge of raw material is injected automatically, just as it is in the die casting machine. After the ram operates, it is now necessary to remove the finished part from the dies. If this is done by hand, the operator places himself at risk by putting his hands and arms in the die area. Just as in die casting, as a safety factor, some molding machines are provided with ejector pins, which push out the part, which then falls into a suitable container.

One of the characteristics of plastic molding is the relatively long cycle time. It is this time which the human operator utilizes to perform secondary operations such as trimming and packaging. These operations are not easily robotized. For example, the trimming of flash is not predictable, and to try to use special dies for this purpose is not cost effective when compared with a human operator using a penknife and eyesight. However robots are already working in plastic molding plants, though they are best suited to dealing with large moldings such as garbage cans. In the larger presses needed for parts of this size the operator is at much greater risk when he physically enters the die space to extract the parts.

Sooner or later the industry will adapt robots on a larger scale because by doing so the process will first of all be better rationalized and because some of the more complex molding procedures such as those requiring steel inserts will lend themselves to higher production rates and better quality when tended by robots. One factor that must be taken into account is the environment. Compared with many of the industrial processes today, the plastic molding shop is relatively benign. The plants tend to be clean and involve no hazards or even unpleasantness for operators.

Current robot use in plastic molding

To date robots have successfully accomplished the following tasks in plastic molding plants:

O Unload one or two injection molding machines
O Trim moldings on removal from machine
O Load inserts into the mold
O Palletize the moldings for despatch
O Package the moldings

It is of interest to look in a little more detail at some of these robot applications.

In one plant a robot is used to unload two injection molding machines making elastomer rubber parts. The specially designed hand enters the open press, strips the part from the die using a combination vacuum and mechanical gripper and removes the part from the press. These are large parts, and when human operators were used they required frequent relief from the noxious fumes which were present in an operating environment approaching a temperature of 400 degrees.

The parts in this application are made two at a time, and therefore it is necessary to separate them. The robot placing the parts over a cutting blade accomplishes this. The part is then deposited on a conveyor and the sprue disposed of as the robot turns to unload the second injection-molding machine. The plant layout for this application is shown in Figure 6.5.

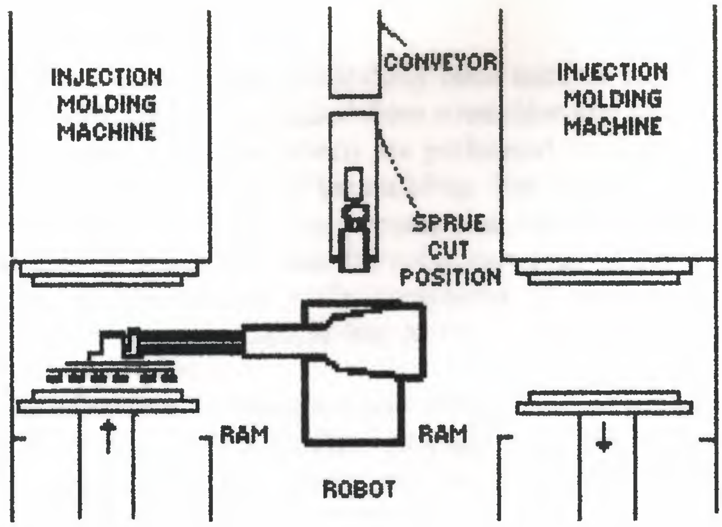


Figure 6.5 Plant layout for injection molding application

The special hand for this application consists of housing containing an air operated mechanical gripper mounted on either side of the housing by two arms pivoted by means of double acting air cylinders.

The mechanical gripper grasps the sprue and breaks the center portion of the part from the die. Air jets within the hand directed between the part and the mold help break the adhesion. Figure 6.6 illustrates the hand mechanism.

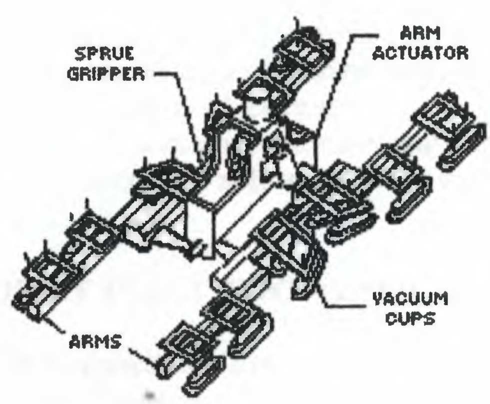


Figure 6.6 Special hand for injection molding application

The cost justification for this robot installation comes partly from increased productivity and from faster cycle times and the savings of one operator per shift for each press. Savings in removing operators from the die area of the press amounted to some $15,000, this rep-resenting the cost of equipment it would have been necessary to install to meet the safety regulations imposed by new legislation.

One of the problems ordinarily encountered by robots - that of part orientation - is eliminated entirely in the plastic molding process since the dies determine precisely the position of the part.

In another application a robot has successfully been used to work on a transfer molding installation. In this, a part is unloaded from a machine and then transferred to a second machine where secondary operations are performed, in this case cleaning a large number of holes and grinding part of the molding. The actual movements of the robot are no more complexes than operating one machine, but it is interesting to note that by means of random program selection the robot can automatically omit one or more of the transfer molding presses while continuing to operate the remaining presses. This is important when a machine has to be taken out of service without shutting down the entire operation.

It is clear that the robot does not have to carry out any new or unusual operations in a plastic molding application. It is a familiar `pick and place' job. Consequently the operation of stacking parts in complex patterns presents no difficulty to the robot.

Again, by designing proper hands and grippers it is possible for a robot to load simple inserts into a molding machine before the cycle are initiated. This generally requires two sets of fingers, otherwise the inserts may have to be presented to the robot by means of a suitable conveyor system.

With the size of injection molding machines on the increase, clamping pressures of 5000 tons being by no means uncommon, the hazards to human operators are considerable and the use of robots is likely to increase dramatically for that reason alone. But there is another factor, which has only recently become obvious. The energy crisis has caused the cost of plastic materials to escalate alarmingly so that there are real incentives to lowering the direct labor cost to offset the material increases.

It can take several hours for a plastic molding machine to reach a stable thermal condition and this is essential for the production of uniform parts, both dimensionally and from the point of view of finish. During the stabilization period parts have to be made and they are often wasted. If a human operator is attending the machine during this warm-up phase, extra costs are incurred, but if a robot is installed no direct labor costs are involved although there are of course some costs due to the need to amortize the robot investment during this non-productive period.

Although it is impossible to generalize with any real accuracy, a robot used in a plastic molding application should be able to pay for itself in a year or eighteen months of operation in a typical case.

6.3 HEAT TREATMENT APPLICATIONS

From the range of robot applications described thus far, it would appear that there is a real chance for these hard-working machines to enter this field, and indeed they have done so.

In die-casting, forging, presswork and molding, the robot has proved that it can thrust its hand into a die, a furnace, or the like, with high precision. It can place a part accurately or grasp one and move it somewhere else. These are all steps in the heat treatment process, added to which the robot will not feel the heat of the sample which it picks up and transfers to a cooling bay or to a quench tank.

No special features are needed for a robot to work in this application; it is a `pick and place' job, governed by synchronizing signals linked to the timing apparatus which determines the duration of the heating and cooling periods. Orientation of the part, at least for the unloading of a heat treatment furnace, is usually no problem, though steps may have to be taken to present the part to the robot if it is to load the furnace as well. There is a bonus to be earned too. Heat cycles are relatively lengthy affairs, so the robot has time on its hand. Rather than waste this, it can be put to other tasks during its free time, provided that the factory layout is such that the robot has the necessary equipment within reach.

In the Caterpillar Tractor factory, a robot picks up a pin from a magazine feeder and inserts it into a groove in the rotary hearth of an induction-heating furnace. The use of a groove provides an accurate registration for later pick-ups. After placing the pin, the robot picks up a hot pin, already in the furnace, removes it, signals the furnace door to close, and then places the hot pin in a quench tank. The whole cycle described takes less than one minute. By employing electrical interlocks, the robot is prevented from loading a pin into the furnace until it has been signaled that the hearth has rotated to accept it and the furnace door is open. Ejection of pins from the quencher is automatic, but the robot cannot load a pin into the quencher until it receives a signal telling it that the previous pin has been removed.

In another installation at a different factory, robots is busily operating a heat treatment and quench cycle on parts weighing 40 to 50 pounds. These are fan-shaped blades which are finally assembled into a helix for the inside of truck-mounted concrete mixer drums.

Before the days of the robot on this job, the blades were cut manually from steel plate by a rotary shear, and then cold-formed. Next they were transferred to a furnace area where gas torches arranged in a specific pattern heated the areas to be hardened. After heat treatment, the parts were quenched.

The furnace in this installation is arranged in four levels. After it has picked up a blade and placed it into a programmed position in the furnace, the robot unloads the blade that has been longest in the furnace, then triggers the door, which closes. The robot, into a 20-ton press then places the heated blade. As the robot withdraws from the press die-area, it signals the press to close on the blade to shape it. As the press concludes its operation and opens, this signals the robot to remove the blade, place it at a discharge point and then to go back for another blade to be loaded into the furnace at the commencement of another complete cycle.

This cycle gave the robot some spare time, so the company fixed up a punch press making a completely unrelated part and had the robot load it from a nearby magazine, and, after the press had operated, unload it into a tote box. The introduction of the robot in this factory increased production, simplified quality control, and reduced both indirect labor and overhead costs.

At a Canadian plant operated by International Harvester, harrow discs were heats treated to toughen the part so that it could better resist breakage when it struck a rock in use.

This process was very unpopular among workers in the factory; it was hot and the parts' were heavy. A decision was made to save three men per shift by introducing three robots as star performers in a well-planned layout. As the system now operates, robot no. 1 stands at the entry conveyor system for the furnace. Its hand is equipped with a vacuum cup, which comes down vertically from the robot arm. The hand lifts the top disc from a palletized stack of about 50 discs, and then transfers the disc to the conveyor, which carries it into the furnace operating at 1650 degrees Fahrenheit. As an interesting and effective detail, if the robot hand finds no disc in the stack, the vacuum cup rest on the framework of the pallet and a pneumatic pressure sensor terminates the program while a new stack of discs is advanced into position. When this is done a signal restarts the program.

Robots nos. 2 and 3 along the line are each equipped with a two-fingered hand, which grasps the disc on its outside diameter. Since the factory makes discs of five different sizes, it is interesting to note that only the fingers have to be changed when a different diameter disc is scheduled.

The ability to switch programs rapidly on the robots is another advantage when the size of discs is changed. A run of any one-disc size lasts an average of eight shifts, but on the other hand it may go on for a week. However the robots can change programs in a matter of minutes. Only three-axis robots are needed to perform this relatively simple sequence of movements. Figure 6.7 shows the production line. As each disc emerges from the furnace it continues on the conveyor to a 'pop-up' station which positions the disc for pick-up by robot no. 2. A gating system holds subsequent discs until the robot has picked up the first one and the pop-up station has dropped back. The robot then moves through about 160 degrees and loads the disc into the die of a press. Before the robot hand enters the die, three key conditions must be met:

1 The die is open, indicated by a ram limit-switch.
2 Robot no. 3 has signalled the removal of a previous disc (now dished through the Press action).
3 An infrared scanner indicates that the disc is within the required temperature range (450 to 600 degrees Fahrenheit) to be properly formed by the press.

If the disc shows itself to be too cool, robot no. 2 switches to a reject program. A too-high reading indicates that two discs have stuck together and that they are rejected also. Robot no. 3 unloads the press and swings some 180 degrees to place the disc on to the entry conveyor of a washing and drying unit.

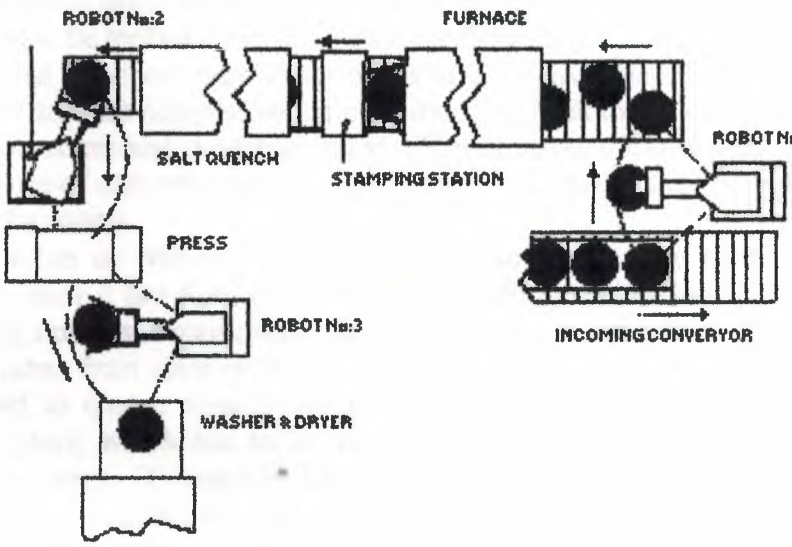


Figure 6.7 Plant layout for robotized heat treatment line

This is necessary because during its journey through the furnace the discs encounter solid quench in the form of salt, so they emerge heavily contaminated with this compound and it must be cleaned off before the discs can go into service. In this particular application, however, the salt build-up on both the press dies and the robot's fingers is minimized by the use of compressed air jets raised to furnace temperature and which play on the vital parts.

An analysis of the effectiveness of this robot installation is worthy of consideration. The plant's maintenance superintendent made no claims for a higher production rate when the robots moved in. About 300 discs an hour was the rate achieved and this was no more than could be expected from human operators. What was significant was the fact that previously there had been days when production could be maintained for no more than five hours. With the increased reliability of the new line and the indefatigability of the robots, production now can go around the clock.

The decision to install robots resulted in improved product quality, increased volume, savings in product cost and less labor costs from a reduction in workforce of between six and nine people according to the number of shifts worked. His assessment was not all starry eyed however. All-in maintenance costs with the robots in the system were higher than during manual operation, but considering aggregate savings, this was not incapacitating.

6.4 APPLICATION IN GLASS MANUFACTURE

Robots are usually at a disadvantage when compared with human beings because they lack dexterity and the freedom of movement provided by the human arms, hands and fingers. In handling glass, however, the robot can be provided with tooling, which in some respects makes it superior to the human operator. When a man picks up a large sheet of glass, he must hold it around the edges, protecting his hands from the sharpness, the stretch of his arms limiting the width of the sheet, which he can carry single-handed. The robot does not use this technique at all, but can be equipped with vacuum cups, which press against the surface of the sheet to adhere to it so tightly that the sheet can now be picked up and moved around. Releasing the vacuum reduces the Adhesion so that the sheet may be set down in a required place. Variations on this form of gripper can provide two sets of cups mounted back to back so that two pieces may be held simultaneously and dropped at different points if this is what the program calls for. The use of such vacuum cups has opened up a whole range of glass handling Applications for robots.

The robot has no role as yet in the float process but only finds applications handling glass after it has been produced. The handling application can be extended further. Setting up a glass manufacturing plant, or a line manufacturing products such as television tubes from glass stock, requires that a wide range of operating variables be harmonized to ensure trouble-free production runs. This is due in part to the properties of glass, which has to be worked at the right temperature in the most suitable environment. To achieve these optimum operating conditions may take a lengthy set-up time, requiring a special crew to come on duty several hours before production personnel, so that everything is working properly when the shift commences. By the same token, if the line should have to be shut down for any reason at all, the outcome could be very costly. It is not uncommon for the line to be left running at all times during the shift. If there should be any equipment failures, the glass manufactured may be allowed to smash on the shop floor rather than stop a process which might take hours to re-establish. So, should a robot fail while working on such a line, the cost of such a failure could be a very serious matter, especially if a human operator was not able to stand in and carry on the robot's job until it was repaired. This is a very important consideration in deciding how to use robots in this industry.

An example of a robot handling flat sheet glass is that of edge-grinding windows for automobiles. Robots are not good at cutting glass. Although they have no difficulty in scribing an appropriate line holding a glasscutter, the sharp tap given by an experienced glass worker who causes the sheet to break around the line requires judgment, which the robot does not possess. Further the second tap, necessary when the operator notices that the glass has not broken cleanly, is beyond the ability of the robot at this stage of its development.

Fortunately, however, the application of edge-grinding of automotive windows is one admirably suited to the robot; it is an application combining robotics and specialized automation, the latter being justified by the very large production runs typical of the automotive industry. The layout of the equipment for this operation is shown in Figure 6.8 the robot serves two edge-grinders. The glass sheets are picked up from an input conveyor using the vacuum cup technique. Each edge-grinder is loaded in sequence, the relatively lengthy grinding operation allowing the robot plenty of time to go through its programmed steps. Using the double-gripper system previously mentioned, the robot deals with two pieces of glass at a time, one a raw piece to be loaded into the grinder which is ready to receive it and the other a finished piece to be deposited on the output stacker.

The way in which the robot and specialized automation support one another in this role is illustrated by the fact that on arrival at the grinder an auxiliary unload-assist device comes into operation, which raises the finished part to allow the robot's

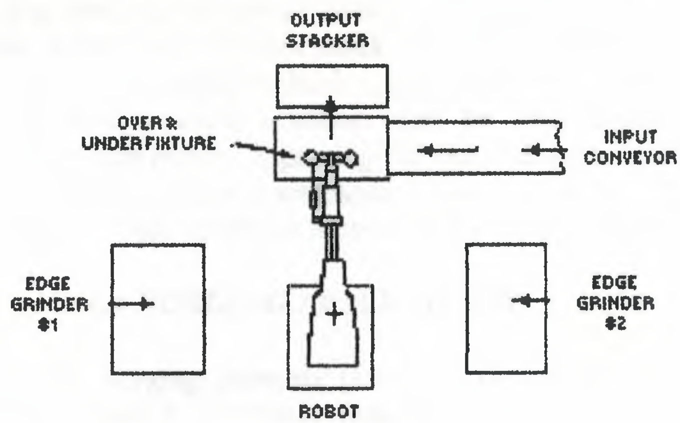


Figure 6.8 Layout for robot in window edge-grinding operation

hand to move underneath the part into a nest on the grinder. Versatility is increased by providing the robot with independent programming of vacuum and/or blow-off on either set of vacuum cups. The hand is supported in such a way that the angular position of the vacuum cups can be varied plus or minus 20%, and the position of the cups relative to the wrist is adjustable. These features permit a wide range of part shapes to be handled. The robot can operate one or both of the grinders as required simply by selecting the appropriate program.

The continuous nature of the glass industry makes this application an exceedingly profitable investment since each robot replaces two workers per shift on a three-shift basis. The application is suited to windshields, side and back windows and any other flat or semi-flat glass shapes. In this application it is usual for the tool to be moved around the glass rather than have the robot manipulate the tool, this being made possible by the large production which justifies the use of special- purpose automation in conjunction with the robot. Figure 6.9 illustrates the special double hand used in this application.

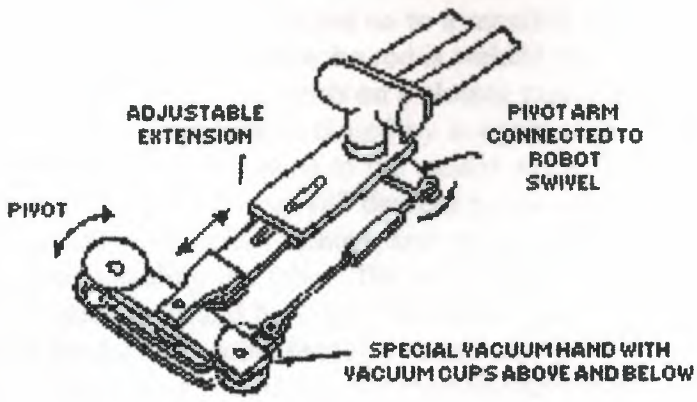


Figure 6.9 Special double hand used in glass handling

Glass in large flat sheets is conventionally transported in containers known as `bucks'. The buck is a steel carrier inside which is a wooden crate containing the glass. This technique is used both for raw stock glass and for shaped parts such as windshields. The stack of glass inside the buck may be as thick as two feet, and due to its density, the weight of the whole container may be considerable. The buck is designed so that its entire face opens, displaying the stack of glass. Robots have been used successfully in unloading these containers. Using its vacuum cups, the robot flexes the top glass sheet slightly to reduce friction as it slides out the sheet.

6.5 FORGING APPLICATION

Despite the fact that forging presents problems for robots, they are already working in this industry and it is certain that their numbers will grow and the complexity of their operations will continue to increase. The robot can pick up hot metal parts and can work for long periods in hostile environments. These characteristics have been put to good use in a company, which forges high-pressure gas cylinders on a production basis. It freed two operators to be assigned to more acceptable jobs and contributed to a labor saving which should result in a payback of the robot's cost on the basis of labor-savings alone.

In this application the human operators had to use a hoist to transfer hot metal performs from a 1500-ton forge press to a 260 by 100-ton draw bench where each perform, weighing between 60 and 360 pounds, was drawn into cylindrical shape of required size. Now, a robot receives a signal from the forge press to tell it that a preform is at the required temperature. A piston pushes up the preform so that the robot can grasp it. A typical preform weighs 143 pounds, and the robot picks it up, swivels through 180 degrees, places it on the draw table, withdraws, swivels back, immerses its gripper in a tank of cooling water and then waits by the forge press for the next signal. In this application, the preform stays in the forge press for 45 seconds, while the robot cycle takes only 10 seconds. There is about 35 seconds of wasted time as a result, but even with this disadvantage, this is one application of a robot in forging which is making economic sense.

In another application, which is only marginally a forging one, the robot again uses its ability to handle hot metals. In the manufacture of a chain, a rod is heated to about 1600 degrees Fahrenheit and ejected on to a mandrel where a robot picks it up and takes it to a welding position. When the rod is welded into a closed ring, the robot takes it to a point where an open ring rests on a closing machine. The robot places the ring over the open link, which then is closed by a signal instructing the machine to operate. The robot then returns the parts to the welder where the open ring is welded. The links are, in addition, turned over 180 degrees by the robot to permit welding on all sides. The robot continues in this fashion, and as the length of chain builds up it is supported to avoid overloading the robot. The robot deals with links of 3 to 5 inches inside diameter. Figures 6.10 and 6.11 illustrate this application in plant layout and the special gripper needed. The robot depicted in this case is a Unimate 2000B.

In yet another application, an automobile manufacturer is using a robot in a high production, hot forging operation - in die forging in fact. The robot picks up a slug of hot metal - at 1800 degrees Fahrenheit - and positions it in the forging die. It receives two blows, then the robot transfers the part to its next location. An operator is still used to monitor the critical heating process, but no longer has to place his hands in or near the die of the forging press, a potentially very dangerous place. This installation turns out 12,000 parts a day on a 24-hour basis.

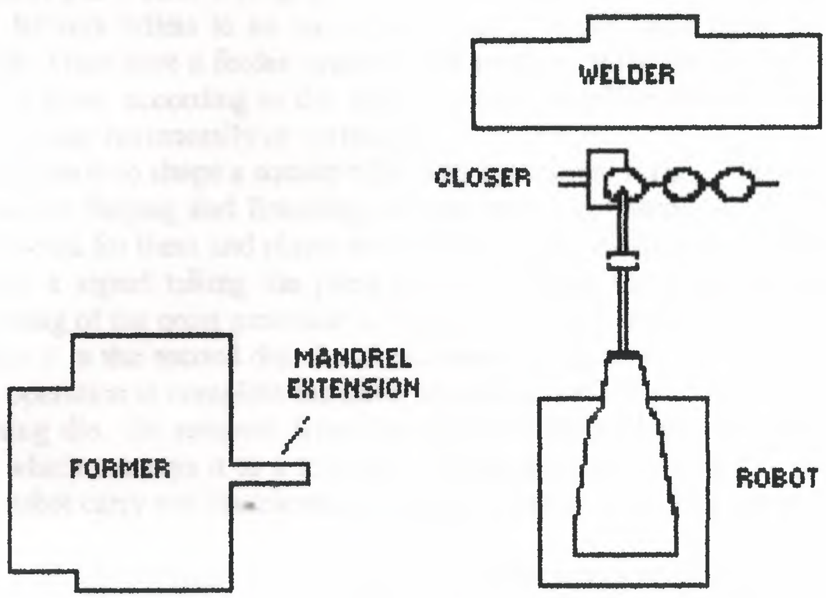


Figure 6.10 Plant layout for robotized chain link manufacture

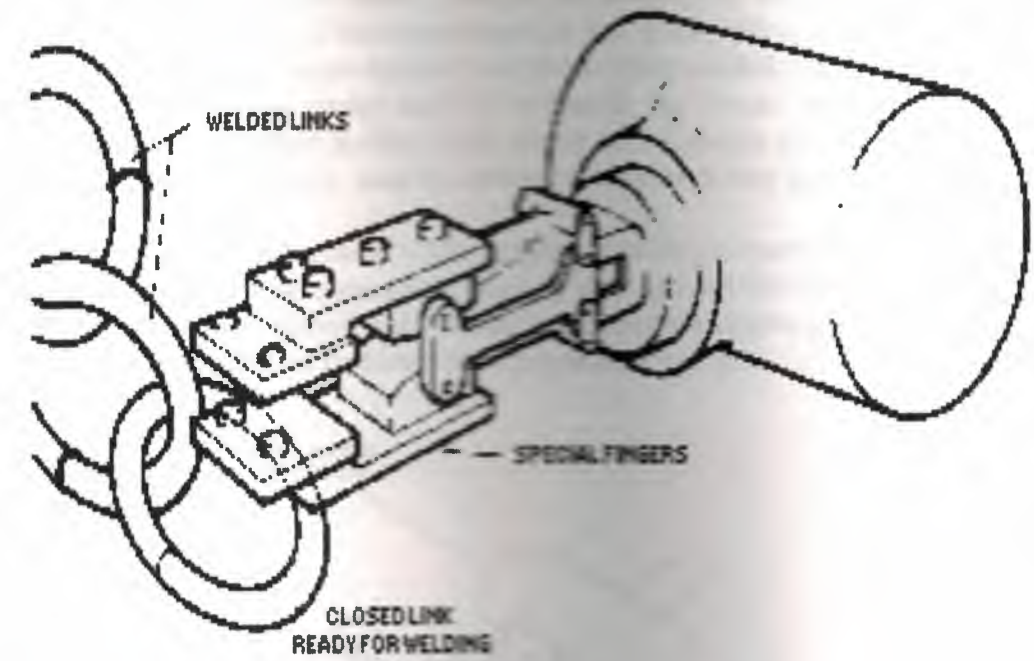


Figure 6.11 Special gripper for chain link manufacture

A more comprehensive use of the robot in a similar application is to be found in an Italian automobile plant. Here a forging press is used rather than a drop hammer. A vibratory feeder delivers billets to an induction furnace, which heats them to 1200 degrees Centigrade. From here a feeder system presents them to the robot in a simple yet effective coded form; according to the way in which the billets are to be forged, they are presented either horizontally or vertically.

The first operation is to shape a square billet into cylindrical form in three stages, blocking, intermediate forging and finishing, and all press operations. As the billets emerge, the robot waits for them and places them vertically in the first die, withdraws, and then generates a signal telling the press to close. When the first operation is complete, the opening of the press generates a signal, which instructs the robot to take the billet and place it in the second die. As it withdraws, it again signals the press to close. When this operation is complete the same sequence is employed to transfer the part to the finishing die. On removal from this die, the robot places the part on a discharge chute, which conveys it to a trimmer. Future development of this process will be to have a robot carry out the trimming operation and finally place the part in a container.

In another forging shop a robot is carrying out a similar series of operations but in a drop hammer machine rather than a press. The robot positions the part in appropriate dies, and waits while the programmed number of blows have been struck. The difference in this case is that the furnace used has been specially developed for this type of fast operation and contains a magazine of billets sufficient for one hour's operation. Another innovation is the inclusion of a photocell device in the transfer line to sense the arrival of a hot billet and initiate a signal to start the forging sequence. Typical grippers for this application are shown in Figures 6.12.

To sum up, robots are good at handling hot metal parts, are capable of accurately positioning those parts and of responding to signals from other equipment used in a multi-machine installation. They are unaffected, to a large degree, by hot, dirty environments and polluted atmospheres, though not entirely so and special design characteristics may be required if the environment is too hostile. What the robot lacks real turn of speed such as a human operator can show when needed, but on the other hand the robot does not tire easily and far outpaces the human operator in its endurance. However, the robot is also blind, and is consequently not able to make visual judgments as are frequently used by operators in determining precisely how to forge a part.

The forging industry needs robots. Older, experienced forge operators are retiring and not being replaced. Techniques are improving and new metals being forged so that the industry is far from played out. Indeed it represents a challenge to the robot designer because of its technical problems and its potential market.

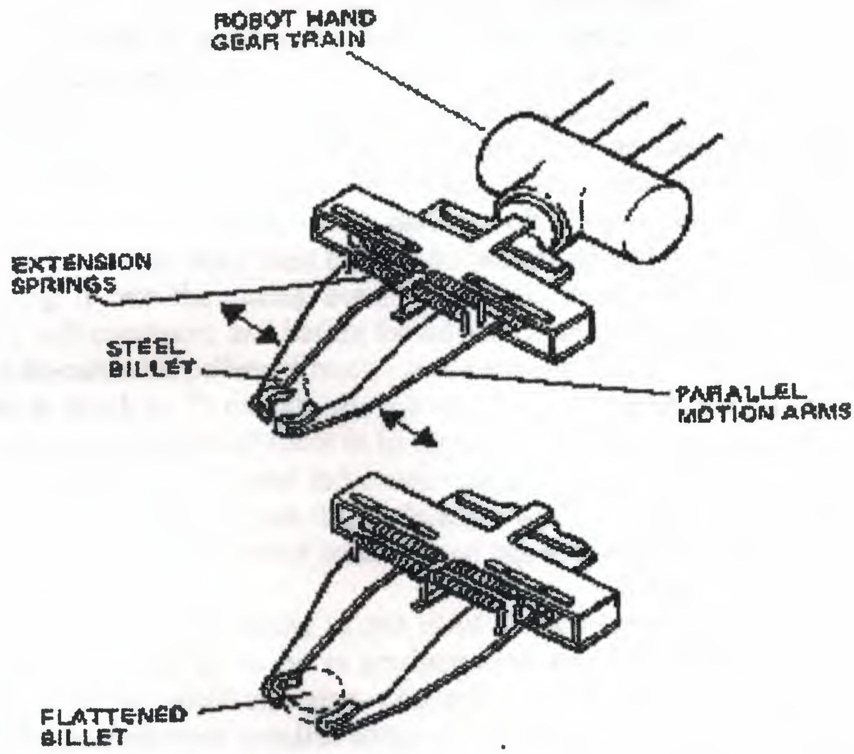


Figure 6.12 Special hand design for holding hot metal billets

6.6 APPLICATION IN BRICK MANUFACTURE

Some early attempts were made to use robots in the brickwork's, notably at the Burns Brick Company in Georgia, USA. It was hoped to reduce the manpower requirements to only three operators for each of the two shifts worked per day. The attempt was not simply to introduce robots, but also to use up-to-date transfer machinery wherever this proved feasible. Such machinery, it should be noted, is capable of working at rates almost twice as fast as those which human operators can attain. Hacking, however, resisted automation; it has not yet been possible to make a robot or any other form of automation work as fast and as reliably as a human operator.

The problems of the brickwork are in many ways related to palletization a field where the robot is undoubtedly able to play a role. It is of interest to describe briefly some of the work carried out at the Burns Brick Company, since it provides a valuable insight into the whole problem of robotization of this industry. Since a brick does not acquire its full strength until it has been fired, it is essential that any handling equipment should be capable of taking up the bricks without damaging them, and this applies particularly to the hacking process which occurs at the pre-firing stage.

In an attempt to meet this requirement, Burns used two Versatran robots located on platform straddling the kiln cars. The first robots were equipped with pneumatically operated `bladder' hands capable of picking up five bricks at a time. These were to be set down in eight courses each containing six groups of a five-brick pick-up, or 240 bricks in all. The production rate achieved by human operators however was a difficult one to match, and to date it has not been possible to use a robot to the exclusion of manual labor in the brickmaking process.

There is one sector of the brickmaking industry however, which is admirably suited to the robot. It is the manufacture of tar-bonded refractory bricks, which are used for furnace linings and similar high-temperature environmental requirements. A standard form of this brick as produced by Dresser Industries in the USA is made from a mixture of magnetism and hot pitch and can be as much as 30 inches in length and weigh 80 pounds. Such bricks are formed in a mechanical press, and as each brick is made, an automatic picker, which places it on a metal pallet, removes it from the press. The pallet and brick then have to be picked up and loaded on to a kiln-car for transporting it into the curing oven. The brick emerges from the mold in a hot, relatively soft condition, and before the advent of the robot, laborers were required to fill the kiln-cars with pallets of brick. Since the combined weight of the pallet and its load was as much as 70 pounds, the job was both strenuous and difficult. Spurred by the success of the industrial robot in handling and positioning comparable loads in the automotive industries, Dresser Industries sought to employ them for this step in their own process. It transpired that the application was no means as simple as it might at first sight appear, but the robot hand proved equal to the task and is now being used very successfully.

Introduced in 1975, robots in this plant service three presses. It is common for two of the three presses to be in production at any time the third being down for maintenance or for mold changing. One man monitors the entire operation whereas previously two men were required to service each press.

The pallet mentioned earlier is used in order to provide accurate registration for the robot pick-up. The automatic picker places the brick on to the pallet, and when three pallets are ready for transfer to the curing ovens the robot picks up all three simultaneously using specially-designed tong-like grippers. The pallets are slid on to shelves in the transport car. This is where the robot is on familiar ground for when one shelf in the car is filled, the next pick-up of three pallets is placed on the next shelf. The car moves along an indexed line, as it is fully loaded.

The best arrangement for the robotized line was found to be rather more complicated than that just outlined. The cars were divided into two halves, each of which had twelve shelves for holding six pallets at each level. Since the robot picks up three pallets at a time, it is required to go to each level twice. In the first operation, one half of the car is empty while the other half contains empty pallets. The robot picks up three filled pallets from an output conveyor and places them into the proper positions (as determined by the program) in the empty half of the car. It then goes to the other half of the car and picks up three empty pallets, which it then deposits on to an input conveyor. The robot then picks up a further three full pallets and loads them behind the first three already placed in the car. This fills the shelf, so the next load is deposited at the next (lower) level. This sequence continues until one half of the car is loaded with full pallets leaving the other half completely empty. The robot then proceeds to perform the second operation as follows. The car indexing system shuttles the half-full car and other car loaded with only empty pallets into position. The robot loads full pallets into the second half of the first car, while removing empty pallets from the first half of the second car. When the second operation is completed the robot programs back to the first operation again while the fully loaded car is indexed into position for removal. Figure 6.13 illustrates the plant layout for this sequence of operations. This apparently complex series of operations minimizes the number of motions required of the robot and optimizes production rates.

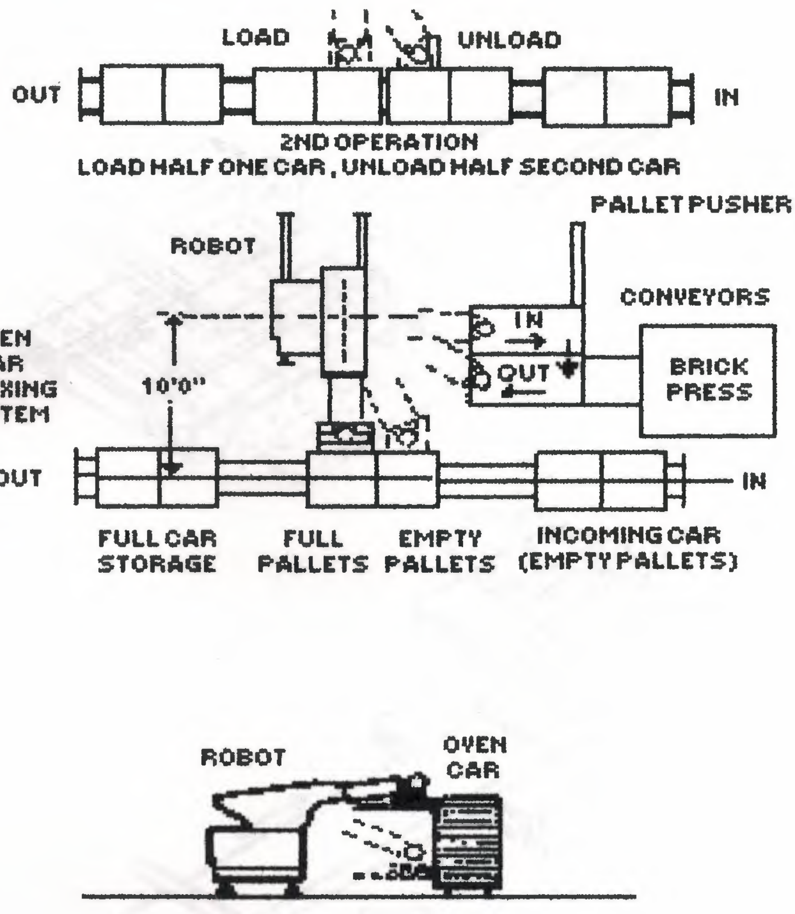


Figure 6.13 Plant layout for robotized pallet handling in brick manufacture

The tooling required consists of arms on which the pallets can rest. For placing pallets on input or output conveyors the pallets are deposited directly by the arms, but for loading into the shelves a pusher mechanism is required. The appropriate tooling is illustrated in 6.14.

A final note on this application; the tar-bonded refractory brick is cured rather than fired during its manufacture. The final firing occurs after it has been installed in a furnace - that is, it achieves its final strength only after it has been placed in service. This means that it is vulnerable to damage at all stages of manufacture and transportation, and thus the process is designed to minimize the handling of the product. The robot never grasps the brick, only the pallet on to which it is placed by the automatic picker.

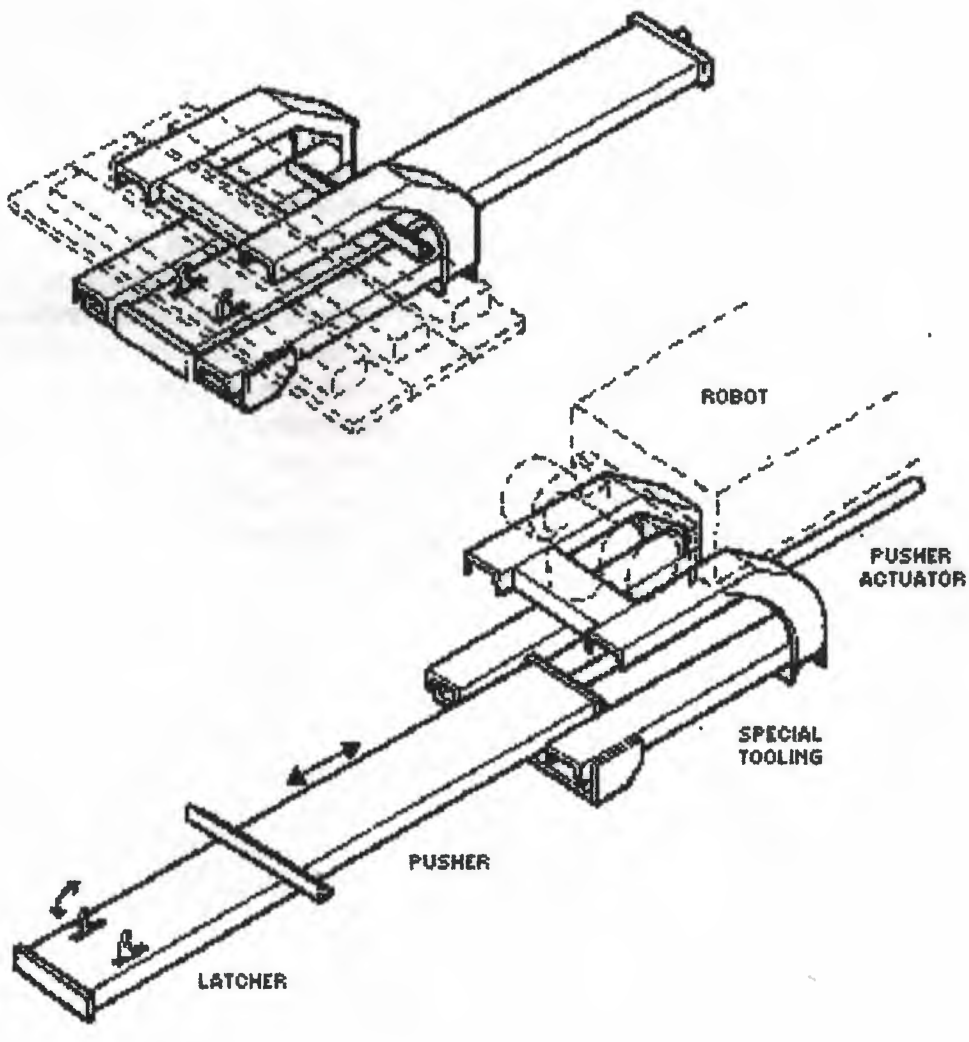


Figure 6.14 Pusher mechanism for robot arms for placing pallets

CONCLUSION

In reviewing robots offered by vendors today, one finds over 100 types of robots produced by around 50 manufacturers. The various designs differ mainly with respect to:

- Size and load-bearing capacity
- Kinematics and number of axes
- Control specifications

It is noteworthy that robot manufacturers offer their equipment with only one control and thus spare the purchaser the problem of having to make a choice. This is also not surprising when one considers the enormously high software development costs for an optimal adaptation of a CNC to a specific robot.

The final reason for choosing a robot to perform a limited range of tasks is that besides the time and cost factors discussed above the device can always handle other manufacturing tasks if necessary. The manufacturer recognizes that even though modification or complete change over of a process can not be anticipated at the time of purchase, the robot will be able to adapt to the new situation if the time ever arises when change is necessary or desirable. Although it does not cost any more to get the ability to change, it is the knowledge that it is there which is comforting. Of course this is one of the major advantages of flexible automation.

We have shown in this project that the use of robots in the manufacturing environment certainly seems to be justified from a qualitative point of view. However, in business, it is often bottom line dictates whether or not a certain policy will or will not be acceptable.

REFERENCES

1. Mark E. Rosheim (1994) Robot Evolution, Printed by John Wiley & Sons, Inc, USA.
2. Arthur J. Critclow (1985) Introduction to Robotics, Printed by Macmillan Publishing Company, Inc, USA.
3. Rcihard D. Kafter (1989) Robotic Engineering Integrated Approach, Printed by Prentice Hall, Inc, USA.
4. John J. Craig (1989) Introduction to Robotics, Printed by Addison Wesley Publishing Company, Inc, USA.
5. Joseph F. Engelberger (1980) Robotics in Practice, Printed by Joseph F. Engelberger, Great Britain.