NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

1988

GRADUATION PROJECT.

ROBOTIC SYSTEMS : ANALYSIS AND APPLICATIONS.

EE – 400.

NAMES: FARID UDDIN (950757) & BILAL AHMED (961026)

DEPARTMENT: ELECTRICAL & ELECTRONICS ENGINEERING

SUBMITTED TO: Prof. Dr. KHALIL ISMAILOV.

CONTENTS

BACK GROUND OF ROBOT

Robot	1
History	1
How Robots Work	2
Uses for Robots	2
Impact of Robots	3
Future Technologies	3

CHAPTER 1. INTRODUCTION TO ROBOTICS

1.1 CLASSIFSCATION OF ROBOTS	4
1.1.1Robotic-Like Devices	4
1.2 Classification by Coordinate System	5
1.2.1 Cylindrical coordinate robots	5
1.2.2 Spherical coordinate robots	6

1.2.3 Jointed arm robots 6

1.2.3a. .Pure Spherical. 6

1.2.3b..Parallelogram Jointed 6

1.2.3c3.Jointed Cylindrical 6

1.2.4 Cartesian coordinate robots 7

1.2.4a. Contilevered Cartesian. 7

1.2.4b. Gantry-Style Cartesian. 7

1.3 BASIC STRUCTURE OF ROBOTS 8

1.3.1 Manipulator 8

1.3.2 Sensory Devices 8

1.3.3 Controller 9

1.3.4 Power Conversion Unit 9

1.4 An Implementation of a Robot Controller 9

SUMMARY 10

CHAPTER 2. CONTROL OF ACTUATORS IN ROBOTIC MECHANISMS

2.1 INTRODUCTION 11

2.2 Closed-loop control in a position servo 11

2.2.1 No Velocity Feedback 11

2.2.2 Position Servo with Tach Feedback 12

2.3 THE EFFECT OF FRICTION AND GRAVITY 13

2.4 CONTROL OF A ROBOTIC JOINT 14

2.5 STEPPER MOTORS 14

2.5.1 Principles of Stepper Motor Operation 15

2.5.2 Half-Step-Mode Operation 17

2.5.3 Microstep Mode 17

2.5.4 Additional Methods of Damping Rotor Oscillations 18

2.5.5 Permanent-Magnet Stepper Motors 18

2.5.6 Stepper Motor Drives 18

2.6 LINEAR STEPPER MOTOR 20

2.7 BRUSHLESS DC MOTORS 21

2.8 DIRECT-DRIVE ACTUATOR 22

2.9 SERVO AMPLIFIERS 24

2.9.1 Linear Servo Amplifiers 24

2.9.2 Pulse-Width-Modulated Amplifiers 25

2.9.3 Effects of Feedback in Servo Amplifiers 27

2.9.4 Voltage amplifier driving a servomotor 27

2.9.5 Current amplifier driving a servomotor 28

SUMMARY 30

CHAPTER 3. ROBOTICS SENSORY DEVICES

3.1 NTRODUCTION 31

3.2 NONOPTICAL-POSITION SENSORS 31

3.2.1 Potentiometers 31

3.3 SYNCHRO SYSTEM 32

3.3.1 Resolvers 34

3.3.2 THE MOTORNETICS RESOLVER 35

3.4 THE INDUCTOSYN 36

3.5 LINEAR VARIBLE DIFFERENTIAL TRANSFORMERS 38

3.6 OPTICAL POSITION SENSORS 38

3.6.1 Opto-Interrupters 38

3.6.2 Optical Encoders 40

3.6.3 Rotary absolute encoders 40

3.6.4 Absolute encoders usually consist of three major elements 40

3.6.5 Optical incremental encoders 41

3.7 VELOCITY SENSORS 43

3.7.1 DC Tachometers 44

3.7.2 Velocity Measurement Using an Optical Encoder 45

3.7.3 Encoder and frequency-to-voltage converter 46

3.8 ACCELEROMETERS 47

3.9 PROXIMITY SENSORS 48

3.9.1 Contact Proximity Sensors 49

3.9.2 non contact proximity sensors 50

3.9.3 Reflected light sensors 50

3.9.4 Fiber optic scanning sensors 51

3.9.5 Scanning laser sensors 52

3.9.6 Ultrasonic sensors 52

3.9.7 Eddy-current sensors 53

3.9.8 Resistive sensing 54

3.10 TOUCH AND SLIP SENSORS 55

3.10.1 Tactile Sensors 55

3.10.2 Proximity rod tactile sensors 55

3.10.3 Photodetector Tactile Sensors 57

SUMMARY 58

CHAPTER 4. COMPUTER CONSIDERATIONS FOR ROBOTICS SYSTEM

4.1 INTRODUCTION 59

4.2 ROBOT PROGRAMMING 59

4.2.1 Robot Control Sequencing 60

4.2.2 Fixed instruction sequence control 61

4.2.3 Robotic extensions of general-purpose programming languages 62

4.2.4 Robot-specific programming languages 63

4.2.5 Languages Selected Summary of Robot 64

4.3 DEMONSTRATION OF POINTS IN SPACE 71

4.3.1 Continuous path (CP) 72

4.3.2Via points (VP) 72

4.3.3 Programmed points (PP) 73

4.3.4 Artificial Intelligence and Robot Programming 73

SUMMARY 74

CHAPTER 5. ROBOTIC APPLICATIONS:

5.1 INTRODUCTION	75
5.2 Current robotic applications	76
5.2.1 Welding	76
5.2.2 Spray painting	77
5.2.3 Grinding	77
5.2.4 Other applications involving a rotary tool	77
5.2.5 Assembly operations	78
SUMMARY	79

ACKNOWLEDGMENTS

It is a great pleasure for me to thanks first of all my parents who provided the support and motivation necessary to start and complete my studies.

In preparing this graduation project, I have been guided by the expertise of my teacher **Mr. Prof. Dr. KHALIL ISMAILOV** and it is also pleasure to acknowledge the enthusiastic support and assistance given to me by him in realizing the project.

I would like to acknowledge the following students and express my sincere appreciation for their helpful suggestions, criticisim and encouragement.

Sohail Raja

Muhammad Saeed Malik

Hyder Abbas Naqvi

Kashif Amjad

Junaid Zafar

Muhammad Ameen

Farukh Mansoor

Finally, I wish to thank my friend, Misbah Hussain Siddiqui, for his patience and extra care in typing this project. Needless to say without all the above help and support, the writing and production of this project would not have been possible.

BACKGROUND OF ROBOT

Robot

Computer-controlled machine that is programmed to move, manipulate objects, and accomplish work while interacting with its environment. Robots are able to perform repetitive tasks more quickly, cheaply, and accurately than humans. The term robot originates from the Czech word *robota,* meaning "compulsory labor." It was first used in the 1921 play *R.U.R.* (Rossum's Universal Robots) by the Czech novelist and playwright Karel Capek. The word robot has been used since to refer to a machine that performs work to assist people or work that humans find difficult or undesirable.

History

The concept of automated machines dates to antiquity with myths of mechanical beings brought to life. Automata, or manlike machines, also appeared in the clockwork figures of medieval churches, and 18th-century watchmakers were famous for their clever mechanical creatures.

Feedback (self-correcting) control mechanisms were used in some of the earliest robots and are still in use today. An example of feedback control is a watering trough that uses a float to sense the water level. When the water falls past a certain level, the float drops, opens a valve, and releases more water into the trough. As the water rises, so does the float. When the float reaches a certain height, the valve is closed and the water is shut off.

The first true feedback controller was the Watt governor, invented in 1788 by the Scottish engineer James Watt. This device featured two metal balls connected to the drive shaft of a steam engine and also coupled to a valve that regulated the flow of steam. As the engine speed increased, the balls swung out due to centrifugal force, closing the valve. The flow of steam to the engine was decreased, thus regulating the speed.

Feedback control, the development of specialized tools, and the division of work into smaller tasks that could be performed by either workers or machines were essential ingredients in the automation of factories in the 18th century. As technology improved, specialized machines were developed for tasks such as placing caps on bottles or pouring liquid rubber into tire molds. These machines, however, had none of the versatility of the human arm; they could not reach for objects and place them in a desired location.

The development of the multijointed artificial arm, or manipulator, led to the modern robot. A primitive arm that could be programmed to perform specific tasks was developed by the American inventor George Devol, Jr., in 1954. In 1975 the American mechanical engineer Victor Scheinman, while a graduate student at Stanford University in California, developed a truly flexible multipurpose manipulator known as the Programmable Universal Manipulation Arm (PUMA). PUMA was capable of moving an object and placing it with any orientation in a desired location within its reach. The basic multijointed concept of the PUMA is the template for most contemporary robots.

How Robots Work

The inspiration for the design of a robot manipulator is the human arm, but with some differences. For example, a robot arm can extend by telescoping—that is, by sliding cylindrical sections one over another to lengthen the arm. Robot arms also can be constructed so that they bend like an elephant trunk. Grippers, or end effectors, are designed to mimic the function and structure of the human hand. Many robots are equipped with special purpose grippers to grasp particular devices such as a rack of test tubes or an arc-welder.

The joints of a robotic arm are usually driven by electric motors. In most robots, the gripper is moved from one position to another, changing its orientation. A computer calculates the joint angles needed to move the gripper to the desired position in a process known as inverse kinematics.

Some multijointed arms are equipped with servo, or feedback, controllers that receive input from a computer. Each joint in the arm has a device to measure its angle and send that value to the controller. If the actual angle of the arm does not equal the computed angle for the desired position, the servo controller moves the joint until the arm's angle matches the computed angle. Controllers and associated computers also must process sensor information collected from cameras that locate objects to be grasped, or they must touch sensors on grippers that regulate the grasping force.

Any robot designed to move in an unstructured or unknown environment will require multiple sensors and controls, such as ultrasonic or infrared sensors, to avoid obstacles. Robots, such as the National Aeronautics and Space Administration (NASA) planetary rovers, require a multitude of sensors and powerful onboard computers to process the complex information that allows them mobility. This is particularly true for robots designed to work in close proximity with human beings, such as robots that assist persons with disabilities and robots that deliver meals in a hospital. Safety must be integral to the design of human service robots.

Uses for Robots

In 1995 about 700,000 robots were operating in the industrialized world. Over 500,000 were used in Japan, about 120,000 in Western Europe, and about 60,000 in the United States. Many robot applications are for tasks that are either dangerous or unpleasant for human beings. In medical laboratories, robots handle potentially hazardous materials, such as blood or urine samples. In other cases, robots are used in repetitive, monotonous tasks in which human performance might degrade over time. Robots can perform these repetitive, high-precision operations 24 hours a day without fatigue. A major user of robots is the automobile industry. General Motors Corporation uses approximately 16,000 robots for tasks such as spot welding, painting, machine loading, parts transfer, and assembly. Assembly is one of the fastest growing industrial applications of robotics. It requires higher precision than welding or painting and depends on low-cost sensor systems and powerful inexpensive computers. Robots are used in electronic assembly where they mount microchips on circuit boards.

Activities in environments that pose great danger to humans, such as locating sunken ships, prospecting for underwater mineral deposits, and active volcano exploration, are ideally suited to robots. Similarly, robots can explore distant planets. NASA's Galileo, an unpiloted space probe, traveled to Jupiter in 1996 and performed tasks such as determining the chemical content of the Jovian atmosphere.

Robots are being used to assist surgeons in installing artificial hips, and very high-precision robots can assist surgeons with delicate operations on the human eye. Research in telesurgery uses robots, under the remote control of expert surgeons, that may one day perform operations in distant battlefields.

Impact of Robots

Robotic manipulators create manufactured products that are of higher quality and lower cost. But robots can cause the loss of unskilled jobs, particularly on assembly lines in factories. New jobs are created in software and sensor development, in robot installation and maintenance, and in the conversion of old factories and the design of new ones. These new jobs, however, require higher levels of skill and training. Technologically oriented societies must face the task of retraining workers who lose jobs to automation, providing them with new skills so that they can be employable in the industries of the 21st century.

Future Technologies

Automated machines will increasingly assist humans in the manufacture of new products, the maintenance of the world's infrastructure, and the care of homes and businesses. Robots will be able to make new highways, construct steel frameworks of buildings, clean underground pipelines, and mow lawns. Prototypes of systems to perform all of these tasks already exist.

One important trend is the development of microelectromechanical systems, ranging in size from centimeters to millimeters. These tiny robots may be used to move through blood vessels to deliver medicine or clean arterial blockages. They also may work inside large machines to diagnose impending mechanical problems.

Perhaps the most dramatic changes in future robots will arise from their increasing ability to reason. The field of artificial intelligence is moving rapidly from university laboratories to practical application in industry, and machines are being developed that can perform cognitive tasks, such as strategic planning and learning from experience. Increasingly, diagnosis of failures in aircraft or satellites, the management of a battlefield, or the control of a large factory will be performed by intelligent computers.

CHAPTER 1. INTRODUCTION TO ROBOTICS

The study of robotic involves understanding a number of device objects. For example several engineering disciplines as well as those relating to physics economics. and sociology must be mastered before Inc can truly acquire more than a nodding acquaintance with the field.

1.1 CLASSIFSCATION OF ROBOTS

Based on the definition, it is apparent that a robot must be able to Operate automatically which implies that it most have some sort of programmable memory. In this section we follow the approach suggested by Engelberger to classify industrial robotic manipulators in two different ways one based one base on the *mechanical configuration of* the device and the other based on the general method used to *controls* its individual numbers (i.e.the (joints) or (axes). Before doing this however we wish to consider several devices that arc not truly robots bat often called by this name in the media.

1.1.1 Robotic-Like Devices

There are a number of devices that utilize certain facts of robot technology and are therefore often mistakenly called robots. In fact, Entelberger has referred to them as near relations. There are at least four such classes of mechanisms.

I Prostheses . These are often referred to as (robot arm) or (robot legs). Even through they can make use or either hydraulic or servo actuator, utilize servo control and have mechanical linkages, they does not have their own (brains) and are not truly programmable. The impetus to produce an action (called the command signal) in such a device originate in the brain of the human being. It then transmitted Via nerves to the appropriate appendage, where electrodes sense the nerve impulse. These are processed electronically by a special-purpose computer (on board the prosthesis), which in turn, controls the motion of the substitute limb (or hand).

2 Exoskeleton. These are a collection mechanical linkages that are made to surround either human limbs or the either human frame. They have the ability to amplify human's power. However, it is clear that they can not act independently and as such are robots. In fact, when an exoskeletal device is used the operator must exercise extreme caution, due to the increased force and/or speed that are possible. An example or such a device is the General Electric Hardima, developed in the 1970, which utilized hydraulically actuated servos

3.Telecherics. As mentioned previously these devices permit manipulation or movement of materials and/or tools that are located many feet away from an operator. Even though telecheric mechanisms use either hydraulic or servo motor actuators which are usually controlled in a in a closed loop manner, they are not robots because they require a human being to close the entire loop and to make the appropriate decisions about position and speed. Such devices are especially useful in dealing with hazardous substances waste. It has been proposed that they be used in undersea exploration. An example of an existing telecheric mechanism is the arm that is installed on the NASA space Shuttle (mistakenly referred to by tile press as a robotic arm).

4 Locomotive Mechnism. These are devices that imitate human heing or animals by having the ability to walk on two or fear leg. Although the multiple appendages can be highly sophisticated collections of linkages that are hydraulically or electrically actuated under closed-loop control, a human operator is still required to execute the locomotive process (i.e. make decisions concerning the desired direction of the device and to coordinate limb motion to achieve this goal). An artists rendering of the previously mentioned and ill-fated General Electric four legged vehicle. Having described what is not a robot, we now devote, the remainder of this section to classifying the various types of robotic devices. As mentioned above the approach. Classification will be performed in two different ways, based on:

- The particular coordinate system utilized it designing the mechanical structure
- The method of controlling the various robot axes.

We consider the coordinate system approach first.

1.2 Classification by Coordinate System

Although the mechanics or a robotic manipulator can vary considerably all robots must be able to move a part (or another type of "load") to some point in space. The major axes of the device, normally consisting of the two or three joints or degrees of freedom that are the most mechanically robust (and often located closest to the base) are used for this purpose. The majority of robots therefore, fall into one of four categories with respect to the coordinate system employed in the designed of these axes. That is they be described as being either cylindrical, spherical, jointed, or cartesian devices. Each of these categories is discussed briefly.

1.2.1 Cylindrical coordinate robots

When a horizontal arm is mounted on a vertical column and this column is then mounted on a rotating base, the configuration is referred to as a cylindrical coordinate robot. That is shown in figure 1-1. The arm has ability to move in and out (in the r direction) the carriage can move up and down on the column (in the z direction) and the arm carriage assembly can rotate as a unit on the base (in the θ direction). Usually, a full 360° rotation in θ is not permitted, due to restrictions imposed by hydraulic, electrical, or pneumatic connections or line. Also there is minimum, as well as a maximum extension (i.e. R) due to mechanical requirement.

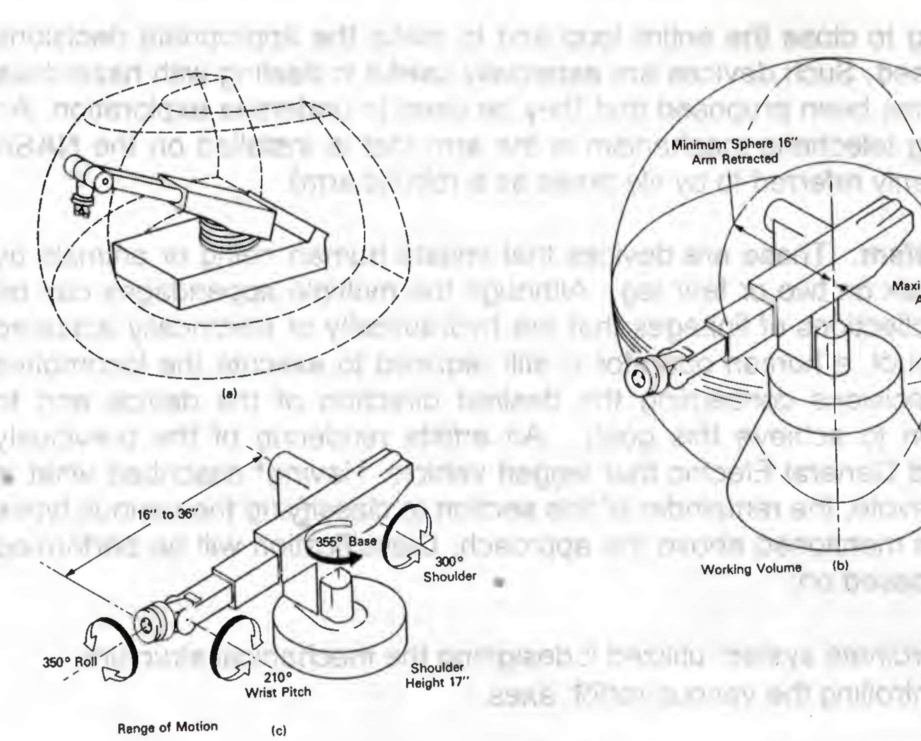


Figure 1–2 A spherical coordinate robot: (a) general view of the geometry of the robot's major axes; (b) working volume (workspace) of such a robot; (c) range of motion for each of the five axes of a typical spherical coordinate robot. (Courtesy of G Heatherston and U.S. Robots, a Square D Company.)

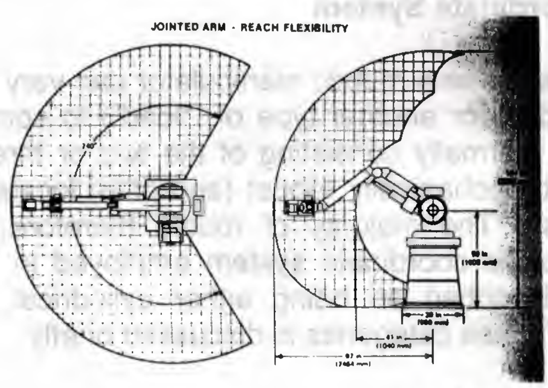


Figure 1–3 Geometry of a pure spherical jointed robot (Courtesy of J. Coshnitzke, Cincinnati Milacron, Cincinnati, OH.)

1.2.2 Spherical coordinate robots

What a robotic manipulator bears a resemblance to a tank turret, it is classified as a spherical coordinate device (see figure I-2). The reader should observe that the arm can move in and out (in the r direction) and is characterized as being a *telescoping boom* can pivot in a vertical plane (in the ϕ direction), and can rotate in a horizontal plane about the base (in the θ direction). Because of mechanical and/or actuator connection limitations the work envelope of such a robot is a portion of a sphere.

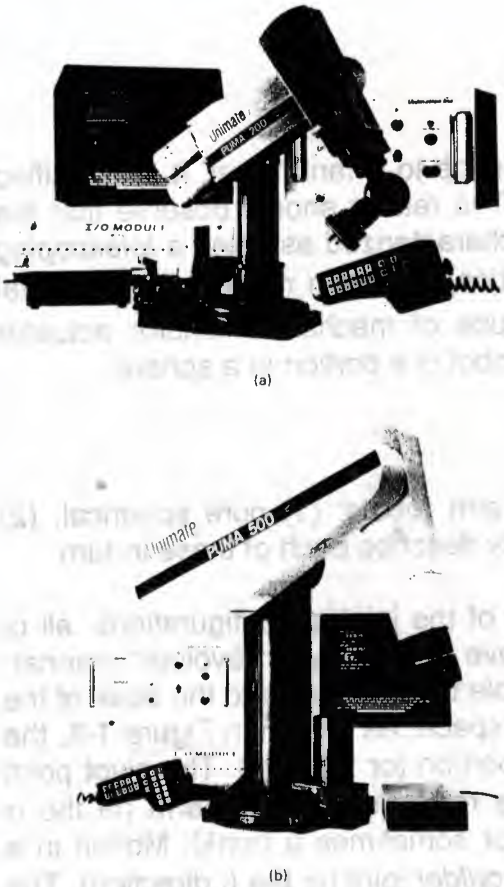
1.2.3 Jointed arm robots

There are actually three different types of jointed arm robots: (1) pure spherical, (2) parallelogram spherical, and (3) cylindrical. We briefly describe each of these in turn.

1.2.3a. Pure Spherical. In this, the most common of the jointed configurations, all of the links of the robot are pivoted and hence can move in a rotary or revolute" manner. The major advantage of this design is that it is possible to reach close to the base of the robot and over any obstacles that are within its workspace. As shown in Figure 1-3, the upper portion of the arm is connected to the lower portion (or forearm). The pivot point is often referred to as an "elbow" joint and permits rotation of the forearm (in the α direction). The upper arm is connected to a base (or sometimes a *trunk)*. Motion in a plane perpendicular to the base is possible at this *shoulder* joint (in the β direction). The base or trunk is also free to rotate, thereby permitting the entire assembly to move in a plane parallel to the base (in the γ direction). The work envelope of a robot having this arrangement is approximately spherical. Examples of commercial manipulators having this geometry are the Puma (Unimation), the Cincinnati Milacron T^3, and those made by ASEA, Niko, and GCA. Three different sizes of Pumas are shown in Figure 1-4.

1.2.3b. Parallelogram Jointed. Here the single rigid-member upper arm is replaced by a multiple closed-linkage arrangement in the form of a parallelogram (see Figure 1.3.10). The major advantage of this configuration is that it permits the joint actuators to be placed close to or on the base of, the robot itself. This means that they are not carried in or on the forearm or upper arm itself, so that the arm inertia and weight are considerably reduced. The result is a larger load capacity than is possible in a jointed spherical device for the same-size actuators. Another advantage of the configuration is that it produces a manipulator that is mechanically stiffer than most others. The major disadvantage of the parallelogram arrangement is that the robot has a limited workspace compared to a comparable jointed spherical robot. Examples of such commercial units are those manufactured by ASEA, Hitachi, Cincinnati Milacron, Yaskawa, and Toshiba

1.2.3c. Jointed Cylindrical. In this configuration, the single *r-axis* member in a pure cylindrical device is replaced by a multiple-linked open kinematics chain, as shown in Figure 1-5. Such robots tend to be precise and fast but will generally have a limited vertical *(z* direction) reach. Often the z-axis motion is controlled using simple (open-



igure 1-4 The Unimation PUMA is an example of a pure spherical jointed obot. Shown are 3 different sizes, the 200, 500, and 700 series. (Courtesy of nimation, Inc., a Westinghouse Company, Danbury, CT.)

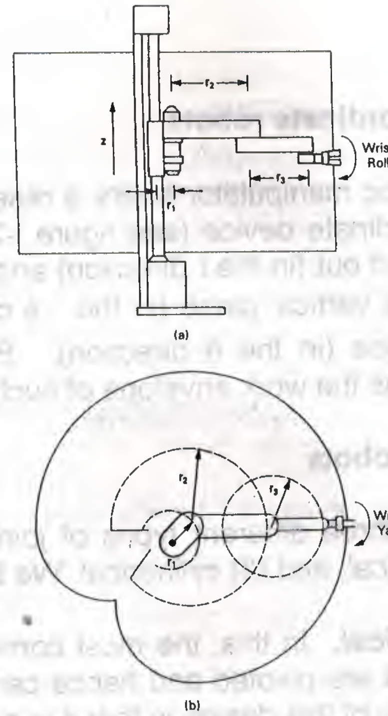


Figure 1-5 Jointed cylindrical workspace and geometry robot: (a) v tical cross-section; (b) top view. In some SCARA robots, $r_1 = 0$ and th axis is located at the wrist. Also, w could have a pitch axis.

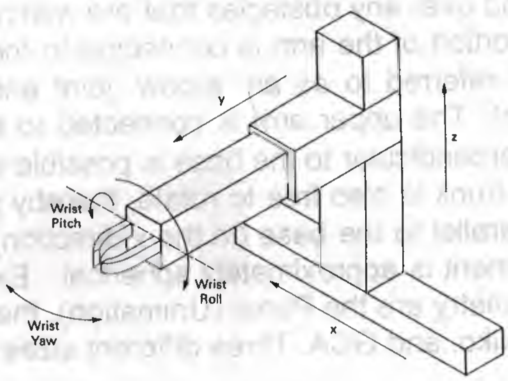


Figure 1-6 Cantilevered Cartesian robot geometry.

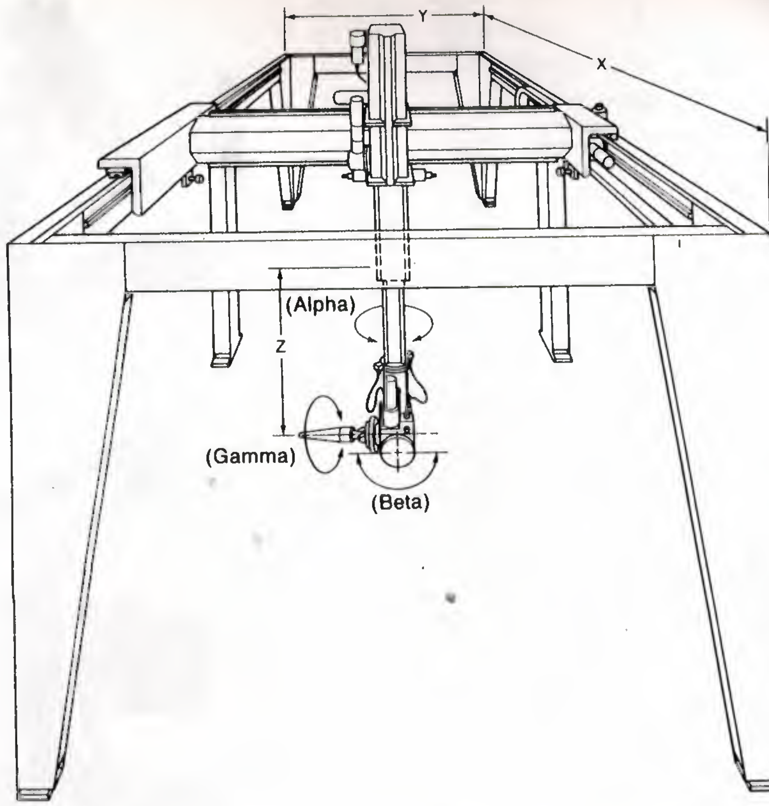


Figure 1-7 Geometry of a Cartesian gantry style robot. (Courtesy of CIM-CORP, Inc., St. Paul, MN.)

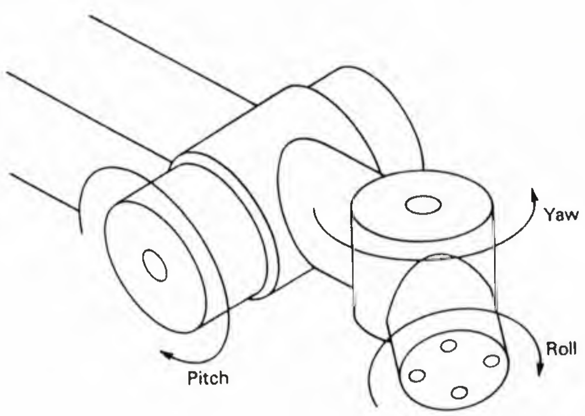


Figure 1-8 Example of a three degree-of-freedom wrist showing the roll, pitch, and yaw axes. These robotic joints are used for orienting objects in space.

loop) air cylinders or stepper motors, whereas the other axes make use of more elaborate electrical actuation (e.g., servomotors and feedback). Robots having this configuration are made by Harima, Reis, GCA, and United States Robots.
A subclass of the jointed cylindrical manipulator is the *selective compliance assembly robot arm* (or SCARA) type of robot [23]. Typically, these devices are relatively inexpensive and are used in applications that require rapid and smooth motions. One particularly attractive feature, selective compliance, is extremely useful in assembly operations requiring insertions of objects into holes (e.g., pegs or screws). Because of its construction, the SCARA is extremely stiff in the vertical direction but has some lateral "give" (i.e., compliance), thereby facilitating the insertion process.

1.2.4 Cartesian coordinate robots

In this the simplest or configurations the links, of the link of the manipulator are constrained to move in a linear manner. Axes of a robotic device that behave in this way are referred to as "prismatic." Let us now consider the two types of Cartesian devices.

1.2.4a. Contilevered Cartesian. As shown in Figure 1-6, the arm is connected to a trunk, which in turns attached to a base. It is seen that the number of the robot manipulated is constrained to move in the direction parallel to the Cartesian x, y and z-axes. Devices like these tend to have a limited extension from the support frame, are less rigid, but have a less restricted workspace than other robots In addition, they have good repeatability and accuracy (even better than the SCARA types) and are easier to program because of the "more natural" coordinate system. Certain types of motions may be more difficult to achieve with this configuration, due to the significant amount of computation required (e.g., straight line in a direction not parallel to any axis). In this respect, Control Automation did manufacture a robot that \vas capable of unrestricted straight-line paths.

1.2.4b. Gantry-Style Cartesian. Normally used when extremely heavy loads must be precisely moved, such robots are often mounted on the ceiling. They are generally more rigid but may provide less access to the workspace. In the last few years a number of smaller devices in this class have emerged. In this instance, a framed structure is used to support the robot, thereby making unnecessary to mount the device on the ceiling. The geometry of a gantry Cartesian device is shown in figure 1-7.

It is important to understand that the classifications above take into account only the major axes. However, a robot is not limited to only three degrees of freedom. Normally, a wrist is affixed to the end of the forearm. This appendage is itself capable of several additional motions. For example as shown in Figure 1-8. Axes that permit *roll* (i.e. motion in a plane perpendicular to the end of the arm), pitch (i.e.. motion in vertical plane passing through the arm), and yaw (i.e. motion in a horizontal plane that also passes through the arm) are possible. Moreover the entire base of the robot can be mounted on a device that permits motion in a plane (e.g. a x-y table or a track located in either the ceiling or floor).

1.3 BASIC STRUCTURE OF ROBOTS

1.3.1 Manipulator

The manipulator consists of a series of rigid number called *links* connected by *joints.* Motion of a particular joint causes subsequent links attached to it move. The motion of the joint is an accomplished by an actuator mechanism. The actuator can be connected directly to the next link or through some mechanical transmission (in order to produce a torque or speed advantage or “gain”). The manipulator end with a link on which tool can be mounted. The interface between the last link and the tool or end effector is called the *tool mounting plate or tool flange.*
The manipulator itself may be through of as being composed of three divisions:

- The major linkages
- The minor linkages (wrist components)
- The end effector (gripper or tool)

The major linkages are the Set or joint-link pair that grossly positions the manipulator in space. Usually they consist the first three sets (counting from the base of the robot). The minor linkages are those joints and links are associated with the fine positioning of the end effector. They provide the ability to orient the tool mounting plate and subsequently the end effect once the major linkage get it close to the desired position. The end effector, which is mounted on the tool plate, consists of the particular mechanism needed at the end of the robotic are to perform a particular task, the end effector may be a tool that does a function such as welding or drilling or it may be some type of gripper if the robots task is to pick up parts and transfer them to another location. A gripper may be a simple pneumatically controlled device which opens and closes or a more complex servo-controlled unit capable of exerting specified forces or measuring the part within its grasp (i.e. gaging).

1.3.2 Sensory Devices

For proper control of the manipulator we must know the state of each joint that is its position, velocity, and acceleration. To achieve this a sensory element must be incorporated into the joint-link pair. Sensory devices may monitor position, speed, acceleration or torque. Typically the sensor is connected to the actuator shaft. However, it could also be coupled to the output of the transmission (so that monitoring of each joints actual position with respect to the two surrounding links is possible).

Other types sensors may also be included in a robot system. Figure 1-9, shows a TV camera, which is part of a vision system. Sensor, along with its associated electronics and control, is used to locate a particular object in its field of view. Once found it relays the coordinates of the object to the robots controller so that the robot can position, its gripper over the object in order to pick it up. Not to be excluded are numerous other types of sensors such as those associated with touch (tactile sensors) and ranging (sonic or optical-type devices). These sensors can also be used by the

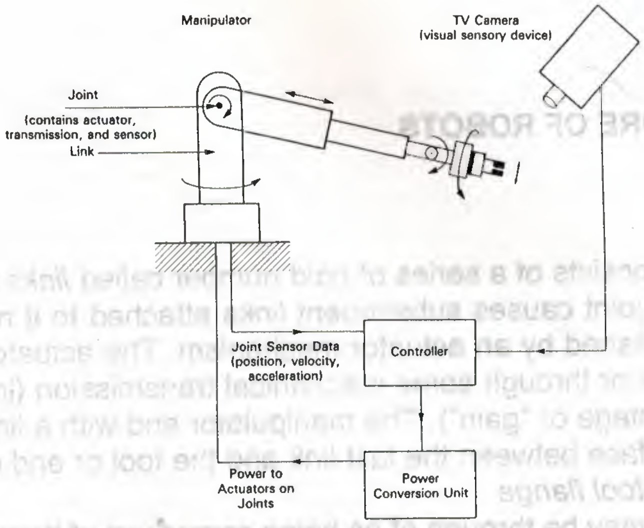


Figure 1-9 Components of a robot system.

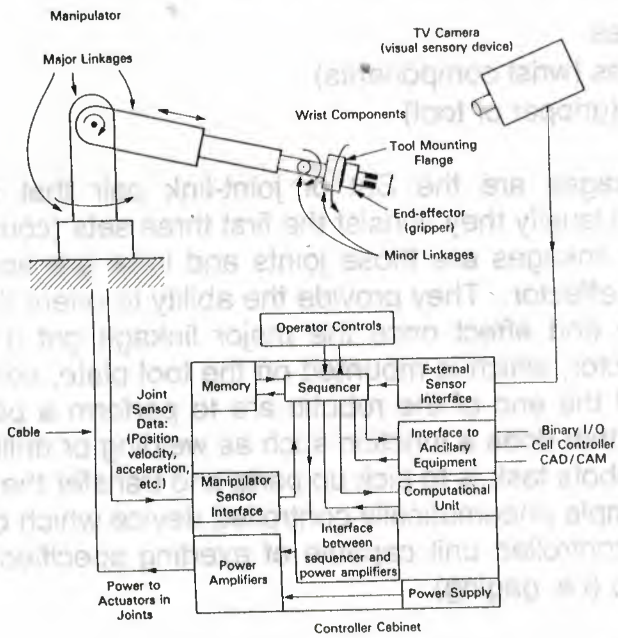


Figure 1-10 Subsystems of robot components.

robot system to gain information about itself or its environment.

1.3.3 Controller

The controller provides the intelligence to cause the manipulator to perform in the manner described by its trainer (i.e. the user). Essentially the controller consists of:

- A memory to store data defining the positions (i.e. such as the angle and lengths associated with the joints of where the arm is to move and other information related to the proper sequencing of the system (i.e. a program).

- A sequencer that interprets the data stored in memory and then utilizes the data to interface with the other components of the controller.
- A computational unit that provide the necessary computations to aid the sequencer.
- An interface to obtain the sensory data (such as the position of each joint information from the vision system) into the sequencer.
- An interface to, transfer sequencer information to the power conversion unit so that actuators can eventually cause the joints to move in the desired manner.
- An interface to ancillary equipment. The robot controller can be synchronized with other external units or control devices (e.g. motors and electrically activated valves) and/or determine the state of sensors such as unit witches located in these devices.
- Some sort or control unit or the trainer (or operator) to used in order to demonstrate positions or points, define the sequence of operations and control the robot. These can take on the form of a dedicated control panel with fixed function controls, a terminal and programming language and/or *teach pendent* or similar device containing *menu* driven instructions with which the operator can train the robot.

1.3.4 Power Conversion Unit

The power conversion unit contained the components necessary to take a signal from the sequencer (either digital or low-level analog) and convert it into a meaningful power lever so that the actuators can move. As an example this element would consist of electronic power amplifier and power supplies for electric robots, while in the case of hydraulic drives, it would consist of a compressor and control valves.

1.4 An Implementation of a Robot Controller

Figure 1-9, shows the details of the four major components of a robot system discussed above and their interconnections. Based on this figure we can propose a number of possible implementations for the robot controller. Figure 1-10, shows one such configuration. Here a single microprocessor is used as both the sequencer and the computational element. The common bus is the link that connects the microprocessor its memory the vision system, the binary I/O interface and the servo loops. By partitioning the system as shown, only the servo loops have to interface to the sensory data from the joints and provide drive signals to the power amplifiers. Also in this implementation the vision system is self-contained and incorporates all the necessary hardware and software to perform its function. By distributing the system, we have removed some of the burden from the sequencer.Figure 1-9.

The real-time clock is used to implement delays and to synchronize information transfer among the various devices connected to the bus. It may generate, interrupt so that the servo controllers always sample the joint positions and generate new set pints at the same instant thus ensuring a uniform sample rate.

From figures 1-9 and 1-10 we can infer another way to organize and describe the components of a robot system.

- Manipulator
- Connecting cable
- Operator cabinet
- Operator controls
- External sensors

As opposed to the functionality approach just described, tills organization is based on the physical packaging of the components and as a matter of fact most industrial robots are packaged this way. Clearly such a description is not as meaningful to the user in terms of the functionality of each subunit. However, it has the advantages of corresponding directly to the actual pieces of hardware.

SUMMARY

In this fairly detailed, nontechnical introduction, we have attempted to give the understanding of what an industrial robot is and what it is not, where it is applicable and where it is not, and finally, how such devices have evolved and how they may cause another industrial revolution to occur. In particular, introduced to most of the terminology associated with these devices and has been shown how to categorize them either by geometry of their major axes or by the type of control uitilized.

In this chapter we also presented a systems approach to the architecture and use of a robot. Additionally, the common specifications used to describe commertially available.

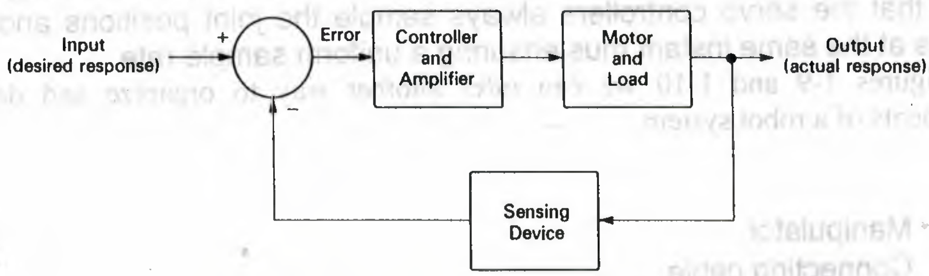


Figure 2-1 General closed-loop control system.

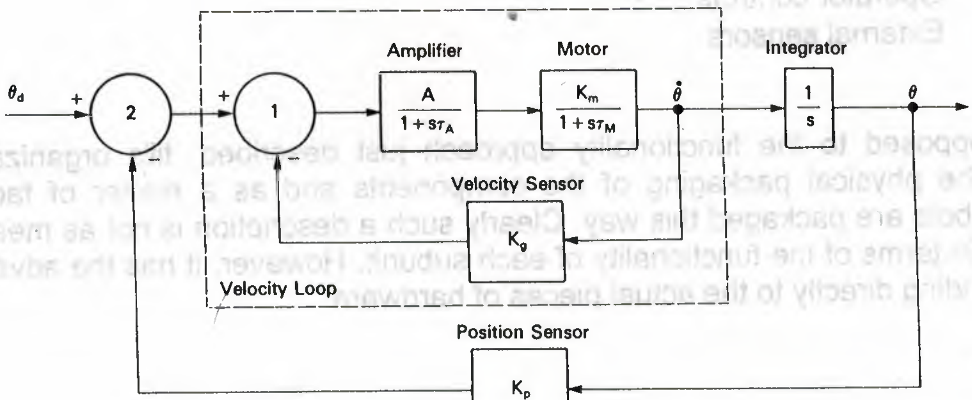


Figure 2-2 Typical analog position servo showing the velocity and position loops. The latter consists of the velocity loop, position sensor, integrator, and summing junction 2.

CHAPTER 2. CONTROL OF ACTUATORS IN ROBOTIC MECHANISMS

2.1 INTRODUCTION

In this chapter we present the practical aspects of control as they relate to robots, with the emphasis being placed on how robotics actuators are driven to achieve desired performance. It will be assumed that the reader has a basic knowledge or "classical" control theory and, hence topics such as Laplace transform and stability theory will not be developed. However these and other concepts will be used in discussing typical and reasonable control models that are applicable to robot system the material will be presented from the standpoint of a servo mechanism rather than from the more traditional theory of controls approach.

2.2 Closed-loop control in a position servo

The block diagram or a typical closed-loop control system is show, in Fig 2-1. Here some desired function or (position) command is the input and the response (or actual position) of the system or joint in the output A controller and an amplifier are used to drive a motor which then drives a load (e.g., the joint) Knowledge of how the joint is moving is provided by one or sensing device (e.g., an optical encoder or tachometer; see Section 4,s) and this information is used to produce an error signal which in turn, drives the controller/amplifier, and so on. For reasons that will become apparent shortly a typical position servo will actually used two sensing signal'; position and velocity let us now consider the specifics of the pure analog position servo shown in Figure 2-2.

In this diagram, θ_d and θ are the desired and actual joint positions with $\omega(t) = (\theta)$ being the angular velocity or the joint. Also, K_p and K_g are position and velocity gains with A and K_m the amplifier and motor gains respectively. Since angular position θ is related to the integral of the angular velocity $\omega(t)$ at integrator is shown in the diagram. It should be understood that this is for modelling purposes only since in practice θ is actually obtained through the use of a sensor.

With respect to figure 2-2, it is interesting to note that since angular velocity is fed back to summing junction 2, the position error can be viewed as a velocity command signal to the block marked "velocity loop". In fact, it is not uncommon to specify the shape of the velocity versus time curve (i.e., the velocity profile). The command position signal is then adjusted so as to produce this profile and drive the joint to the desired final position.

2.2.1 No Velocity Feedback

First consider the case where there is no "tach" or velocity feedback; that is, the velocity sensor (e.g., a tachometer) is removed so that $K_g = 0$ and hence the velocity loop in Figure 2-1 is open. Then the overall open-loop transfer function for the system in this figure becomes

$$GH\,(s) = \frac{AKpKm}{s(1+s\tau a)(1+S\tau m)} \qquad (2\text{-}1)$$

Typical values for the reciprocals of the motor and amplifier time constants τm and τa are 10 to 20 md/s and 6000 to 60,000 rad/s, respectively. Defining the open-loop gain constant to be

$$K = AKpKm \qquad (2\text{-}2)$$

Using standard root-locus techniques, it is found that the system will become unstable if $K > Kcross$ ($= 1/\tau a + 1/\tau m$) that is, the poles of the closed-loop system will move into the right-half portion of the s-plane and the response of the system will increase without bound ("blow up"). Also, for K less than *Kcross and* greater than KB (the gain where branches of the root locus leave the real axis), the poles of the closed-loop transfer function are complex conjugates (with negative real parts), so that the system exhibits underdamped performance. That is, a step command in θd will cause θ to respond. Usually, this type of behavior is undesirable, as it will not produce the fastest moves for a robot joint. That is, the *final* steady-state position will not be reached in the shortest time. Also, significant stresses on the mechanical components may be produced due to the rapid acceleration and deceleration required, as this final position is overshot (or undershot) and the servo is forced to make several corrections to bring the joint back to the desired point. To reduce or eliminate such response, it is necessary to provide the axis (joint) with some type of damping in order to reduce or eliminate the oscillations entirely. A certain amount of damping is inherent in the components themselves (e.g., motor and gear frictions) and in some instances may be sufficient to produce an acceptable response characteristic (i.e., either critically damped or just slightly underdamped). When this is not the case, however, another source of damping must be employed. This usually takes the form of viscous friction [i.e., a friction torque that is proportional to angular velocity *w(r)* and is obtained from "tach" or velocity feedback.

2.2.2 Position Servo with Tach Feedback

Now let us restore the tack feedback in Figure 2-2, that is, consider the ease where Kg is not zero. It will be demonstrated shortly that doing this will produce the desired damping in the position loop. Using standard block diagram simplification techniques, the open-loop transfer function for the joint with tach and position feedback is found to be

$$GH(s) = \frac{AKm(Kp+SKg)}{s(1+s\tau m)(1+s\tau a)} \qquad (2\text{-}3)$$

Note that velocity feedback causes a zero to be added to the overall open-loop transfer function (at $s = -Kp/Kg$) Using Eq. (2-2), the root locus for this system (plotted as a function of K). It is observed that in both cases, the system is now stable for all (positive) values of K, however, there is a dominant set of poles. Physically, this means that the pole due to the amplifier has little effect on the closed-loop response of the system. Under these conditions, the joint-position servo can be approximated by a *second-order* system, and the closed-loop transfer function becomes

$$T(s) = \frac{\theta(s)}{\theta d(s)} = \frac{AKm}{s\tau m + s(1 + AKgKm) + AKmKp} \qquad (2\text{-}4)$$

It can be shown that for a second-order system, a general form of the closed-loop transfer function is given by

$$T(s) = \frac{\omega n}{s + 2\xi\omega ns + \omega n} \qquad (2\text{-}5)$$

where ωn is the undamped natural radian frequency and ζ is the damping coefficient.* Comparing Eqs. (2-4) and (2-5), the damping coefficient for the joint with position and tach feedback is then found to be

$$\xi = \frac{0.5(1 + AKgKm}{\sqrt{AKmKp\tau m}} \qquad (2\text{-}6)$$

Several important conclusions can be drawn from this equation:

1. The more tach feedback (i.e., as Kg is in creased), the more damping there will be in the position servo, Thus the joint response will tend to become *less underdamped as Kg is increased, and vice versa.*
2. The more position feedback (i.e., as $K\sim$ is increased), the less damping there will be. Thus the joint response will tend to *become more underdamped as* is *increased, and vice versa.*

2.3 THE EFFECT OF FRICTION AND GRAVITY

As mentioned previously, one the major goals of robot is to move a tool or a part from one point to another in an accurate and repeatable manner. Anything that prevents this goal from being achieved it clearly undesirable and must, therefore, either be compensated for or eliminated. In practically or electromechanical, pneumatic-mechanical, or hydraulic-mechanical system, friction in various components will create a position error. Also, gravity will produce a position error of varying magnitude in one or more joints for most robots.

2.4 CONTROL OF A ROBOTIC JOINT

In the previous sections, it was assumed that the position and velocity information was "available" with no thought being given as to how one actually obtains this information. Also, it was assumed that all components in the position servo were analog devices. Although this is useful in the analysis or servo loops, it is not usually the case in practice. What is normally done in a robotic joint servo is to utilise either a digital approach where all sensory information is obtained and processed in a digital fashion, or else both analog and digital techniques are used to obtain and process information.

Regardless of the scheme employed. However, the command signal to the joint servo of a robot is invariably obtain from a microprocessor (i.e., the "master") and is, therefore, digital in nature. It is important to understand that this implies that the input to the joint is not a continuous-time function but is, instead, a "sampled" signal which is updated (i.e., changed) only periodically (e.g., every 25 ms) by the master or a special math coprocessor. Such an approach is taken because the master must send information to all the joint servos (e.g., six in a sixaxis robot). Consequently it must have enough time to complete the various computations required in the path-planning algorithm and then to communicate this information to the individual joints. We will call the update (also referred to as either a sample or set *point)* time *Ts*.

Although it is quite feasible to convert the digital position command into an analog signal by using a digital-to-analog converter, this is not often done. Instead, the individual joint processors themselves perform an interpolation between two consecutive set points output by the master. For example, if Ts = 25ms, the *interpolated* set points might occur every 3.33ms (Ts/8), implying that the master update interval is divided into eight subintervals. As a consequence, considerably smoother manipulator motion is produced.

In practice, it is possible to obtain both position and velocity data in either an analog or a digital fashion using the same or separate device (i.e., sensors) for monitoring these signals.

2.5 STEPPER MOTORS

It is possible to construct a motor in which the rotor is able to assume only discrete stationary angular positions. Rotary motion occurs in a stepwise manner from one of these equilibrium positions to the next, and as a consequence such a device is called a *stepper motor.* Although to date, the actuators have been used in only a few robots (e.g., the Merlin manufactured by the American Robot corporation), they have been employed in a variety of applications, the most notable of which is in the held of computer peripherals (e.g., printers, tape drives, capstan drives and memory access systems). Steppers have also been used in equipment related to the areas of process control, machine tools, and medicine.

There are several general characteristic of a steeper motor that have made it the

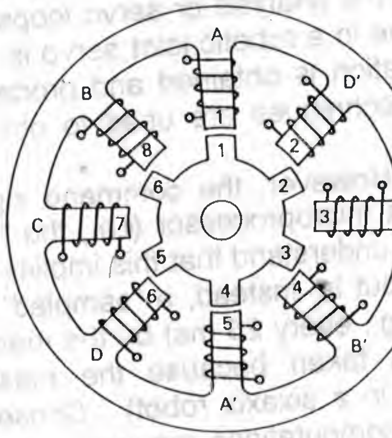


Figure 2-3 Basic structure of a variable reluctance-type stepper motor (Redrawn with permission of the Superior Electric Co., Bristol, CT.)

actuator of choice in such a large number of applications:
1. The device can be operated in an open loop manner with a positioning accuracy of ±1 step* (assuming that the rotor angular velocity is low enough so that no steps are lost during a move). Thus if a certain angular distance is specified, the motor can be commanded to rotate an appropriate number of steps, and the mechanical elements coupled to the shaft will move the required distance.
2. The motor exhibits high torque at small angular velocities. This is, of course, useful in accelerating a payload up to speed.
3. The motor exhibits a large holding torque with a dc excitation. Thus it has the property of being a "self-locking" device when the rotor is stationary. In fact, the rotor can move only when the terminal voltage *changes* with time.

In addition to these characteristics, there are other advantages that often make designers of various pieces or equipment select the stepper motor over the DC servomotor;

- The stepper is directly compatible with digital control techniques. Consequently, it can readily be interfaced with digital controllers and/or computers.
- It exhibits excellent positioning accuracy, and even more important, errors are *noncumulative*.
- Since open-loop control can be employed with the motor, it is often unnecessary to use a tachometer and/or an encoder. Thus cost is reduced considerably.
- Motor construction is simple and rugged. There are usually only two bearings and the motor generally has a long maintenance-free life. For this reason it is a cost-effective actuator.
- The stepper can be stalled without causing damage (due to overheating).

Several of these traits make the stepper motor potentially useful in certain types of robots. An obvious one would be the relatively low cost with respect to the dc servomotor.

2.5.1 Principles of Stepper Motor Operation

There are two basic varieties or stepper motors that can be constructed: (1) the variable reluctance (VR) types and (2) the permanent magnet (PM) types. Although the PM style is most often used in a broad range of applications today, the operation of the VR stepper is easier to understand and therefore we consider it operation in this section.

The structure of a typical VR stepper motor is shown in figure 2-3. It is observed that unlike the servomotor, both the stator and the rotor are *toothed* structures. Fundamental to the operation of this motor is that the rotor and stator does not have the same number of teeth. For example, the stator shown in Figure 2-3 has eight (located every 45°) and the rotor has six (located every 60°). In addition each stator tooth has a coil wound on it with oppositely placed coils (e.g., A and A') being grouped together and referred to as a *phase*. In this example it is seen that there are four phases (labelled *A* B C and P).

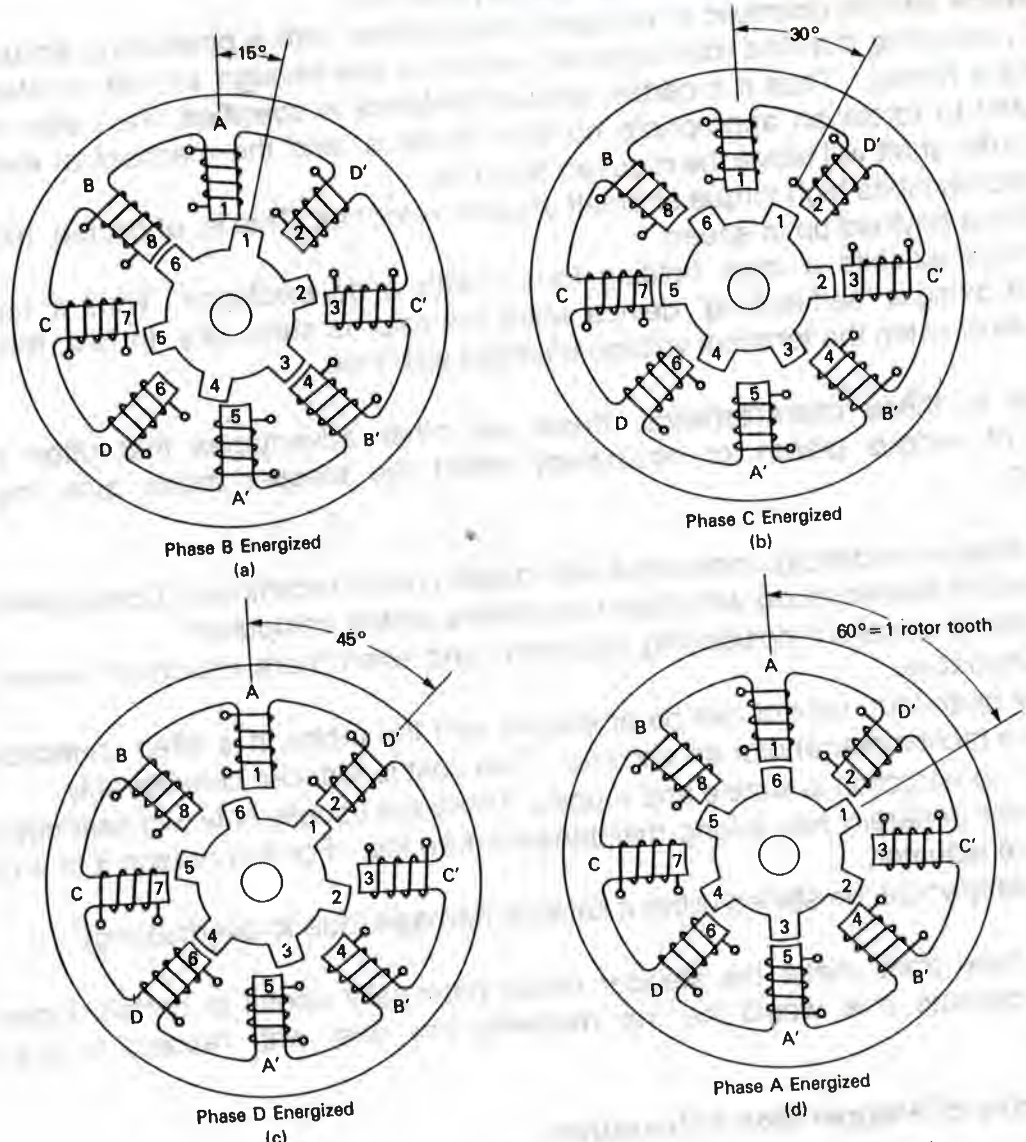


Figure 2-4 VR stepper motor excited in the sequence BCDA. The motor is assumed to start in the position shown in Fig. 4.6.1. (Redrawn with permission of the Superior Electric Co., Bristol, CT.)

The operation of the VR stepper is quite simple being based or the principle of minimum reluctance whereby a magnetic structure always attempts to reorient itself so as to minimize the length of any air gap in the magnetic path. One can think of the magnetic device moving so that the magnetic field can find the path of "least resistance". Thus for the example is shown in figure 2-3, when phase A is energized rotor teeth 1 and 4 (R4, 1) will align with stator teeth 1and 5 (S5, 1) and will remain in this position as long as the coils in the same phase are energized.

This is said to be a stable equilibrium pint and represents "one step" of the motor. It is important to understand that as long as the excitation remains on coils A-A' there is a *holding torque*, so that if any applied external torque is less than the value, no motion will occur. Also increasing the current through this phase will not cause the rotor to move but will, in actuality, increase this holding torque. Thus the motor will tend to "lock" more under increased current excitation. This should be compared to the servomotor, where increasing the excitation will tend to make the rotor turn faster.

Now suppose that the excitation is removed from phase A and placed on phase *B*. R6, 3 will align with S8, 4 as shown in Figure 2-4a. It can be seen that the rotor has moved clockwise through an angle of 15° (60 - 45 = 15). This process can be repeated for phases C, *D*, and then back to *A* (see Figures 2-4b, c and d). In each case, a 15° step occurs with a complete sequence of phase excitations (e.g., A, B, C, D, and A), producing a rotation of 60°. Consequently in this example, it requires six such cycle to cause one complete rotor revolution, and we would therefore describe this as a "24-step/revolution"' motor.

The step angle is related to both the number of stator teeth Ns and rotor teeth Nr. Specifically, it can be shown that

$$step-angle = 360° \frac{Ns-Nr}{NsNr} \longrightarrow \text{(2-3)}$$

and that the

$$number-step/rev = \frac{NrNs}{Ns-Nr} \longrightarrow \text{(2-4)}$$

Physically, rotors of all stepper motor exhibit an underdamped response in moving from one step to another. This can most easily be seen by recognizing that when the excitation is changed to an adjacent phase, the rotor travels toward the new equilibrium point. Although the accelerating torque is zero when the rotor and stator teeth are in alignment, the *angular velocity* or the rotor is not zero. As a result an overshoot of the equilibrium position occurs. There will not be a torque on the rotor that will accelerate it back toward the equilibrium point that was just passed (i.e., in the opposition direction). In fact, this process may actually be repeated several times before the rotor comes to rest. In general, the magnitude and duration of the damped oscillation is dependent on the step angle (i.e., the larger the angle, the larger the overshoot). In certain applications, such behaviour may not be acceptable (e.g., a robot) whereas in others it will be perfectly all right (e.g., a printer).

Stepper motors can be made with a wide range of steps/rev. From the standpoint of cost, a practical upper limit is 200 and produces a step angle of 1.8°. For many applications, this relatively small angle is quite adequate (e.g., the rotor oscillation will not interfere with the device operation). However, where finer step-angle resolution is required (note that this is one way to reduce the. angular overshoot problem described above), other techniques can be used.

2.5.2 Half-Step-Mode Operation

In this previous discussion, one phase was energized at a time and produced a step angle given by Eq. (2-3). This method or operation is called *full-step mode.* Now suppose that two adjacent phases (e.g., *A* and B) *are energized simultaneously.* It is logical that an equilibrium point is created somewhere between the two full-step points as determined by separately exciting phases A and B). In fact, if the electrical properties of the coils in the two phases are identical and if the same excitation amplitude is applied to both sets of coils, the new equilibrium point will be about halfway between the full-step points. This process can be repeated for phases *BC, CD,* and *DA* so that additional "halfway" equilibria can also be obtained. It should now be clear to that if the phase excitation sequence is *A, AB, B, BC, C, CD, DA, A* and so on, the rotor will make twice the number of clockwise moves as before (i.e., with respect to the full-step mode), and thus the stepper in Figure 2-3 will now have 48 discrete equilibrium points, per revolution. The name given to such an operation is not surprisingly, *"half-step mode".* Since the rotation angle per step has been cut (approximately) in half, the angular overshoot of the rotor in moving from point to point is reduced. Reversing the phase excitation sequence will cause the rotor to turn in the counterclockwise direction as before. The switching circuitry needed to produce half-step operation is somewhat more complicated and is therefore more costly then the relatively simple full-step electronics.

2.5.3 Microstep Mode

A little thought should convince that there is nothing "sacred" about exciting two adjacent phases actually (e.g., both with V volts). In fact, it is possible to use an excitation voltage anywhere between O and *V* in order to energize the second phase. In this case the stable point will occur at some location (but not halfway) between the tow full-step equilibria. This scheme produces a mode of operation generally referred to as microstepping. Most often, the microstep size is determine by dividing the angular distance of a full step by an integral power of 2 (e.g., 2, 4, 8, 16 or 32; this produces the smallest computational burden on the stepper motor controller). Although microstepping requires considerably more complex switching circuitry to implement so that the cost is quite a bit higher than that required for full-step operation its use generally produces smoother low-speed operation of the motor. In a robot application this is an important consideration since oscillation at the desired final point is usually unacceptable.

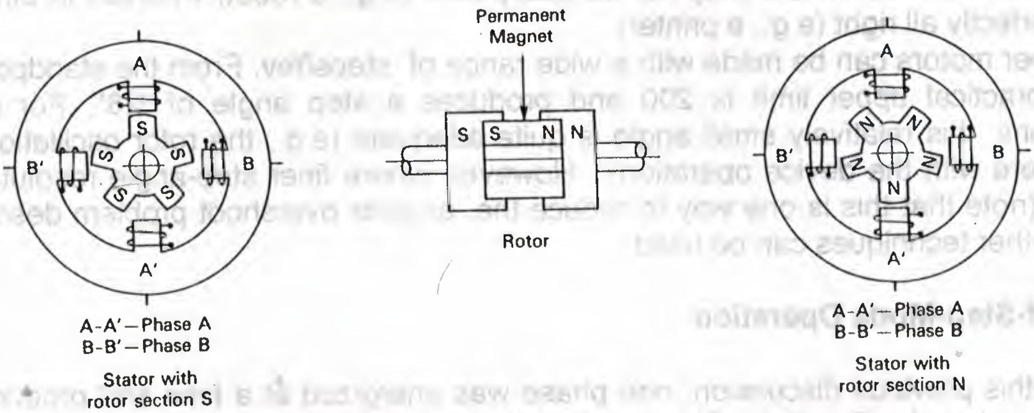


Figure 2-5 A simplified diagram of a permanent-magnet stepper motor. (Redrawn with permission of the Superior Electric Co., Bristol, CT.)

2.5.4 Additional Methods of Damping Rotor Oscillations

There are other ways to damp out rotor oscillation described in Section 2-3. For example, by adding viscous inertia (often called a Lanchester damper) or by using either a friction disk or eddy-current damper. Although these techniques achieve the desired goal, they also add inertia and may therefore adversely affect the transient response of the rotary system (this will not be true if the system inertia is already high).

An electronic technique exists that avoids the problem created by added inertia. Called *bang-bang damping* the idea is to accelerate the motor in the normal way, however before the rotor reaches it desired position, the phase excitation sequences reversed causing the rotor to decelerate more rapidly. If the phase reversal is timed correctly, the rotor can be made to come to rest at the equilibrium point with almost no overshoot. Clearly, the timing of the reversal is critical and it turns, out the switching instants is a function of system parameters (e.g., friction and load inertia). In a robot, where the inertia of any joint usually various significantly with position (and hence with time during any move) a very sophisticated scheme required to sense these changes and then modify the phase reversal times accordingly.

2.5.5 Permanent-Magnet Stepper Motors

As stated previously the PM stepper motor is the most commonly used type. It consists of a multiphase stator and a two-part permanent-magnet rotor. Just as with the VR stepper, both of these structures are also toothed (sec figure 2-5). The major difference in this case is that the opposite ends of the rotor are north and south poles of a permanent magnet with the teeth at these ends being offset by half a tooth pitch. It is worthy of mention that the PM stepper can be operated in full half or microstep mode. Table 2-1, indicates the major differences between the two classes of stoppers.

2.5.6 Stepper Motor Drives

As indicated in Table 2-1 the PM stepper rotor position is dependent on the polarity of the phase excitation. Consequently a *bipolar* signal is required to achieve bi-directional control. With respect to figure 2-5.

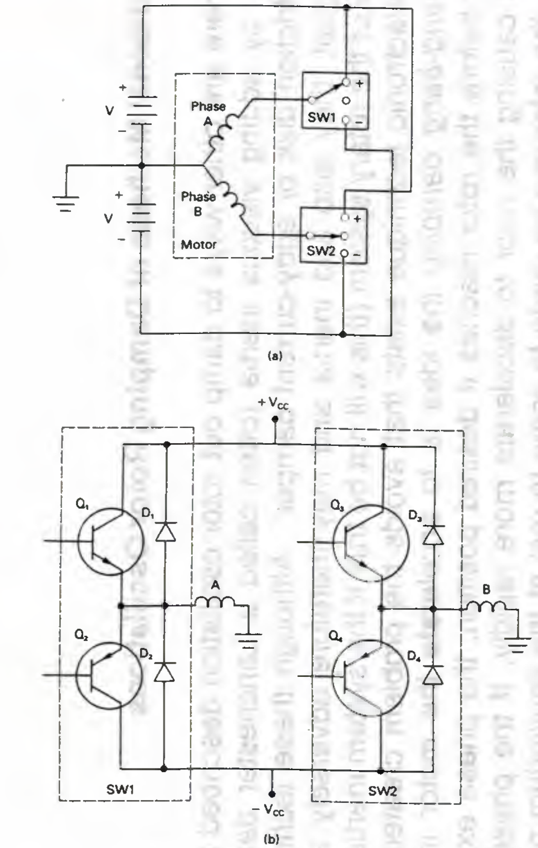


Figure 2–6 (a) Simplified bipolar two-phase switch stepper motor drive; (b) possible realization of the circuit in part (a).

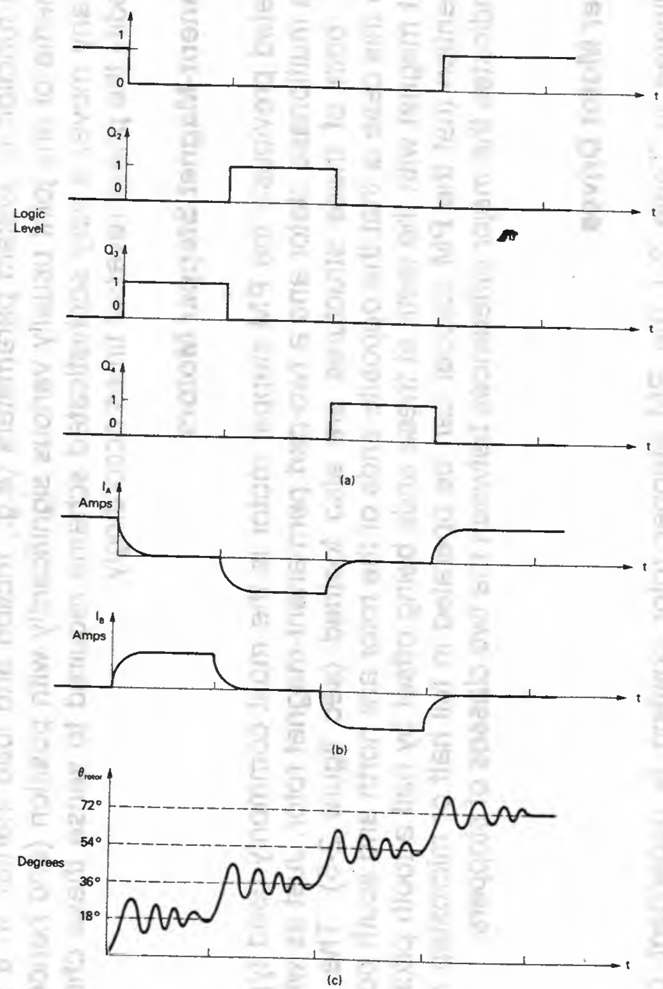


Figure 2-7 (a) Transistor logic signals; (b) motor phase currents; (c) rotor motion for the motor in Fig. 4.6.3 and the drive in Fig. 4.6.4a.

TABLE 2-1. DIFFERENCE BETWEEN PM AND VR STEPPER MOTOS

Character	PM motor	VR motor
1. Motor	Magnetize	Non magnetized
2. Rotor position	Depend on stator excitation polarity	ndependent of stator excitation polarity
3. Rotor inertia	High due to magnet	Low (no magnet)
4. Mechanical	Not as good (due to high inertia)	Good (low inertia device)
5. Induction	Low due to rotor offset	Generally high for same torque rating
6. Electrical response	Faster current rise (due to low inductance)	Slower current rise (due to higher inductance

Operation is accomplished with only two phases where that the equivalent VR stepper required four phases (see Figure 2-3).

Using a double-ended power supply, the motor in Figure 2-5 can be driven in the full-step mode with the switching arrangement shown in Figure 2-6a. It is observed that two *tristate* switches, SW1 and SW2 are required. This figure shows phase A positively energized and phase B off. A possible method or synthesizing such a device is shown in Figure 2-6b. The "fly-back" diodes are normally used to protect the power transistors from the "inductive kick" that occurs when an open circuit is suddenly placed in series with an energized inductor (e.g., Q1 is switched off). Without this protection it is possible to apply a voltage well in excess of the transistor's collector-emitter breakdown value during the switching interval.

For the motor in figure 2-5, it can be shown that each step is 18° and that there are therefore 20 steps/rev. The excitations and simple logic signals to the transistors that produce four rotor steps (i.e., 72°) are given in Figure 2-7. It is assumed that each step takes the same amount of time implying that any load attached to the rotor is moving at a constant velocity. During acceleration or deceleration of a load, the step spacing would, of course vary with time.

In actual operation, a microprocessor (e.g., the master) would determine the number of steps needed to cause a load to be moved a certain distance. This would be done for each joint in a robot application. The processor would then transmit the information together with direction and step timing data to a discrete digital hardware package. The latter would keep track or the total number of steps moved and would implement the appropriate switching sequence. Clearly this would represent open-loop joint control.

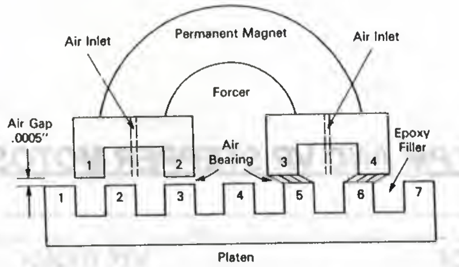


Figure 2-8 Components of a two-phase linear stepper motor. An air bearing is formed between the bottom of the forcer and the top of the platen. (Redrawn with permission of General Signal Corp., Santa Clara, CA.)

(a)
(b)
(c)
(d)

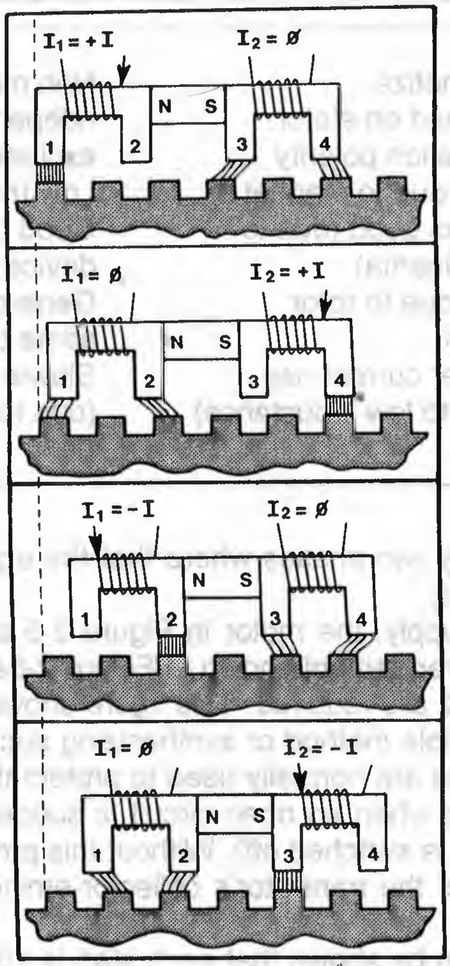


Figure 2-9 Motion in a two-phase linear stepper motor. Each new phase excitation produces a quarter pitch movement of the forcer relative to the platen. Thus a total motion of ¾ of a full pitch is indicated. (Reproduced with permission of General Signal Corp., Santa Clara, CA.)

2.6 LINEAR STEPPER MOTORS

An interesting variation of the conventional rotary stepper motor is the Sawyerprinciple linear stepper motor. Invented in 1969, this patented device is manufactured by Xynetics Corporation, Santa Clara, California, and consists of two major mechanical components. The first, a movable armature that is referred to as a forcer, is suspended over the second or fixed stator (also called a platen) (see Figure 2-8). A bearing is used to ensure that there is a constant space between the armature and the stator. In contrast to a conventional rotary stepper, which has a closed geometry, the platen's length is variable and depends on how far it is desired to move a load attached to the forcer. This configuration also differs from the rotary stepper in that the payload is directly driven by the motor and no mechanical advantage can be obtained through the use of a transmission.

As may be observed in Figure 2-8, the forcer consists of a permanent magnet (PM) and two electromagnets (EM) with four poles (two per electromagnet). The faces of the poles are grooved to form the pitch of the motor. Grooving of the platen produces a similar pattern. As will be seen shortly, the use of grooves allows finer resolution steps. In addition, when the spaces between both sets of grooves are filled with a nonmagnetic material, the resultant flat surfaces can be used to construct an air bearing between the bottom of the forcer and the top of the platen. This is accomplished by supplying air under pressure from tiny holes located in the forcer. The air bearing produces a surface with negligible starting and running friction.

The permanent magnet causes the platen and the unenergized forcer to be drawn together (except for the space provided by the bearing). Therefore, it is possible to position the platen so that the forcer travels above or below it. With no current flowing, the PM flux closes its path through the air gap, platen, and the poles of the electromagnets. The flux splits equally at both EM poles since the magnetic paths have approximately the same reluctance (e.g., see Figure 2-9a, poles 3 and 4). If current is switched through the electromagnets, commutation occurs. In general, the flux produced by the permanent magnet is about equal to that produced in the magnetic circuit by the current flowing through the windings. Thus as the current changes, the flux swings from a maximum value to almost zero.

The commutation, together with the relative positions of the forcer and platen teeth, causes forces to be produced which are perpendicular to the teeth and parallel to the platen. Since the teeth of the EMs are arranged in spatial quadrature from one pole face to the next, the PM's flux can be commutated by the electromagnets and emerges at polefaces whose teeth are misaligned with respect to those of the platen. The result is a tangential force that causes the forcer and platen teeth to move in such a manner as to minimize the gap (i.e., reduce the reluctance). This force produces motion along the length of the platen. A normal force also exists which pulls the forcer and platen toward one another, thereby providing the preload for the air bearing.

Figure 2-9a-d is used to illustrate the principles of operation outlined in the following paragraphs. In each of the figures, the direction of current and flux flow is indicated by the arrows. If electromagnet A (EMA) is energized, maximum flux density occurs at pole 2 and alignment is as shown in Figure 2-9a. When EMA is deenergized and EMB is energized, the maximum flux density occurs at pole 3 and the minimum density at pole

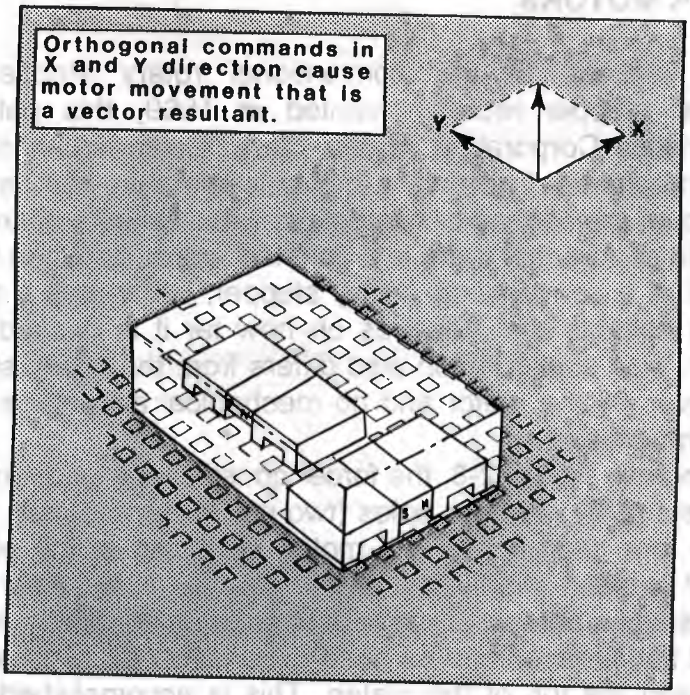


Figure 2-10 Two-axis linear stepper motor and platen. (Reproduced with permission of General Signal Corp., Santa Clara, CA.)

4. The attractive force at pole 3 causes the alignment of this pole with the platen's tooth on the right. Therefore, the motion is one-quarter of a tooth to the right and the motor and the forcer have the spatial relationship shown in Figure 2-9b. If EMB is deenergized and then EMA energized (with a current flow opposite to that shown in Figure 2-9a), motion again occurs to the right since pole I now has the maximum flux density, pole 2 the minimum, while poles 3 and 4 have the flux supplied by the PM. The forcer now resides at the location shown in Figure 2-9c. Finally, with EMA deenergized and EMB energized (also in the opposite direction from before), pole 4 has the maximum flux density, pole 3 the minimum, and poles I and 2 the flux supplied by the PM. To complete the cycle, EMA is again energized as in Figure 2-9a and the system has moved a distance of one tooth (i.e., the pitch) of the platen (the equivalent of full step mode in a rotary stepper). The frequency of the current cycling establishes the velocity with which the forcer moves.

Obviously, the positions of the forcer relative to the platen are discrete in nature if the current is cycled as described above. Used in this manner, the linear stepper has a full-step resolution defined by the spacing of the teeth on the poles. A typical pitch is 0.040 in. Thus for the sequence shown in Figure 2-9, the resolution is one-quarter of the pitch or 0.010 in. These positions are sometimes referred to as cardinal steps. To obtain a finer resolution between steps, it is possible to use current values that are between those used in the full-step mode. That is, the motor is operated in microstep mode (see Section 2-3).

It is also possible to construct this type of motor so that it consists of two orthogonally oriented forcers assembled on one motor frame. To complement the forcer, the platen is constructed of square teeth in a Waffle pattern, as shown in the two-axis linear stepper of Figure 2-10. This configuration allows motion in both the x and y directions or along any vector in the x-y plane

As indicated previously, the linear stepper is a direct-drive motor. This implies that the control resolution and force needed to position and move the load are defined solely by the motor's capabilities. Thus for any application requiring resolution better than that of the tooth pitch, a controller capable of microstepping to the desired resolution must be used. Additionally, the speed-force curve for the motor-driver combination must be examined carefully to ensure that the motor can produce the required forces over its operating speed range.

2.7 BRUSHLESS DC MOTORS

In electrically actuated robots, brush failures in the dc servomotors used on the joints account for a major source of downtime. These devices wear, causing the effective terminal resistance of the armature to increase significantly, thereby reducing the efficiency of the servo. Increased heating and torque reductions are two of the major consequences. In addition, as the motor turns, arcing between the brushes and commutator segments occurs due to the sudden interruption of current in the particular armature coil being commutated.* Besides contributing to mechanical deterioration of the brushes themselves, which can limit their use in "clean room" applications (e.g., in the handling of semiconductors), this situation also prevents robots so actuated from being used in explosive environments. Finally, the electromagnetic interference (EMI)

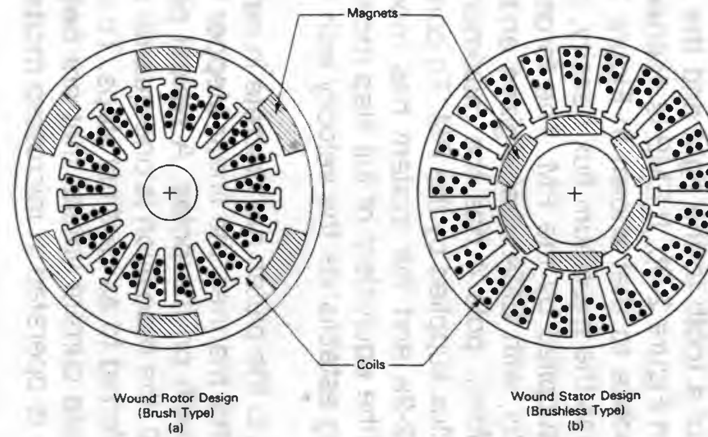


Figure 2-11 Cross sections of two servomotors: (a) standard DC (brush-type); (b) brushless DC-type. (Redrawn with permission of Electro-Craft Corp., Hopkins, MN.)

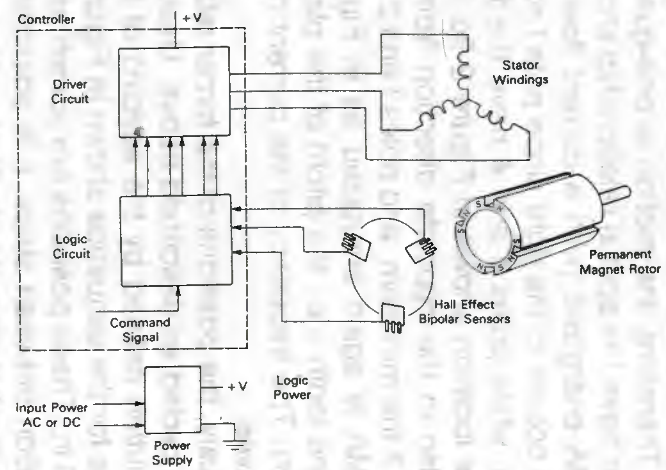


Figure 2-12 Eight-pole, three-phase winding brushless DC motor with electronic commutation scheme utilizing three Hall effect sensors. (Redrawn with permission of Micro Switch, a Honeywell Division, Freeport, IL.)

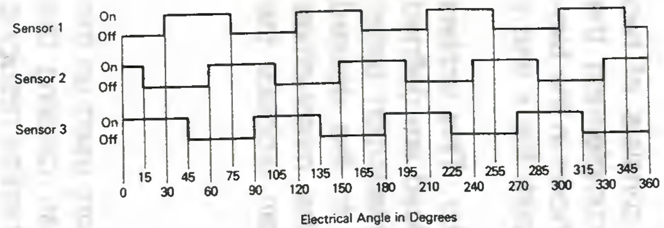


Figure 2-13 Outputs of three Hall effect sensors used in the 8-pole, 3-phase BDCM of Fig. 4.7.2. (Redrawn with permission of Micro Switch, a Honeywell Division, Freeport, IL.)

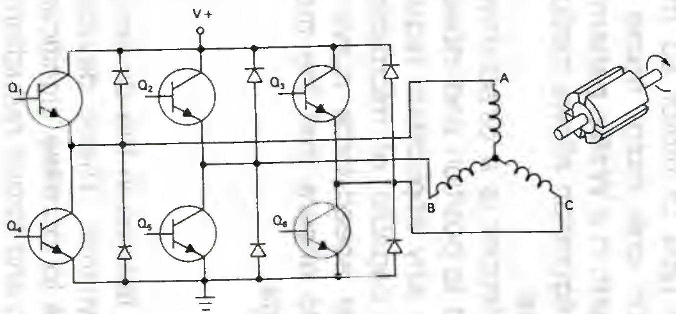


Figure 2-14 Three-phase driver circuit for the BDCM with a wye configuration stator. (Redrawn with permission of Electro-Craft Corp., Hopkins, MN.)

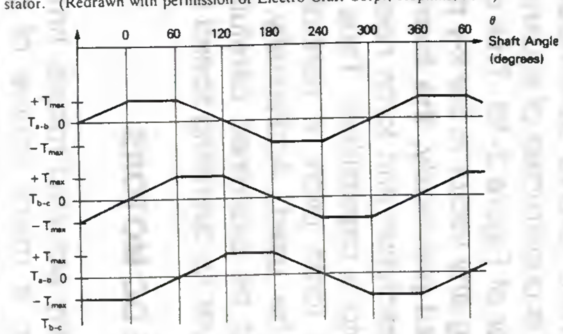


Figure 2-15 Torque versus motor shaft angle θ for a BDCM assuming a *constant current* in each of the three phases. (Redrawn with permission of Electro-Craft Corp., Hopkins, MN.)

produced by the electrical spark can also create reliability problems for other electronic devices working in the vicinity of the robot.

In recent years, dc motors have been developed which avoid many of the difficulties attributable to the brushes of a standard servomotor. As shown in Figure 2-11, the brushless dc motor (BDCM) can be viewed as an "inside-out" version of a standard dc servomotor. It is observed that the rotor of the brushless device contains the permanent magnets (two in this case, thereby producing a four-pole motor) whereas the stator consists of the coil segments and iron.
Since there is no mechanical commutation of the coils in a BDCM due to the elimination of the brushes and commutator bars, a method of properly energizing the stator coil segments must be provided. This is often accomplished by placing inside the motor itself solid-state devices (e.g., Hall effect bipolar sensors) that determine the actual position of the magnets as the rotor turns. A simple logic circuit then processes the information provided by these sensors, thus enabling the appropriate stator coil to be excited. As an example, consider the eight-pole (four north and four south), three-phase winding BDCM and the electronic commutation scheme shown in Figure 2-12. The output of a Hall effect sensor is high (logical 1) when the south pole of a permanent magnet is in close proximity to it. The output is low (logical 0) if the magnet's north pole is passing by. With the sensors placed approximately 120 mechanical degrees apart and the four magnets 90 mechanical degrees apart as indicated in Figure 2-12, it is easily demonstrated that the outputs of the three sensors are the waveforms shown in Figure 2-13. These signals can be processed by a simple logic circuit to determine the position of the magnets at any instant of time. This information is then utilized by the driver circuit to cause the appropriate motor windings to be energized.

Ideally, it is possible to produce a torque output that is constant with respect to angular position. To see this, consider the three-phase driver circuit shown in Figure 2-14 and assume that the BDCM windings are arranged in a wye configuration. If a constant current is applied to each winding and the rotor is moving at a constant angular velocity, the torque produced by each of the phases as a function of the angular position of the motor shaft 0 is shown in Figure 2-15. It is important to understand that the total or net torque Tgen produced by the motor is the algebraic sum of the three torques Ta-b, Ta-c and Tb-c, Clearly, if we permit a constant current to flow in each of the three phases, Tgen will not be constant as is desired.

2.8 DIRECT-DRIVE ACTUATOR

One of the major problems with commercial robots is that at certain speeds (usually but now always low), mechanical resonance's are excited and exceedingly rough motion results (a so-called "palsy" is exhibited). Although some of the difficulty can be traced to the structure of the manipulator itself, it has been found that one of the primary causes of poor motion is the mechanical devices used to couple the motion of the actuator to the output of each joint mechanism. For example, the harmonic drive, which is currently used extensively for this purpose, contributes significantly to the low-speed performance degradation due, in part, to its compliance (i.e., "springiness") and also to machining errors which are inherent in the design and cannot be entirely eliminated. A design that does not employ such mechanical units is obviously desirable

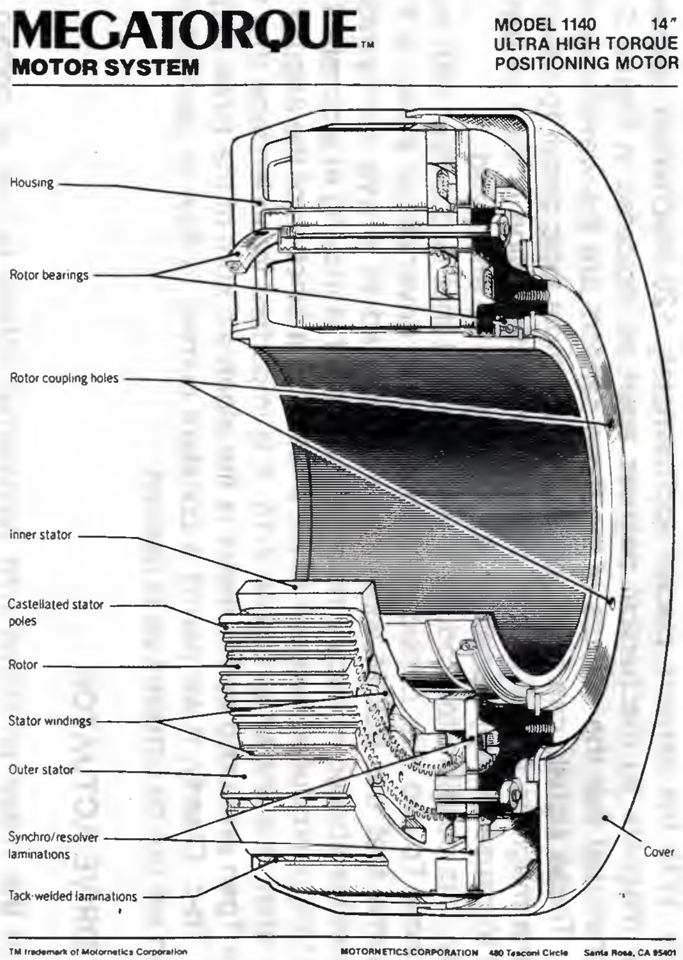


Figure 2-16 Cross section of a Motornetics, Inc. "Megatorque" direct drive motor. (Reproduced with permission of B. Powell and the Motornetics Corp., a subsidiary of NSK, Santa Rosa, CA.)

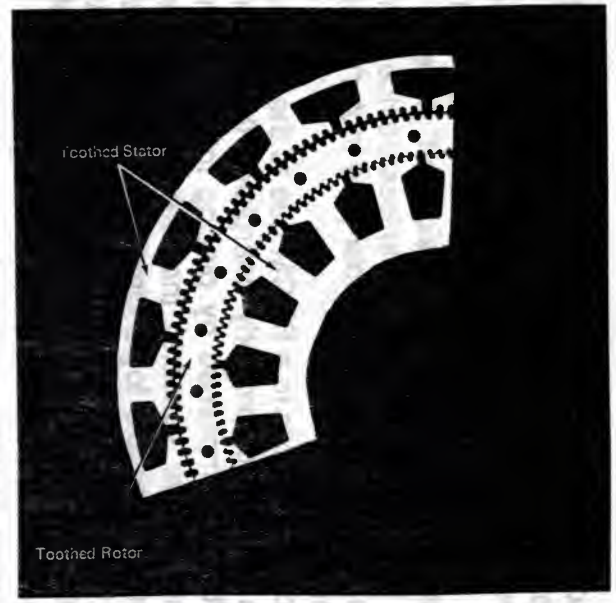


Figure 2-17 A quarter section of the laminations of a toothed megatorque motor showing two stators sandwiched around a toothed rotor. (Reproduced with permission of B. Powell and the Motornetics Corp., a subsidiary of NSK, Santa Rosa, CA.)

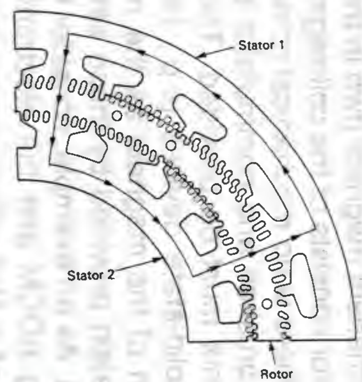


Figure 2-18 Flux path for megatorque motor laminations. The center portion (the rotor) moves with respect to the outer and inner laminations (stators 1 and 2). (Reproduced with permission of B. Powell and the Motornetics Corp. a subsidiary of NSK, Santa Rosa, CA.)

i.e., a direct connection between actuator and load is indicated; this is referred to as a direct-drive approach). Nevertheless, despite the acknowledged difficulties with coupling devices such as the harmonic drive, they are still utilized extensively in manipulators. The justification for this is that "torque multiplication" and increased position resolution that such components afford are absolutely critical in the successful design of robots. Without these attributes, motors would have to be extremely large, bulky, and quite costly since they normally produce maximum torque at speeds too high to be of any use in a direct-coupled application (e.g., thousands instead of tens of rpm). In addition, it would be necessary to employ very high resolution encoders that also would be costly (this is, in fact, one of the primary arguments against a direct-drive design).

In the early 1980s, however, a new motor was developed which does permit a practical direct-drive robot to be constructed. This novel actuator, manufactured by Motornetics Corporation and called a Megatorque motor, produces extremely large torques (e.g., 35 to 1000 ft-lb) at low angular velocities (e.g., 30 rpm) without the need for a speed reducer. In addition, a position-sensing element that is an integral part of the motor has been developed and permits the resolution of a robot based on such a motor to be at least as good as those manipulators that currently utilize more traditional sensors (e.g., optical encoders).

In effect, the Megatorque motor is a three-phase synchronous device that is operated as a brushless dc actuator, i.e., electronic commutation is employed. Unlike the BDCM discussed in the preceding section, however, this one is a variable-reluctance device and consequently does not contain a permanent magnet (see Figure 2-16). The heart of the motor is a series of laminations that combine the rotor and stator. One such lamination is shown in Figure 2-17. It is observed from this figure that there is a thin annular rotor mounted between two concentric stators. Both stators react with the rotor, thereby producing a significant torque multiplication (over a single stator design). The large number of magnetic teeth located on the rotor and the two stators is also instrumental in large torque production.

The three-phase magnetic field is produced by 36 stator windings (18 on each of the two stators). There are 150 teeth on each of the stators and the rotor, which perform as motor poles in this design. Torque is produced by sequentially energizing these poles. For a single rotor revolution, there are 150 ac cycles, which, in effect, creates a 150:1 gear reduction with the corresponding torque multiplication. * It is interesting to note that without the toothed rotor/stator combination, a more conventional motor would require 300 poles per phase or 900 windings in order to yield the same performance!

Another advantage of the sandwiching of the rotor between the two stators is that the magnetic flux travels over an extremely short path, as shown in Figure 2-18. It is observed that the flux from one stator passes radially through the thin rotor into the other stator. (This is to be compared with the more conventional motor design, where a flux path of 180° through the rotor is typical.) Such a configuration lowers the magnetic resistance, and hence the motor has a high torque-to-input power ratio (high flux per ampere-turn).

Unlike the more conventional dc servomotor, the rotor of the Megatorque motor does not carry any current. As a consequence, there is little heating of the rotating member and therefore heat dissipation problems are minimized. This is a particularly important attribute in robot applications, where the actuator is often operated in a stall

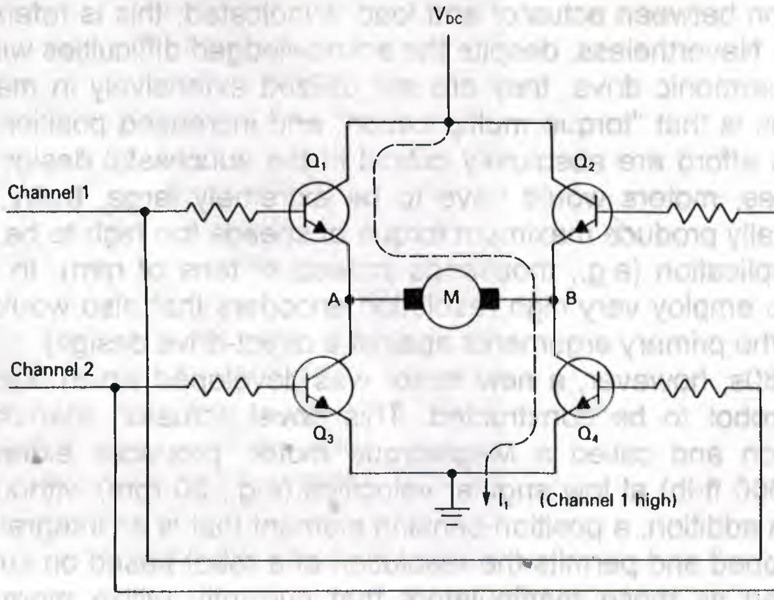


Figure 2-19. "H" type servo amplifier (one power supply required). Path of current I_1 is shown for channel 1 high and channel 2 low. (Redrawn with permission of Electro-Craft Corp., Hopkins, MN.)

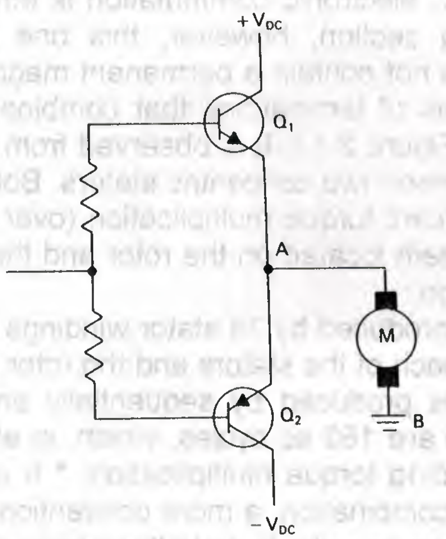


Figure 2-20. "T" type servo amplifier (two power supplies required). (Redrawn with permission of Electro-Craft Corp., Hopkins, MN.)

condition [e.g., when a load is being held (against gravity) in one place]. Any heat that is produced (in the stators) is easily conducted away by the case. In addition to essentially removing the temperature limitations, which are associated with other motors, the direct-drive device does not have the demagnetization problems associated with many of these actuators. Thus there is no danger of causing a permanent degradation of the motor's electrical performance due to a large (and perhaps inadvertent) current spike.* All that happens in the Megatorque unit is that the iron laminations are driven into saturation, a situation that is completely reversible by reducing the flux (current excitation).

When combined with an integral position-sensing element also developed by Motornetics, we have discussed of this resolver-like device), a successful direct-drive robot is feasible. It is important to understand, however, that not every robotic configuration can utilize this new actuator technology. The primary reason is that the Megatorque motor is extremely heavy, and thus it must be incorporated into a manipulator that does not require the actuator to be carried by the particular axis. one such design is the SCARA-type robot, where the two major axis motors can be placed so that their weights are supported by the manipulator structure rather than by a motor-produced torque. The first commercial direct-drive robot was demonstrated at the Robots 8 conference held in Detroit in June 1984 by the Adept Corporation. Called the Adept 1, this SCARA class manipulator incorporates two Megatorque motors and shows impressive low- and high-speed performance.

2.9 SERVO AMPLIFIERS

As shown in Figure 2-2, a servo amplifier must be used to convert the low-power command signals that come from the master computer and are then "processed" by the joint computer to levels that can be used to drive the joint motor. An amplifier that can provide the necessary logic and drive for a stepper motor was described in previous section. In this section we consider possible configurations that can be used to drive a dc servomotor. Specifically, pulse-width-modulated (PWM) and linear amplifiers incorporating voltage feedback or voltage and current feedback will be discussed.

2.9.1 Linear Servo Amplifiers

Two basic classes of linear servo amplifiers exist: (1) the H type and (2) the T type. These are shown in Figures 2-19.1 and 2-20.2, respectively. The first of these, the H is sometimes called a bridge amplifier, and has the advantage of requiring a single or unipolar dc supply. However, it is not always easy to operate in a linear fashion, and because the motor must be "floated" with respect to the system ground, current and/or voltage feedback is not easy to achieve. In actual operation, one set of diagonally opposite transistors is turned on [e.g., Q. and Q4 (or Q. and Q3)]. It can be seen that if the first of these sets is made to conduct by applying a positive control voltage to channel 1 (and grounding channel 2), the armature voltage $V_{AB} < +V$ and the motor will turn (e.g., in the clockwise direction). When the control signals on channels 1 and 2 are reversed, the second set of transistors conducts, thereby making $V_{AB} > -V$. The motor will now turn in the opposite or counterclockwise direction. The actual size of thc armature voltage, and hence the motor speed, will depend on the amount of base current supplied by the control circuitry that precedes the power amplifier stage (e.g., a preamplifier, not shown in Figure 2-19).

The second general type of servo amplifier, the T. requires a bipolar dc supply, as shown in Figure 2-20. However, it is easy to drive and since the motor does not have to float with respect to ground, current and/or voltage feedback is easy to implement. Since complementary power transistors are employed (see Figure 2-19), a single bipolar control signal can be used to turn on either Q_1 or Q_2 thereby making V_{AB} either $< +V$ or $> -V$ and producing the desired bidirectional rotation. In the T configuration, it is important to bias the transistors so that Q_1 and Q_2 are not both on at the same time since output transistor failures are likely to occur if this happens (i.e., they conduct simultaneously).

An undesirable characteristic of a T servo amplifier is the "deadband" or "crossover distortion" that exists around zero output voltage. This produces an armature drive voltage that is a nonlinear function of the servo amplifier input for small positive and negative inputs signal. The problem can be reduced by keeping both transistors on around the zero-voltage region. From what was said above, it is clear that care must be taken to prevent simultaneous operation of the transistors from occurring when large currents flow.

It is important to note that the amplifiers in Figures 2-19 and 2-20 do not have any type of flyback protection shown. However, this is absolutely essential since the inductance in the servomotor armature can produce an "inductive kick" when the power amplifier transistors are either suddenly all turned off or when the motor is "plugged" (i.e., the armature voltage is rapidly reversed to provide dynamic breaking). Thus regardless of which configuration is used, flyback diodes (or some other method of protecting the power transistors from breakdown) must be placed across the collector-emitter terminals of the output transistors. Failure to do this risks a collector-to-emitter short circuit, which, as explained previously, can cause a runaway condition—an especially dangerous occurrence in robotic applications.

There are other factors that must be considered when working with linear servo amplifiers: for example, the power dissipation capability of both the power transistors and the associated heat sinks, the need to provide some type of active cooling (e.g., by using a fan), and the need to protect both the power transistors and the motor from current overloads by using current limiting. The last of these factors is particularly important in robotic applications since it is not at all uncommon for a stall to occur in the middle of a move due to the manipulator coming into contact with a foreign object that has accidentally found its way into the workspace. Clearly, one or more motors will stall in this case and some type of protection is absolutely essential in order to prevent amplifier and/or motor damage or destruction. Current limiting is one such technique, although fusing of motors and software methods.

2.9.2 Pulse-Width-Modulated Amplifiers

One of the major difficulties with the linear amplifiers described in the preceding section is that, very often, the output is only a fraction of the total supply voltage, for example, during the initial or final portions of a move or when the move is deliberately

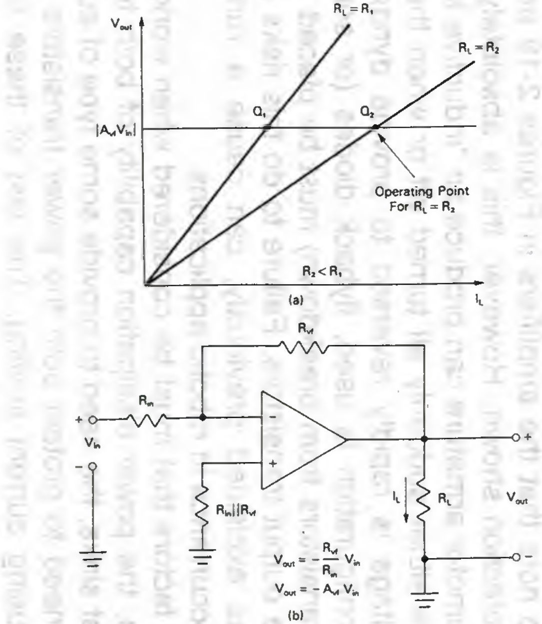


Figure 2-22. Voltage feedback amplifier: (a) *V-I* operating characteristic; (b) circuit diagram.

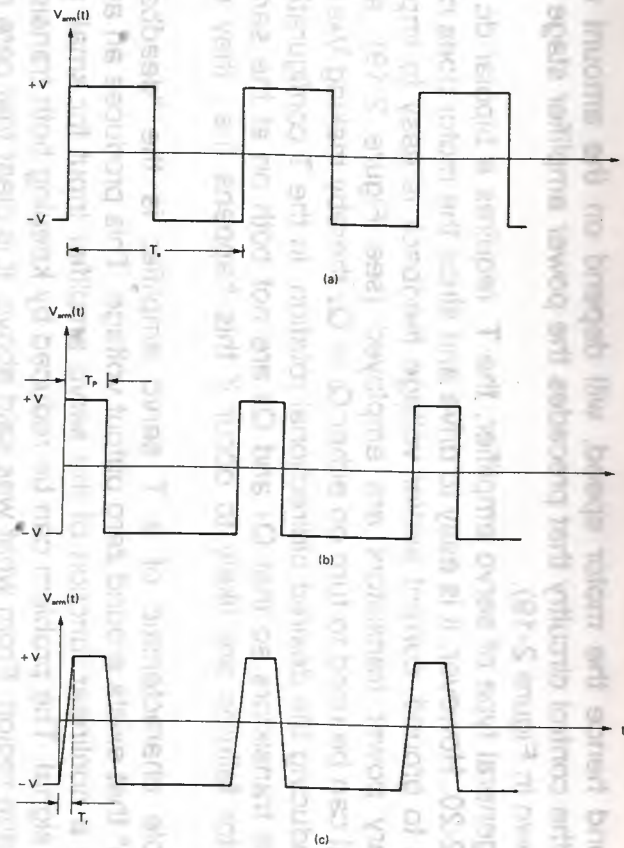


Figure 2-21 Typical PWM waveforms: (a) no load PWM output, ideal switch, $(V_{arm})_{ave} = 0$, motor does not turn; (b) loaded PWM output, ideal switch, $(V_{arm})_{ave} = -V/2$, motor turns CCW; (c) same as part (b), except nonideal switch and power transistors in active region during T_r.

performed at low speeds. This is accomplished by operating the power transistors in their active (i.e., linear) regions, which means that the collector-to-emitter voltage drop VCE of the transistor(s) that is (are) conducting is significant. Consequently, the power dissipated in the collector (i.e., the product of collector current and the collector-to-emitter voltage) can be large (on the order of tens of watts and sometimes as high as 100 W), so the transistors and heat sinks must be sized accordingly. Although it is certainly possible to obtain these large transistors with the technology currently available, the added cost incurred is not always warranted. Fortunately, it is now possible to use a different approach that is generally more cost-effective (i.e., pulse-width modulation, PWM).

With the advent of power transistors that can be switched at megahertz rates, the use of PWM amplifiers to drive servomotors in robotic applications, as well as other incremental motion applications, has become quite practical and attractive. The major advantage of a switched device over a linear device is that in the former the power transistor is either "off" or in (or close to) saturation. In either case, the power dissipated in the collector is considerably less than in an equivalent linear amplifier. This is easily understood by recognizing that since little or no collector current flows when the transistor is turned off, the power dissipation is quite small. When current does flow, however, the transistor is in saturation, which means that the drop across its collector is only 1 or 2 V. Thus the dissipation is still quite small (i.e., under 12 W for a continuous armature current of 6 A). An equivalent linear device might dissipate 72 W (assuming a 12-V drop across the collector).

Just as with linear servo amplifiers, PWM devices can be of the H or T type and the same comments concerning the advantages and disadvantages of both are pertinent (see Figures 2-19. and 2-20.). However, unlike the linear case, the output voltage of the T or H circuit will be almost equal to the full value of either the positive or negative dc supply voltage (see Figure 2-21.).

How can these types of signals provide the required variation in armature voltage and hence rotor speed? The answer to this question is found by recognizing that the servomotor is a low-pass filter. With Ts defined as the period of the switching signal waveform, then if the radian switching frequency $\omega s = 2\pi/Ts >> \omega s$ the electrical pole of the motor (i.e., $\omega s > 100\ \omega e$), the filtering action of the motor will cause the effective armature voltage to be the "average value" of the waveforms in Figure 2-21.,

The waveforms in Figure 2-21., it is seen by inspection that the motor will not move for the square wave in part (a) because (Varm)ave = 0. Whereas the nonzero average value of this quantity for the waveforms in (b) and (c) will produce rotor motion. It will not be strictly correct if the switching frequency is too low. For example, if it is only about 10 times higher than the electrical pole of the motor, the effective armature voltage will be somewhat less than the average value and the armature current may exhibit significant ripple.

In actual use, a PWM servomotor drive can be made to produce practically any type of acceleration, velocity, or position profile that might be required in a given application. For example, if it is desired to cause a servomotor to turn with a trapezoidal velocity profile (see Figure 2-22), this can be achieved by making the pulse width, Tp in

Figure 2-21, vary trapezoidally with time. In a robotic application the joint processor converts the velocity error samples into equivalent values of Tp. This is accomplished by causing the associated control logic to command the appropriate power transistor(s) in the PWM amplifier to turn on for Tp milliseconds. In view of the discussion of the preceding paragraph, faithful reproduction of the desired profile will occur only provided that the switching frequency is "high enough." This statement, in effect, implies that the frequency must be chosen so that the sampling theorem is satisfied.

Unlike the linear servo amplifier, there is another cause of power dissipation in a PWM device, and this places a practical upper limit on the switching frequency.

Since switching cannot physically occur instantaneously but rather takes a finite time T., the power transistors spend a portion of the switching cycle in the active region (see Figure 2-21c). If the switching rate is extremely high, it is possible for this time to become a significant portion of the overall switching period, with the result that the overall power dissipation can be quite large, approaching that of the linear ease. As a Consequence, practical PWM servo amplifiers usually work at switching rates of 1 to 15 kHz. (The lower limit is often determined by human factors, since a low-frequency switching rate can produce annoying and sometimes intolerable audible noise.)

2.9.3 Effects of Feedback in Servo Amplifiers

In this section the effect of using voltage, current, and combined voltage and current feedback with the power amplifier is Considered. The reader should recall that the use of feedback can "stabilize" whatever quantity is being fed back. Thus, in the ease of voltage feedback, an amplifier's output voltage is held constant regardless of changes in the load s impedance. This is sometimes referred to as a voltage-stabilized amplifier. Figure 2-23 shows the voltage-current Characteristics for an amplifier with voltage feedback and a corresponding op amp implementation. Note that the intersection of any single constant resistance line with a particular constant-voltage output Curve defines an operating point. It is important to understand that regardless of the value of the load resistance, the voltage-stabilized amplifier will produce an output V_{out}, which are proportional to both the input Vin and the gain factor V_{vf} (units of v/v).

2.9.4 Voltage amplifier driving a servomotor

A dc servomotor driven by an amplifier utilizing voltage feedback is shown in Figure 2-24ab. As discussed in previous sections, this amplifier usually consists of a power stage preceded by a preamplifier (normally implemented by an op amp). The characteristic of the device is that it has low output impedance and is therefore termed a voltage amplifier. It is convenient to think of a system utilizing this group of components as producing a voltage-controlled velocity.

The voltage amplifier itself will have a finite bandwidth, and in most robot applications a single-pole model will adequately represent its frequency). Thus we will - assume that its transfer function is given by

$$\frac{Vout(s)}{Vin(s)} = \frac{Avf}{1+s\tau a} \qquad (2\text{-}1)$$

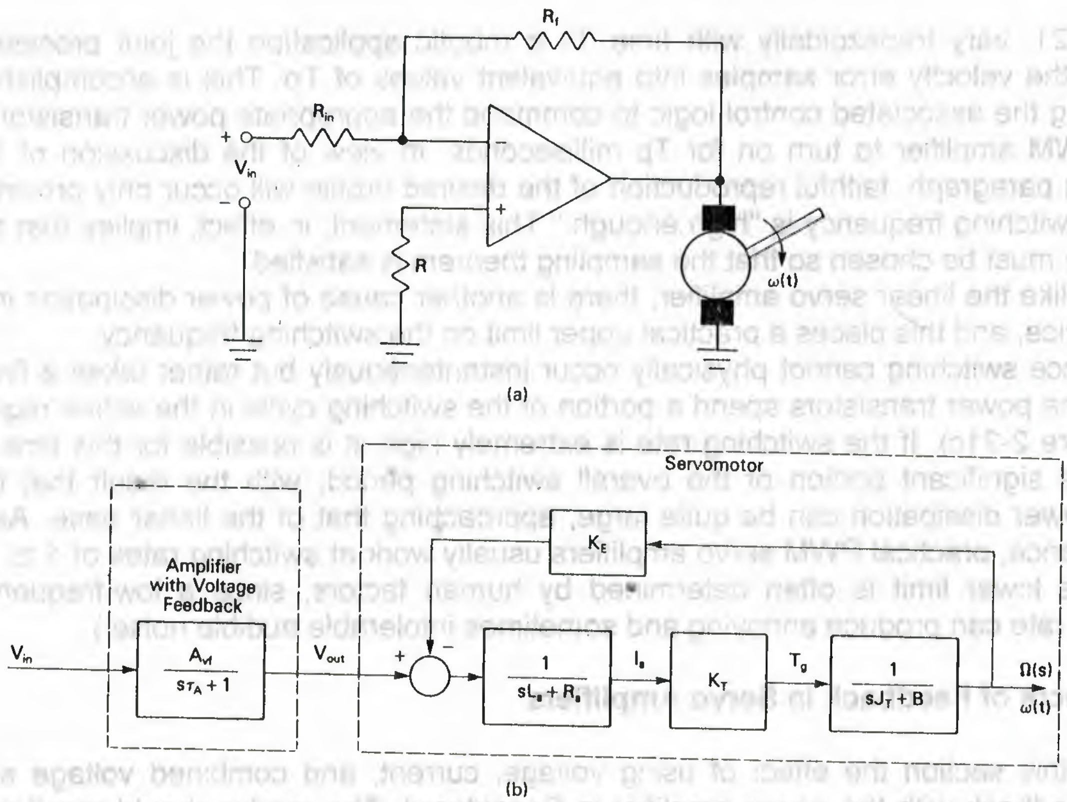


Figure 2-24: (a) Servomotor driven by an amplifier utilizing voltage feedback; (b) block diagram representation of part (a). [Part (b) redrawn with permission of Electro-Craft Corp., Hopkins, MN.]

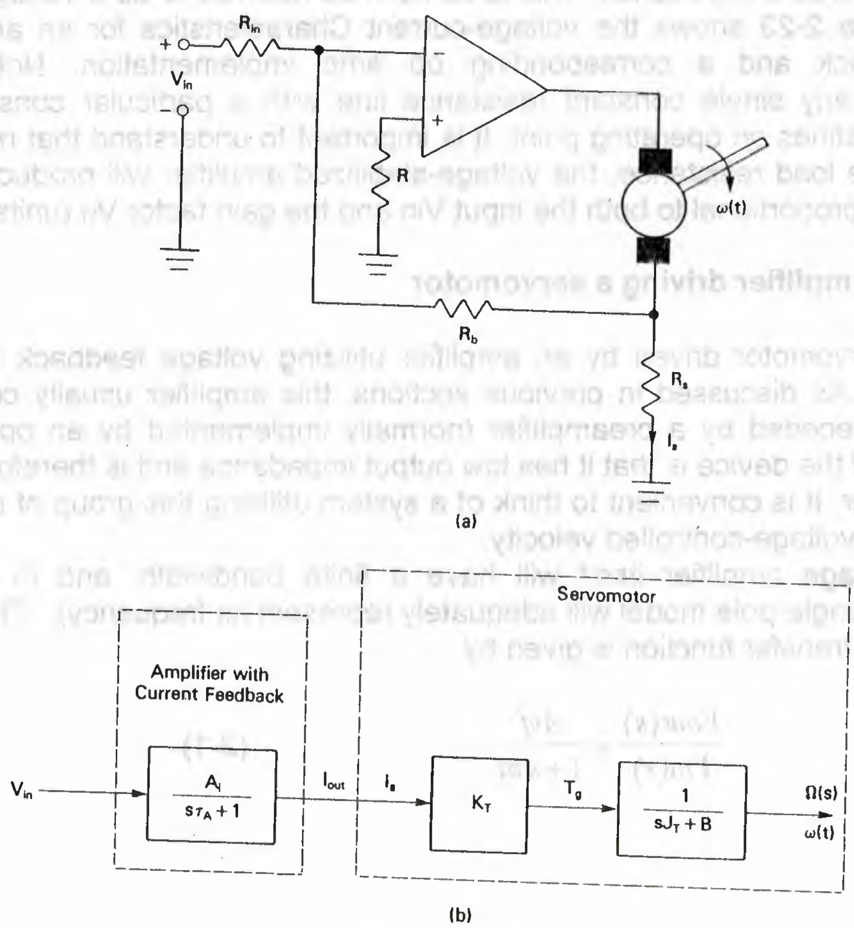


Figure 2-25. (a) Servomotor driven by an amplifier with current feedback; (b) block diagram representation of part (a). [Part (b) redrawn with permission of Electro-Craft Corp., Hopkins, MN.]

where $1/\tau a$ is the amplifier radian frequency bandwidth and *Avf* is the magnitude of the gain, given by the relationship

$$\left| Avf = \frac{Rf}{Rin} \right| \qquad (2\text{-}2)$$

Figure 2-24b show-s a block diagram representation of a de servornotor driven t)v such an amplifier. The overall transfer function of this configuration is obtained by multiplying the transfer functions of Eqs. (2-1) and (2-2) and relates the motor shaft speed (transform) to the input voltage (transform). Thus

$$\frac{\Omega(s)}{Vin(s)} = \frac{(Kt / LaJt)}{s + [(RaJt + LaB)]s + (KtKe + raB) / LaJt(1 + s\tau a)} * \frac{Avf}{(1 + s\tau a)} \qquad (2\text{-}3)$$

The important point to note in this equation is that the use of voltage feedback has not affected the location of the motor poles. This, of course, assumes that no loading exists between the two devices, which is true for a zero-output impedance amplifier.

2.9.5 Current amplifier driving a servomotor

An alternative method of driving a servomotor is with a current amplifier (high-impedance source), as shown in Figure 2-25ab. The reader will recall that an amplifier with current feedback produces a constant output current for a given input voltage. Examination of the dynamic equations of a dc servomotor reveals that this implies that the terms associated with the electrical behavior (e.g., armature resistance and inductance together with the back EMF Constant) do nest influence the current actually being delivered to the motor. * By using the Laplace transforms of equestions, ignoring the constant-torque terms for the same reasons given in previous section .

$$KtIa(s) \doteq (sJt + B)\Omega(s) \qquad (2\text{-}4)$$

If Eq. (2-4) ii rearran5ed to obtain a transfer function relating armature current to shaft

velocity,we have

$$\frac{\Omega(s)}{Ia(s)} = \frac{Kt}{sJt + B} \qquad (2\text{-}5)$$

Equation (2-5) can be viewed as modeling a device that produces a *current-controlled velocity.* Another way of looking at this result is to recognize that since torque is proportional to armature current, a current amplifier actually produces *torque control.*

In reality, the current amplifier has a finite bandwidth and can be modeled as

$$\frac{Iout(s)}{Vin(s)} = \frac{Ai}{1 + s\tau a} \qquad (2\text{-}6)$$

where $1/\tau a$ is again the amplifier bandwidth and *Ai,* the gain of the amplifier (units of amperes/volt), is a function of the input, feedback, and sense resistors. That is,

$$\left| Ai = \frac{Rb}{RinRs} \right| \qquad (2\text{-}7)$$

In Eq.(2-7) it is assumed that Rs << *Rb.*
The overall transfer function of a motor driven by a current amplifier is the product of the transfer functions in Eqs. (2-5) and (2-6), so that

$$\frac{\Omega(s)}{Vin(s)} = \frac{KtAi}{(sJt + B)(1 + s\tau a)} \qquad (2\text{-}8)$$

Figure 2-25b shows a block diagram representation of this system. Comparison of Eqs. (2-3) and (2-8) indicates that the poles of the motor have been altered by the use of current feedback. In fact, the motor is seen to behave like a one-pole, *rather than a two-pole device.* Note that only the mechanical parameters of the system (i.e., the total inertia Jt and the viscous damping *B* have an effect on the behavior of the servo. It can be shown that the elimination of the pole (sometimes referred to as the *electrical pole)* due to the armature elements results in a larger velocity loop bandwidth.

SUMMARY

In this chapter we have presented a detailed discussion of the typical control structure used in each of the joints of a modern industrial robot. It has been demonstrated that we must ensure that the position servo bandwidth is large enough to reproduce faithfully the profile of the desired position (thereby keeping the tracking error well within bounds) but not so large as to permit the servo to respond to the individual updates (i.e., the set points). It has also been found necessary to include an integrator as part of the servo loop's compensator due to the gravitational and friction disturbances that invariably act on most of the joints of a robot. Thus it is not surprising that a typical robot servo utilizes a PID controller. We have also found that most robots either employ a single position sensor (on each axis, e.g., an incremental encoder), and derive the velocity digitally from the information provided by this device, or else both an encoder and an analog tachometer are utilized to provide the required position and velocity data. In addition to the material on the operation of the servo itself, we have presented detailed information on a variety of robotic actuators, including the dc servomotor. Stepper (both rotary and linear) and brushless motors, and finally, a novel type of direct-drive motor that has already been incorporated into a commercial, high performance robot.

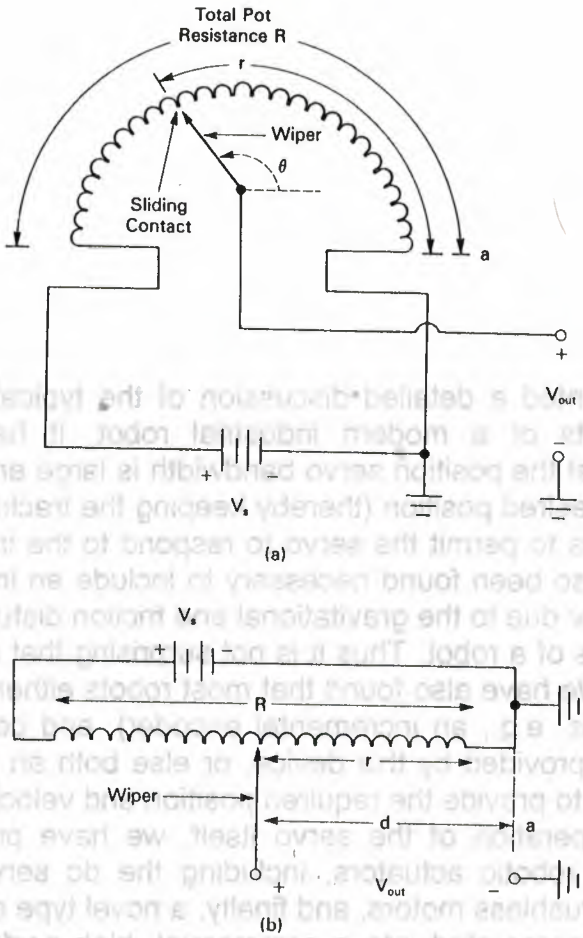


Figure 3-1 Wire wound potentiometer ("pot"). Wiper makes physical contact with wires on the resistive coil. Note: point "a" corresponds to zero output (i.e., zero resistance): (a) rotary—output proportional to θ; (b) linear—output proportional to "d."

CHAPTER 3. ROBOTICS SENSORY DEVICES

3.1 INTRODUCTION

In this chapter we describe the operation of a variety of sensory devices that either are now used on robots or may be used in the future. In general, it is found that some arc inherently digital devices, whereas others are essentially analog in nature. Sensors can be divided into two basic classes. The first, called internal state sensors, consists of devices used to measure position, velocity, or acceleration of robot joints and/or the end effecter. Specifically, the following devices that fall into this class will be discussed:

- Potentiometers ("pots")
- Synchros
- Resolvers
- Linear inductive scales
- Differential transformers (i.e., LVDTs and RVDTs)
- Optical interrupters
- Optical encoders (absolute and incremental)
- Tachometers
- Accclerometers

3.2 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 potentiometer, synchro, resolver, and LVDT. It will be seen that some of these devices are inherently analog and some are digital in nature.

3.2.1 Potentiometers

The simplest device that can be used to measure position is the potentiometer or "pot." Applied to robots, such devices can be made to monitor either angular position of a revolute joint or linear position of a prismatic joint. As shown in Figure 3-1, a pot can be Constructed by winding a resistive element in a coil configuration. By applying a dc voltage V_s across the entire resistance R. the voltage V_{out}, is proportional to the linear or

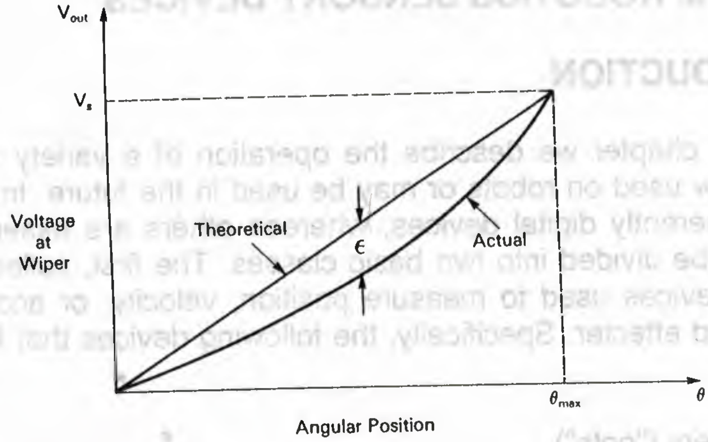


Figure 3-2 Output of a pot showing the theoretical linear and actual nonlinear characteristics as a function of angular position θ.

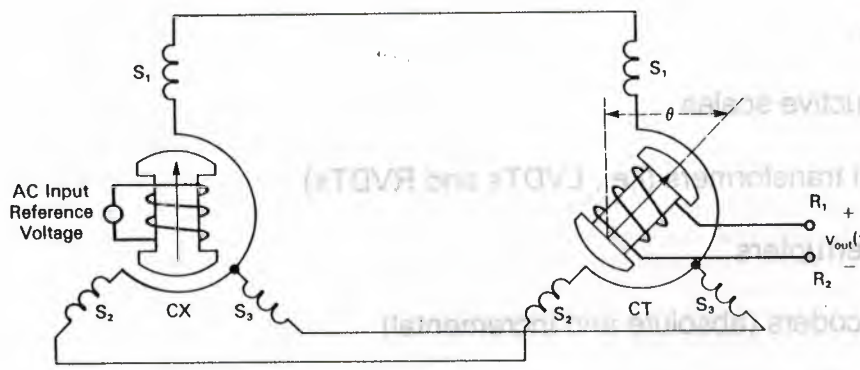


Figure 3-3 A two-element (control transmitter CX and control transformer CT) synchro system used to measure angular displacement. θ is the *relative* angle between the rotors of CX and CT.

rotary distance of the sliding contact (or "wiper") from reference point a. Mathematically, if the resistance of the coil between the wiper and the reference is r, then

$$V_{out} = (r/R)V_s$$

For the pot to be a useful position sensor, it is important that the resistance r be linearly related to the angular distance traveled by the wiper shaft. Although it is possible to obtain pots that are nominally linear, there is always some deviation from linearity as shown in Figure 3-2. Generally, the nonlinearity of a pot (expressed as a percent) is defined as the maximum deviation δ from the ideal straight tine compared to the full-scale output. That is,

$$N.L. = 100\ (\delta/Vmax)$$

The inevitable presence of this nonlinearity in any pot makes its use in systems where excellent accuracy measurement is required difficult and often impractical. Thus except in the case of robots where extreme accuracy is not needed (such as in educational devices), the pot is not generally used as a primary positionmonitoring sensor. In a later section of this chapter, it will be seen, however, that it is possible to utilize this type of device as one of the components in a positionmeasuring scheme.

3.3 SYNCHRO SYSTEM

As mentioned above a significant practical problem with the pot is that it requires a physical contact in order to produce an output. There are, however, a variety of sensing devices and techniques that avoid this difficulty. The first one that we discuss is the synchro, 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 recent years however, advances in solid-state technology have again made the cincher a possible alternative for certain types of systems, among them robots. Normally, a cincher system 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 3-3. It is observed that an ac voltage is applied to the rotor of the CX and that the wye-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 Vout(t) is dependent on the relative angle 0 between the rotors of the CX and CT. In particular, this output voltage is

$$V_{out}(t) = V_m \sin\theta \sin\omega_{act} \qquad (3\text{-}1)$$

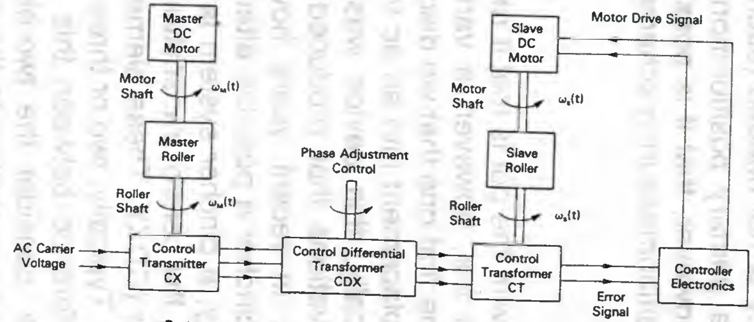


Figure 3-4 An example of a servo using a three element synchro system. To maintain a uniform product (e.g., steel sheets), the slave roller's speed $\omega_s(t)$ must be synchronized to that of the master, i.e., $\omega_M(t)$. The CDX is used to provide the desired angular relationship between the master 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.

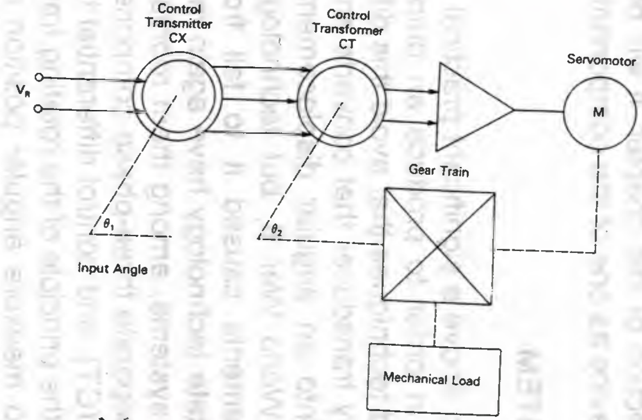


Figure 3-5 A synchro used in a position servo loop. The desired angular position is θ_1 whereas θ_2 is the actual angular position of the motor shaft. [V_R is the ac reference (carrier) voltage]. (Redrawn with permission of Walter Lewis, ILC Data Device Corp., Bohemia, NY. From the *Synchro Conversion Handbook*, p. 10, 3rd printing, 1982.)

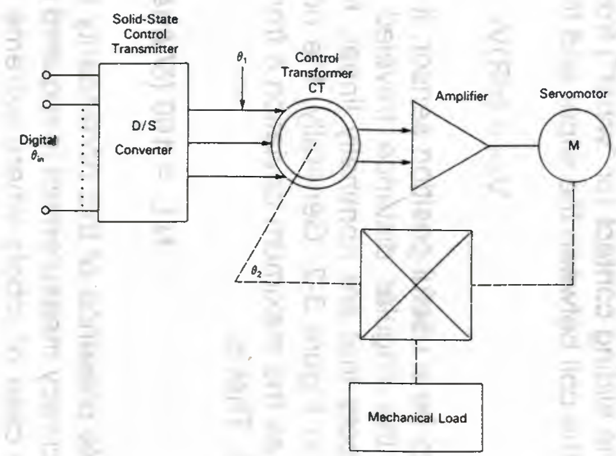


Figure 3-6 The synchro control transmitter CX of the position servo shown in Fig. 5.2.8 is replaced by a D/S converter. This scheme permits the desired input θ_{in} to be a digital quantity, i.e., makes the system microprocessor compatible. (Redrawn with the permission of Walter Lewis, ILC Data Device Corp., Bohemia, NY. From the *Synchro Conversion Handbook*, p. 11, 3rd printing, 1982.)

Where V_m and ω_{ac} are, respectively, the amplitude and radian frequency of the reference (or "carrier") ac voltage. Those readers familiar with elementary communications theory will recognize that Eq. (3-1) 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 the 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. (3-1) and Figure 3-3, 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 0 if $-70° < \theta < 70°$. It is for this reason that where a linear relationship between output and angular position is important, the synchro must be used about an operating point of $\theta = 0°$.

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 3-4. 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 3-5. It is observed that the command or input (i.e., the angle θ_1) will produce a command voltage from the CX. The CT will then produce an error voltage in accordance with Eq. (3-1), where $\theta = \theta_1 - \theta_2$. This error signal is amplified and causes the servomotor to rotate until θ is again zero. In such an application, the two-element synchro provides a rugged, reliable, and costeffective method of 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 3-6, replaces the CX in the position servo of figure 3-5. 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 θ_1.) The CT once again acts as a position error sensor and the system

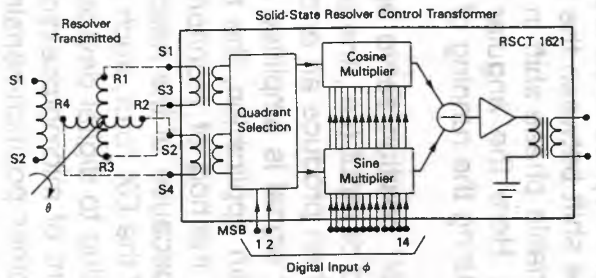


Figure 3-9 Resolver transmitter and RSCT 1621 Solid-State Resolver Control Transformer functional diagram. Use of this hybrid device permits elimination of separate input and output transformers. θ is the measured or actual angular position and φ is the desired angular position. (Redrawn with permission of Analog Devices, Inc., Norwood, MA. From *Synchro and Resolver Conversion*, Fig. 5-11, p. 110. © 1980 Memory Devices Ltd., Surrey, UK.)

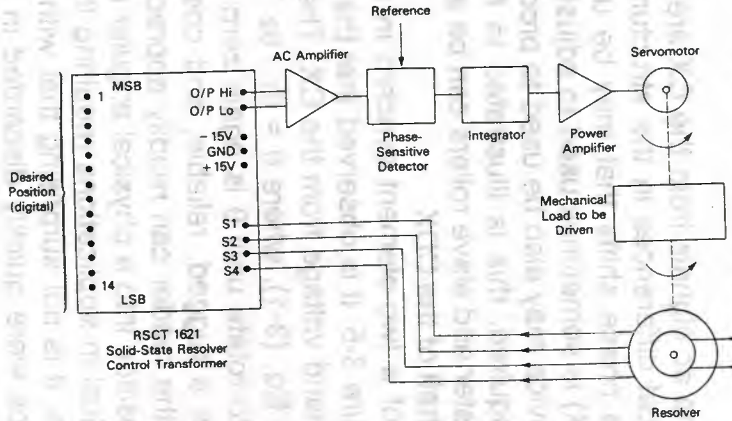


Figure 3-10. A position servo that uses a D/R converter (i.e., RSCT 1621). An ac amplifier is used because the D/R output is an ac voltage. (Redrawn with permission of Analog Devices, Inc., Norwood, MA. From *Synchro and Resolver Conversion*, Fig. 5-12, p. 110. © 1980 Memory Devices, Ltd., Surrey, UK.)

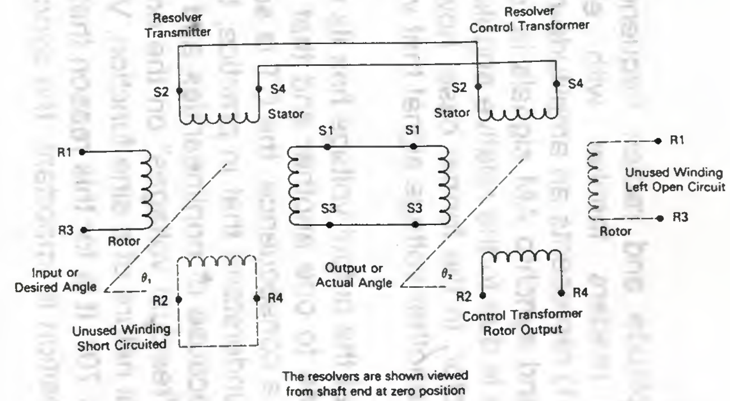


Figure 5.2. 3-8 esolver transmitter connected to a resolver control transformer. (Redrawn with permission of Analog Devices, Inc., Norwood, MA. From *Synchro and Resolver Conversion*, Fig. 1-13, p. 8, © 1980, Memory Devices, Ltd., Surrey, UK.)

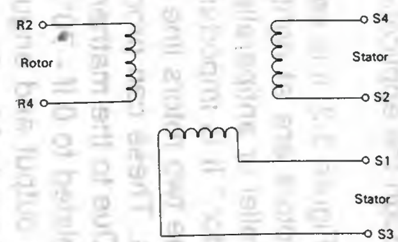


Figure 3-7 Electrical circuit of a simple resolver. With an ac carrier voltage input to the rotor, the output voltage amplitude of the two stator windings will be dependent on the sine or cosine of resolver shaft angle θ. (Redrawn with permission of Analog Devices, Inc., Norwood, MA. From *Synchro and Resolver Conversion*, Fig. 1-11, p. 7 © 1980, Memory Devices Ltd., Surrey, UK.)

behaves in a manner that is identical to that of the one in figure 3-5. The use of the D/S converter produces a position servo that is part digital and part analog.

3.3.1 Resolvers

The resolver is actually a form of synchro and for that reason is often called a "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 3-7. With the rotor excited by an ac carrier voltage B sin $\omega_{ac}t$, the two stator voltages become

$$V_{1\text{-}3}(t) = V\sin\theta\sin\omega_{ac}t \qquad (3\text{-}2)$$

$$V_{2\text{-}4}(t) = V\sin\theta\sin\omega_{ac}t \qquad (3\text{-}3)$$

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

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. (3-2) and (3-3) 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 3-4. Figure 3-8 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., $\theta_1 - \theta_2$). It is important to understand that although angular position can be monitored using a single resolver [see Eqs. (3-2) and (3-3)], 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 3-9 can be used in place of an RT. As can be seen, a 14-bit digital representation of a command input ¢ and the analog output of an RX. representing the actual angle fig are input to the D/R converter. The output of this device is then an analog voltage that is proportional to $\theta - \varphi$. 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 appropriate input and output transformers. The only significant difference between a D/R and a D/S converter is in the transformer configurations.

A position servo utilizing such a chip is shown in figure 3-10. Note that since the output of the D/R converter (or DRC) is an ac voltage, it is necessary to use an ac

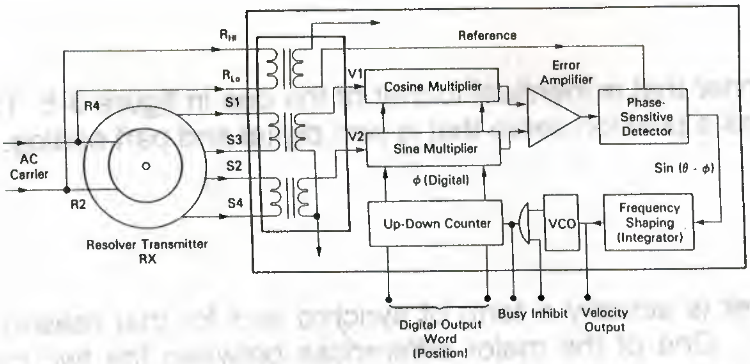
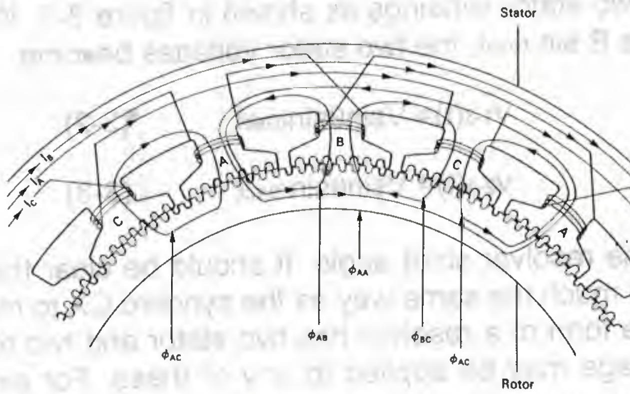
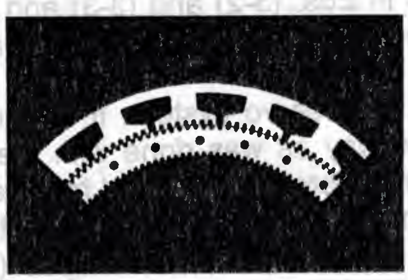


Figure 3-11 A tracking resolver-to-digital converter. The RX senses the actual position θ. The chip outputs a digitized version of this angle. The velocity is $d\theta/dt$ and is an analog quantity. (Redrawn with permission of Analog Devices, Inc., Norwood, MA. From *Synchro and Resolver Conversion*, Fig. 3-3, p. 46, © 1980 Memory Devices, Ltd., Surrey, UK.)



(a)



(b)

Figure 3-12. Motornetics resolver: (a) sketch showing current and flux paths. (Redrawn with permission of B. Powell and Motornetics Corp., a subsidiary of NSK, Santa Rosa, CA. From I.B. Cushing, "A New High Accuracy Angular Position Transducer" Proc. of Motor-Con, April, 1984, p. 284, Fig. 1.); (b) portion of actual device.)

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.

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 3-11 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. For example, if a 12-bit converter is used, a minimum angular change of 0.088° (360/22) in the resolver shaft will initiate a conversion. Note that unlike many A/D converters, there is no need to trigger the R/D externally.

Tracking synchro-to-digital (S/D) converters 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.

3.3.2 THE MOTORNETICS RESOLVER

As mentioned previously, a new type of motor with the trade name *megatorque* was introduced in the early 1980s. Capable of producing the extremely large torques required by direct-drive robots, the motor would have been less attractive in this application without the concurrent development of a high-resolution position sensor. Fortunately, such a sensor was developed by Motornetics Corporation [1,2]

As shown in Figure 3-12ab, in schematic cross section and actually appears when fabricated, this novel reluctance-based type of resolver has annular ring geometry and consists of a single multipole toothed stator with windings together with a toothed rotor without windings. In effect, the primary and secondary windings are combined so that all of the active magnetic area is utilized. This causes the sensor's accuracy to be improved and its signal level to be increased. In addition, it needs only a total of tour wires, which is an extremely important benefit in robot applications.

Although the stator and rotor of the Motornetics Resolver have the same number of teeth, tooth alignment varies in unison every third pole. This is accomplished by changing the mechanical phasing of the teeth of each pole (with respect to the immediate neighbors on either side of any tooth) by one-third of a tooth pitch. The reader can easily verify that such is the case from Figure 3-12a. Electrically, every third winding is connected in series so that the self- and mutual inductances (with respect to the other two phases) vary cyclically. The cycle repeats each time the rotor moves one complete tooth pitch. In this way the mechanical angle is equal to the electrical angle divided by the number of rotor teeth N.* For example, if N = 150, the device can be thought of as behaving like a standard resolver placed on the input side of a 150: 1 speed reducer since the electrical signal will go through 150 cycles for each mechanical

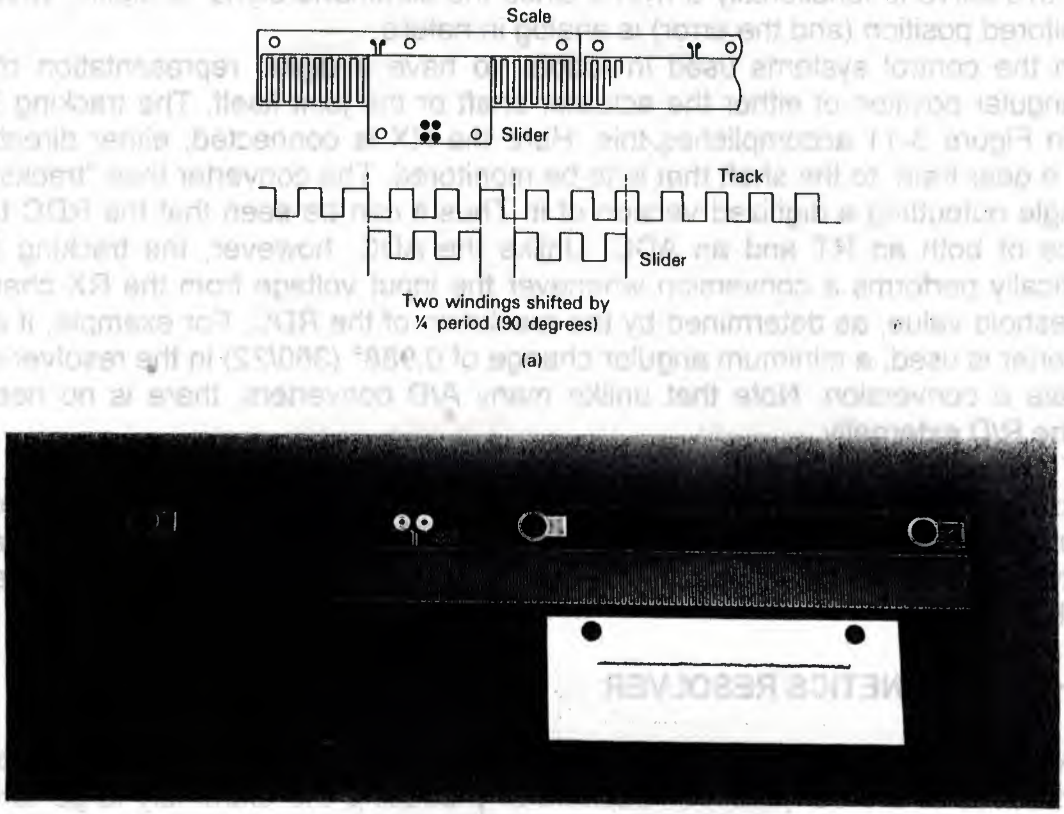


Figure 3-13 Linear Inductosyn: (a) sketch of slider and scale with windings shown magnified. (Redrawn with permission of Farrand Controls, a division of Farrand Industries, Inc., Valhalla, NY.); (b) photo of actual device. (Courtesy of Farrand Controls, a division of Farrand Industries, Valhalla, NY.)

revolution.

Although the Motornetics Resolver's three-phase nature makes it more closely resemble a synchro, electronic circuits are normally used to modify the signals so that more commonly available RDCs can be used to digitize the analog position information. A fairly inexpensive 10-bit RDC will produce an overall resolution of 153,600 (150 x 1024) "counts" per motor revolution. The corresponding number for a 12-bit RDC is 614,400. In either case, this is considerably greater than the resolution generally used by industrial robots of the mid-1980s (e.g., in the order of 40,000 to 60,00() counts/rev). However, as robot resolution requirements increase, it is clear that this sensor will be a candidate in certain applications.

It is important to understand that unlike the standard single-cycle resolvers described in the preceding section, the multiple-cycle device is an incremental position-sensing device rather than an absolute one. This means that when a robot utilizing such a sensor is powered up, the true position is unknown since the actual position is determined only within one cycle, but there is no way to know which cycle, of the possible N. is being sensed. The apparent difficulty is easily overcome by first causing the robot to execute a calibration procedure. For example, all joints may be driven (without regard to the position sensors' outputs) until they encounter mechanical end stops. Then the motors are reversed, causing the robot joints to "back away" a specified number of "counts" from these end stops. All digital position counters are then zeroed. To obtain absolute position information it is only necessary for the hardware to keep track of both the count and the cycle number, which can easily be done.

3.4 THE INDUCTOSYN

A device that is used extensively in numerically controlled machine tools is the Inductosyn, a registered trademark of Farrand Controls, Inc., which developed it. Acknowledged to be one of the most accurate means of measuring position, it is capable of accuracies of 0.1 mil linear or 0.00042° rotary.

In actual operation, the Inductosyn is quite similar to the resolver. Regardless of whether the configuration is linear or rotary, there are always two magnetically coupled components, one of which moves relative to the other. For example, consider the linear Inductosyn shown in Figure 3-13. The fixed element is referred to as a scale and the moving element as a slider. Both of these are fabricated using printed-circuit technology, which is one of the major reasons for the high degree of accuracy that is achievable. A rectangular-wave copper track having a cyclical pitch of 0.1, 0.2, or 2 mm is normally bonded to the substrate material. The scale usually has one continuous track that may be many inches long (e.g., 10, 20, or longer). The slider, on the other hand, is about 4 in. long and consists of two separate tracks of the same pitch as the scale but separated from one another by 4 of a period (or 90°). The slider is mechanically able to travel over the entire length of the scale, the gap between these two elements being about S mils. (An electrostatic screen is placed between them to prevent accidental short circuits due to externally applied forces.)

As in the case of the resolver. an ac voltage V sin e is applied to the scale. Here, however. The carrier frequency ($\omega_{ac}/2\pi$) is in the range 5 to 10 kHz. The output at the two-slider tracks is then

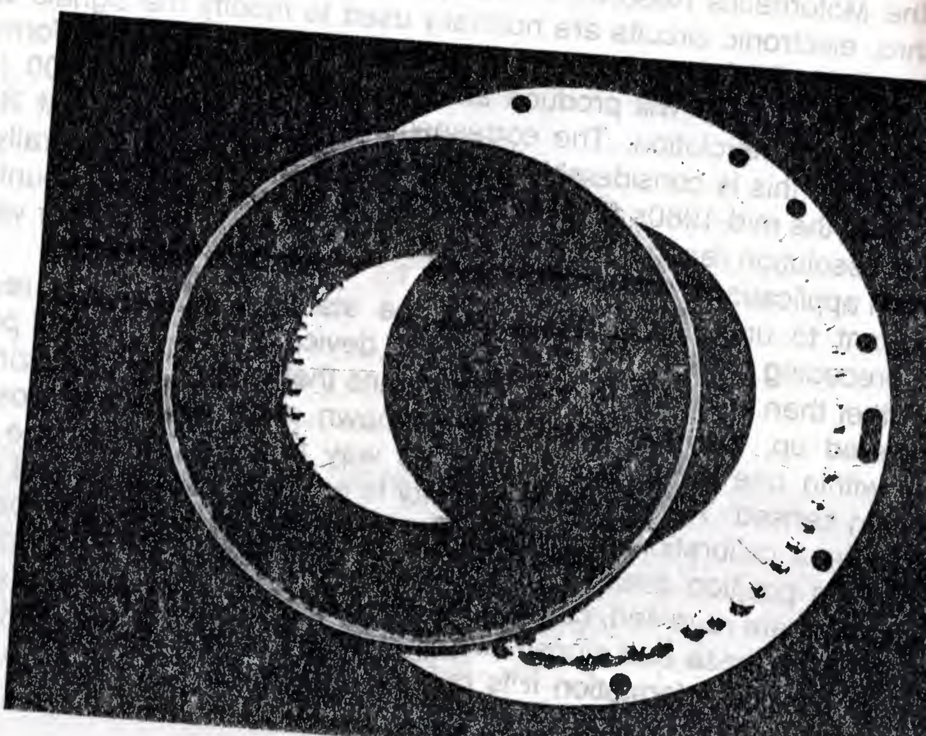


Figure 3-14 Rotary Inductosyn®. The stator corresponds to the slider of the linear Inductosyn®. The ac carrier voltage is applied to the rotor. (Courtesy of Farrand Controls, a division of Farrand Industries, Valhalla, NY.)

$$Vs1 = V \sin(2\pi X/S) \sin \omega act \qquad (3\text{-}4)$$

$$Vs2 = V \cos(2\pi X/S) \sin \omega act \qquad (3\text{-}5)$$

Where X is the linear distance along the scale and S is the wave pitch. The amplitude of the sinusoidally varying input voltage is modulated spatially in much the same manner as the resolver [e.g., see Eqs. (3-4) and (3-5)]. Unlike the resolver, however, this spatial variation repeats every cycle of the scale track. Moreover, since Eqs. (3-4) and (3-5) represent the average voltage across a number of poles (i.e., cycles) of the scale, any variations in the pitch and/or conductor spacing are minimized, again contributing to the high degree of accuracy achievable with the device.

In its rotary form, shown in Figure 3-14, the stator (surprisingly) corresponds to the slider of the linear Inductosyn. Two separate rectangular track waveforms are placed radially on a circular disk. Again there are separate sine and cosine tracks, which, because they alternate physically, permit most of the error due to spacing variations to be averaged out. As a consequence, the rotary Inductosyn is probably the most accurate means currently available for monitoring position in commercial machine tools. As mentioned previously, typical accuracies are in the order of ± 0.42 millidegrees. Note that although laser devices are capable of giving considerably higher accuracies, their excessive cost makes them unattractive for this type of application.

The rotor of the rotary Inductosyn corresponds to the scale of the linear device in that it has a single, continuous, and almost rectangular printed track. Typically, there are anywhere from 128 to 1024 cycles (or 256 to 2048 "poles") on the disk. Because of the rotary configuration, however, the ac input voltage is applied to the rotor using brushes and slip rings. (A brushless configuration is also possible.) The output voltage of the device is monitored across the stator and has the same form as that shown in Eqs. (3-4) and (3-5) except that $(2\pi X/S)$ is replaced by $N\theta/2$, where N is the number of poles of the rotor, and θ is the angle of rotation of the rotor with respect to the stator.

In actual operation, either form of Inductosyn can be used like a resolver. For example, one Inductosyn can act like a transmitter (RX) and the other like a receiver (RC) in a simple position servo. Alternatively, a resolver can be used as the RX and the Inductosyn as the RC. The advantage of the latter approach, however, is that one complete rotation of the resolver due to a position command signal will produce only a single cycle motion of the Inductosyn. Thus depending on the resolution of the latter device (i.e., the number of cycles per unit length over 360°), use of the Inductosyn would permit positioning of a machine tool to close tolerances. For example, a Emil linear resolution would not be unreasonable at all.

The configuration described above would be potentially attractive for lose in either prismatic or rotary joints of robots. However, gears or harmonic drives would still be acquired to obtain the torque multiplication from actuator to output Thus the added cost of the Inductosyn, together with the additional electronics needed to digitize its output signals, would probably make the Inductosyn less attractive than other position-monitoring sensors. However, if extremely high accuracies are required in the future, this device may someday be useful in the design of robots.

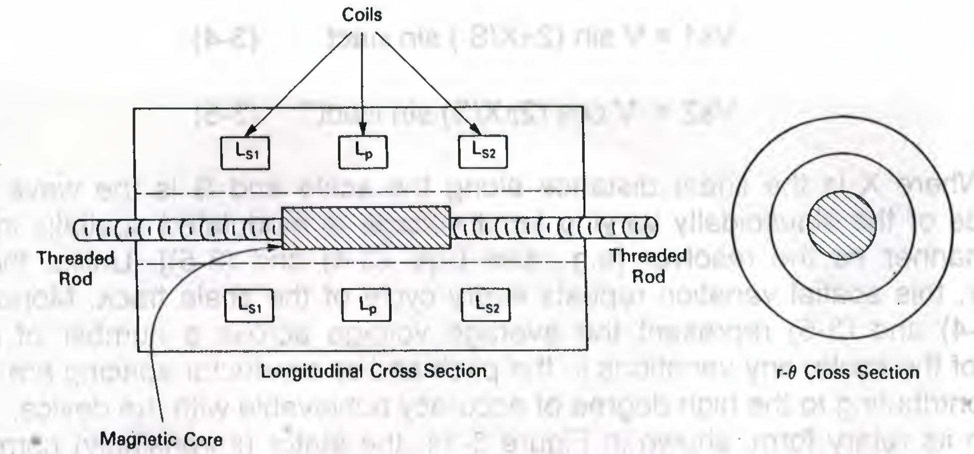


Figure 3-15 A linear variable differential transformer (LVDT) showing the single primary and the two sets of secondary coils. The magnetic core is generally the moving element of this sensor. (Redrawn with permission of Schaevitz Engineering, Pennsauken, NJ.)

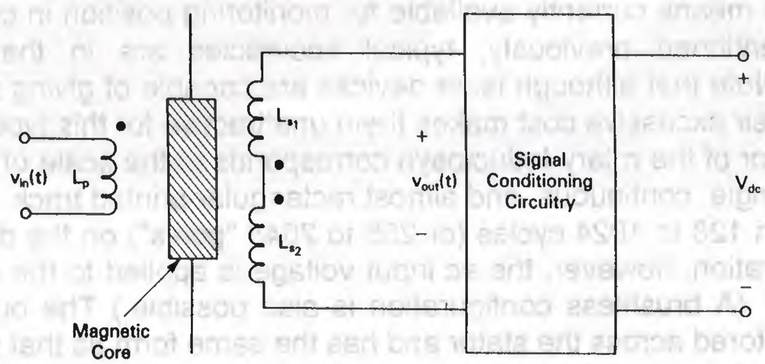


Figure 3-16 Electrical circuit of an LVDT showing the magnetic core. The secondary coils are connected in series opposing so that when the core is at or near the center of the LVDT $v_{out}(t)$ is zero. The signal conditioner is used to "demodulate" $v_{out}(t)$ and produces a dc voltage that is proportional to the core's linear distance away from the null (center) position. (Redrawn with permission of Schaevitz Engineering, Pennsauken, NJ.)

3.5 LINEAR VARIBLE DIFFERENTIAL TRANSFORMERS

Another device that is both extremely rugged and capable of accurate position determination is the linear variable differential transformer (LVDT; see Figure 3-15). 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 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 (e.g., a voltmeter).

A common design of the LVDT has three equally spaced coils (L_p, L_{s1}, and L_{s2}) on a cylindrical coil form (see Figure 3-15). 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 the flux linking the three coils.

To understand the operation of the LVDT, we consider the equivalent electrical circuit of the device shown in Figure 3-16. As can be seen, an ac voltage is applied to L_p, the primary side of the coil structure (this corresponds to the center coil in Figure 3-15). Since L_{sl} and L_{s2} on the secondary side are connected in series opposing (note the position of the dots on the windings), $V_{out}(t)$ 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.

If, however, the core is moved away from the central position, the coupling between L_{sl} and L_p will differ from that of L_{s2} and L_p. For example, the former will increase, whereas the latter will decrease. Consequently, the voltage induced in L_{sl} and L_{s2} will increase and decrease, respectively, with respect to their center core values. Thus $V_{out}(t)$ will be nonzero.

3.6 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.

3.6.1 Opto-Interrupters

It will be recalled that point-to-point-type robots require only that the beginning and end points be accurately. 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

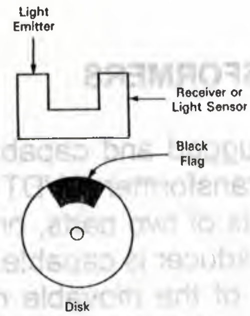


Figure 3-17 Simple opto-interrupter showing light emitter-receiver assembly and disk with "black flag."

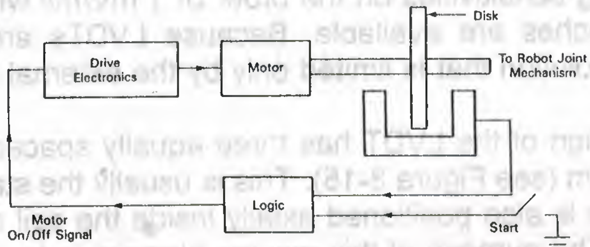


Figure 3-18 Block diagram of a simple unidirectional motor control circuit. The motor begins to rotate when the switch is closed.

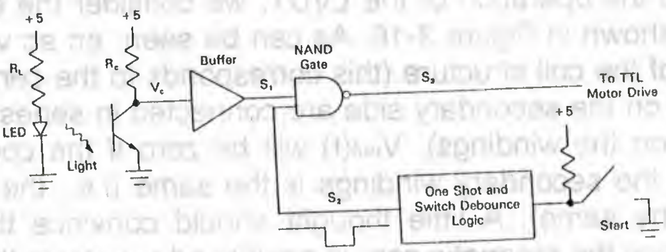


Figure 3-19 Possible realization of sensor and logic circuits for simple motor controller of Fig. 5.3.2.

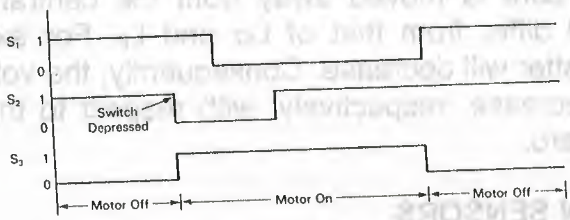


Figure 3-20 Timing diagram for the digital signals associated with the sensor and logic circuit of Fig. 5.3.3.

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. (It is used in educational-type robots, however.)

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 3-17. 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 and 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.

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 3-18. 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. (If desired, additional circuitry can be added to produce dynamic braking, thereby stopping the motor much more quickly.)

A possible realization of the logic and sensor electronics is shown in Figure 3-19. The waveforms of the digital signals S_1, S_2, and S_3 are shown in Figure 3-20. 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 (S_1 and S_2 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 of the phototransistor will be about 5 V, so that S_1 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, S_2 will normally be high also. Therefore, the signal to the motor drive circuitry is low and the motor does not turn.

As seen in Figure 3-20, when the start switch is depressed, S_2 goes low, which in turn causes S_3 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 phototransistor. 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 bidirectional 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.

3.6.2 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: (l) 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.

3.6.3 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.

3.6.4 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

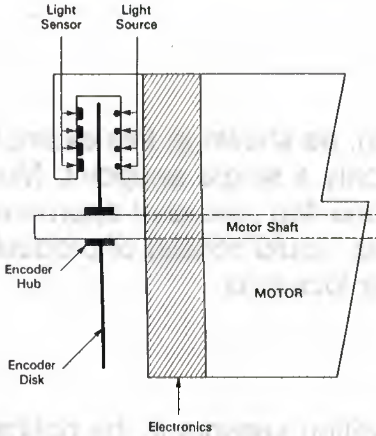


Figure 3-21 An absolute encoder mounted on a motor. Shown are the various components of the encoder including the disk, light sources, light detectors, and electronics.

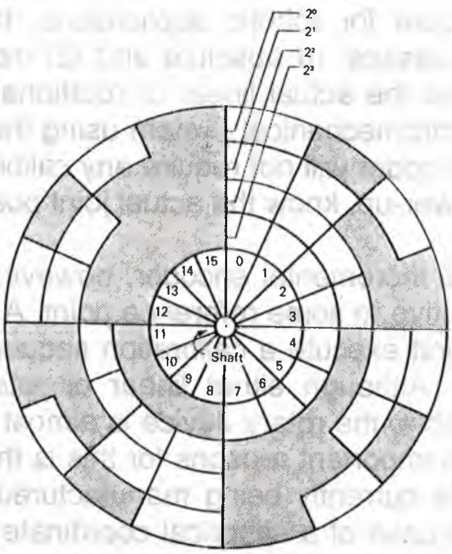


Figure 3-22 A four-track, sixteen-sector pure binary-coded disk used in an absolute encoder. (Redrawn with permission of Dynamics Research Corp., Wilmington, MA. From "Techniques for Digitizing Rotary and Linear Motion," Fig. 3-1, p. 3-2, 1976.)

Figure 3-23 An incremental encoder disk having 200 radial lines. A resolution of 1.8° is possible with such a device.* (Courtesy of Dynamics Research Corp., Wilmington, MA.)

Normally, light emanating from a linear, N-element light source (e.g., LEDs) is made to pass at 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 3-21). 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 3-22 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 3-22 that the resolution of the disk is 22.5° (360/16) since one complete disk revolution is 360° and there arc 16 sectors. 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 3-22, 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 photodetector outputs.

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

3.6.5 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 of machine 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. This is usually not considered to be a major disadvantage, since such an operation generally has to be executed only after power has been applied. It is important to understand that if power is accidentally lost during an operation, calibration must be performed again since the incremental encoder has no "memory."

Just as in the case of the absolute device, the incremental encoder in its simplest form consists of a disk, an LED light source, and a corresponding set of light receivers (e.g., phototransistors). However, there are significant differences between the two. For example, there are usually only a single LED and four photodetectors. Also, the thin circulator disk (usually made of glass, Mylar, or metal) contains a single track consisting of N radial lines, as shown in Figure 3-23. The resolution of an encoder containing such a disk is normally defined as the number of lines, N. This implies that the encoder can resolve an angular position equal to 360°/N. Typically, encoders with resolutions of 100,

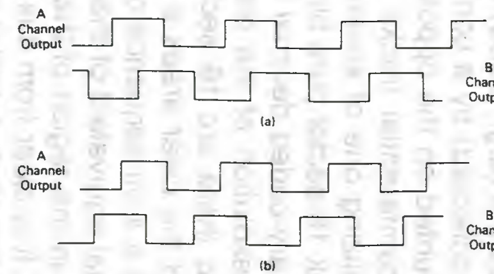


Figure 3-24 TTL outputs of the A and B channels of an incremental encoder: (a) A leads B when clockwise rotation occurs; (b) A lags B when counterclockwise rotation occurs.

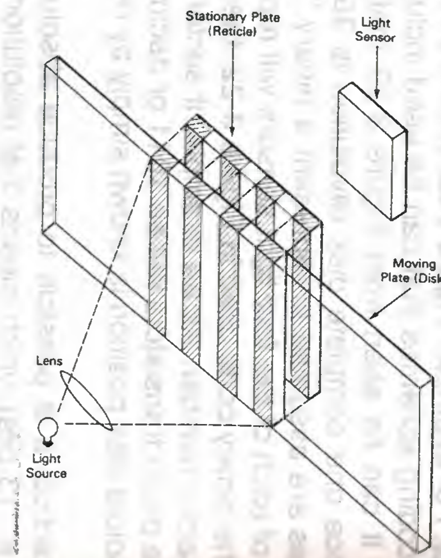


Figure 3-25 Section of a single channel encloder. (Redrawn with permission of Dynamics Research Corp., Wilmington, MA. From "Techniques for Digitizing Rotary and Linear Motion," Fig 4-1, p. 4-2, 1976.)

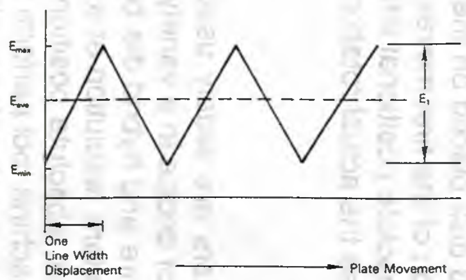


Figure 3-26 Output voltage of the sensor in Fig. 5.3.9 as the encoder disk moves relative to the reticle. (Courtesy of Dynamics Research Corp., Wilmington, MA. From "Techniques for Digitizing Rotary and Linear Motion," Fig. 4-2, p. 4-3, 1976.)

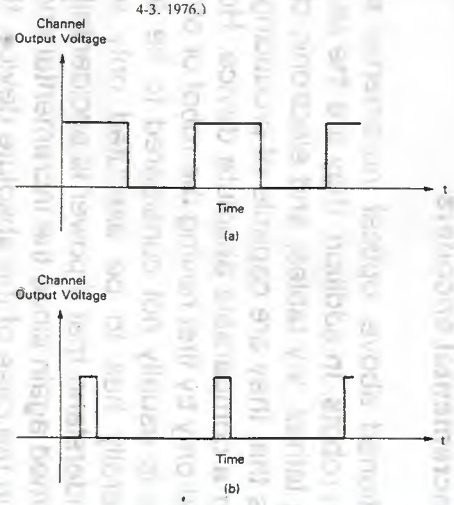


Figure 3-27 TTL output of a single encoder channel. The disk is assumed to be rotating with a constant angular velocity: (a) ideal 50% duty cycle; (b) the duty cycle is not 50% due to drift in the average value of the sensor voltage E_{ave}.

128, 200, 256, 500, 512, 1000, 1024, 2000, and 2048 lines are available, meaning that angular resolutions ranging from 3.6° down to 0.175° are achievable. Generally, in robot applications, 200- to 1000-line disks are quite adequate, even where it is necessary to position a part or tool to within + l or 2 mils. We will shortly see that it is possible to increase this resolution electronically. However, before discussing this, let us describe how the incremental encoder produces positional information.

If the encoder disk is mounted on a rotating shaft (e.g., of a servomotor as shown in Figure 3-21), 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 (e.g., more than 5000 rpm).

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 (see Figure 3-24).

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 3-25. 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 3-26. 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 E_{max} is proportional to the intensity of the LED. The minimum voltage Emit is not zero because light cannot 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 50% duty cycle provided that the disk is rotating at a constant velocity (see Figure 3-27a). However, if E_{ave} drifts due to LED

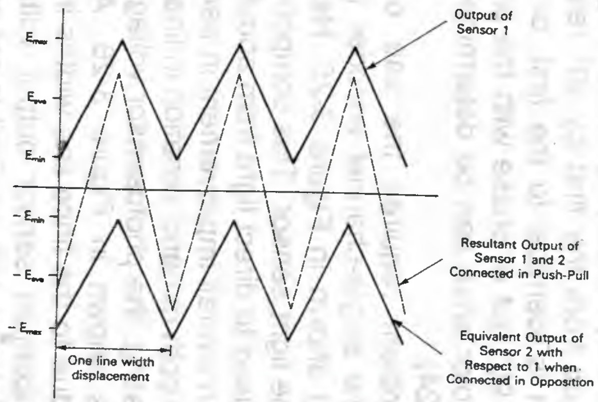


Figure 3-29 Push-pull output from two sensors. This arrangement significantly reduces the effect of average value drift on the TTL duty cycle. (Redrawn with permission of Dynamics Research Corp., Wilmington MA. From "Techniques for Digitizing Rotary and Linear Motion," Fig 4-6, p. 4-6, 1976.

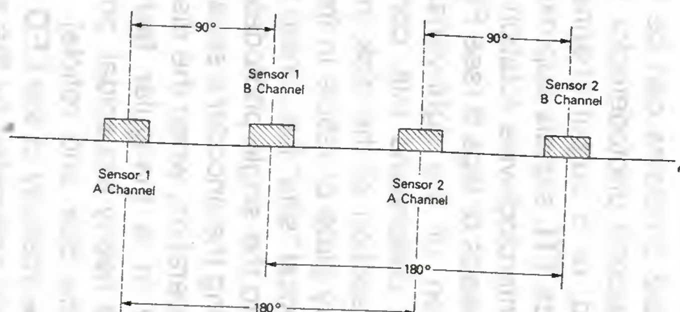


Figure 3-30 Two sets of sensors, placed 90° apart, are used to determine the direction of rotation.

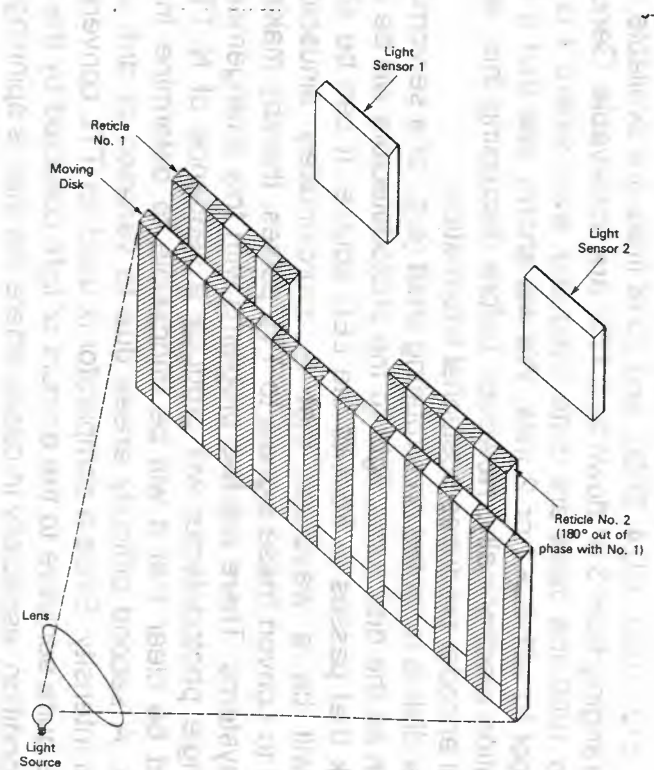


Figure 3-28 Use of two sensors with the *same* light source to reduce the variation in the average value of the encoder's output, E_{ave}. (Redrawn with permission of Dynamics Research Corp., Wilmington, MA. From "Techniques for Digitizing Rotary and Linear Motion," Fig 4-4, p. 4-5, 1976.)

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 3-27b. 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).

This problem can be overcome by employing a second sensor (and reticle) placed 180° out of phase with the first, as shown in Figure 3-28. Note that the same light source is used to illuminate both sensors. If the outputs of the two photodetectors are connected in "push-pull" so that the two signals are subtracted, a triangular waveform centered about zero and having approximately twice the peak-to-peak amplitude of either signal will be generated (see Figure 3-29). In practice, differences in the two sensors cause the average value to differ somewhat from zero. 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 sensors 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.

As mentioned above, the single sensor encoder cannot give any information about the direction of rotation. A little thought should convince the reader that the use of a second photodetector placed 180° out of phase with the first one does not alter this situation. The encoder obtained is still a single-channel device. To determine direction, a second set of photosensors, placed 90° out of phase with the first set, must be used as shown in Figure 3-30. Here the (push-pull) output of the first set becomes the A channel, whereas that of the second is channel B. A typical two-channel output is shown in Figure 3-31. Using the same convention as in Figure 3-24, the situation in Figure 3-31, would represent clockwise (CW) disk rotation. Note, however, that such an assignment is arbitrary, and therefore one could just as easily consider "A leading B" to be a counterclockwise (CCW) rotation. Which definition is used is unimportant, but consistency must be maintained.

3.7 VELOCITY SENSORS

As noted that, 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.

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.

3.7.1 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, 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 underlying principle 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 3-32. 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 (e.g., 11) 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

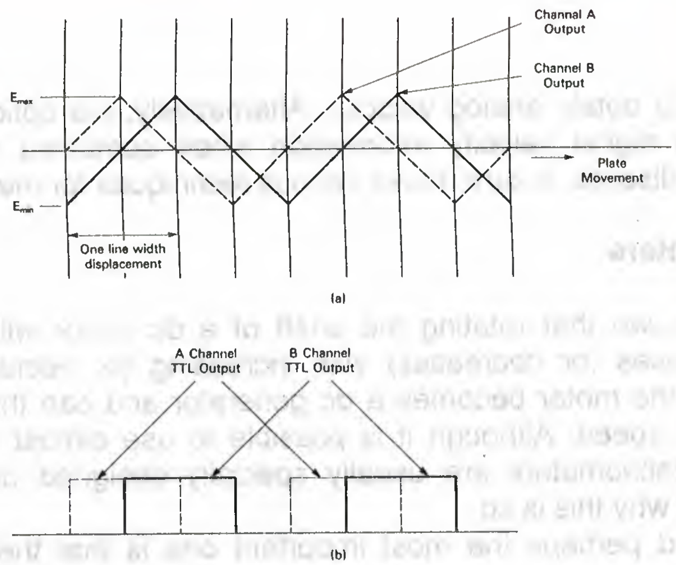


Figure 3-31. Two channel, incremental encoder outputs: (a) "raw," triangular signals; (b) digital signals. A channel leads B. [Part (a) redrawn with permission of Dynamics Research Corp., Wilmington, MA. From "Techniques for Digitizing Rotary and Linear Motion," Fig. 4-7, p. 4-7, 1976.]

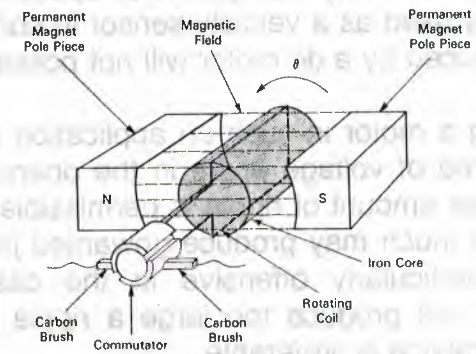


Figure 3-32 Analog tachometer showing one coil (of many) mounted on an iron core.

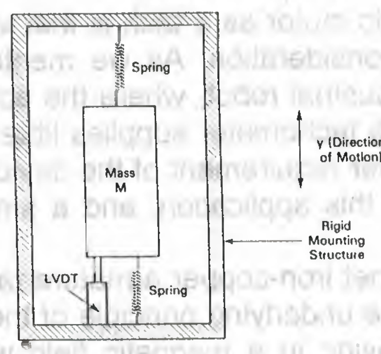


Figure 3-33 Basic elements of a linear accelerometer. The LVDT is used to monitor the relative displacement of the lower spring, a quantity that is proportional to the acceleration.

brushes which touch the various segments of the commutator.

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 do 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.

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 (e.g., 19 to 23). 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 I YG 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.

3.7.2 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, as describeds, 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.

3.7.3 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 A. Here, At is the "sampling" interval (or period) and Sax 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 5.6.3 in block diagram form. This unit will produce a 0- to 5-V output for pulse repetition rates of dc to 1() 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 large bandwidths, this delay can create stability problems which must then be dealt with using additional 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.

3.8 ACCELEROMETERS

Besides monitoring the position and velocity of a physical system, it is also possible to monitor its acceleration. Normally, linear acceleration is measured, whereas angular acceleration is most often derived from angular velocity by differentiation. Let us consider briefly how a device that can be used to obtain linear acceleration, and referred to as an accelerometer, operates.

From Figure 3-33, it can be seen that an accelerometer consists of three basic elements:

1. A mass M

2. Some type of linear displacement sensor (e.g., an LVDT)

3. A set of springs having an equivalent spring constant K

Based on one of Newton's laws (i.e., F = Ma, where F is the force needed to accelerate the mass (M) a linear units per second per second), it is easy to understand the operation of this device. Suppose that the entire accelerometer begins to move (i.e., is accelerating) in a downward direction. The force required to do this (e.g., Ma) will be opposed by the springs supporting M. as they bend upward a distance y, this force will be equal to Ky. Thus

$$Ma = Ky$$

Solving for the linear acceleration a gives

$$a = K/My$$

From this equation it is apparent that the acceleration of the mass is Proportional to the distance. If an LVDT is used to determine linear position, as is often the case in commercial accelerometers, the output of that sensor will be proportional to the actual acceleration. Thus the signal-processing device used in an LVDT and described previously can be made to read acceleration directly.

Not mentioned in the discussion above is the fact that to be a useful sensor, the accelerometer must also include enough damping so that the spring-mass combination does not "ring" significantly (i.e., produce damped sinusoidal oscillations). Normally, it is desirable to have a small amount of overshoot of the final displacement position so that a damping constant of 0.6 or 0.7 is used. Under these conditions, an accelerometer can monitor motions having frequency components that are at least 2.5 times lower than the undamped (or "natural") frequency of the second-order mechanical system composed of the mass, spring, and damping.

Although commercial accelerometers can be obtained that will measure accelerations ranging from ± S to thousands of g's,* these devices have had limited use in robots currently being manufactured. One reason is that, as mentioned above, it is usually possible to determine only linear acceleration directly. Since most robot joints are revolute rather than prismatic in nature, it has been deemed more useful to measure position and/or velocity of the joint directly. If necessary, the acceleration signal can be derived from these data. Another reason for not using acceleration is that even if it were easy to measure this rotational quantity directly, it would still be necessary to process the information so as to get the desired position and velocity data to control the robot joint, that with robots, we are dealing with a position servo so that it seems to make more sense to monitor the position directly rather than indirectly (i.e., by performing two integrations on the acceleration signal). Certainly, one would expect fewer errors and/or uncertainties using the direct approach.

There have been a few experiments with accelerometers and robots, however. In most cases, the sensor has been used together with standard encoders to provide an estimate of the actual motion of the joint being controlled. It will be recalled that most position sensors are mounted on the actuator output before the rotary motion is geared down, in order to achieve the desired position resolution. The assumption is made that if the actuator's motion is controlled, the joint will respond in an identical manner. Clearly, this is not always true because mechanical linkages and/or couplings are not perfectly rigid. Generally, mechanical resonances that occur as a result cannot be compensated for by sensors so placed. However, an accelerometer mounted on the joint structure can provide information about what the joint is actually doing. These data, together with those from the actuator position sensor, can be possessed to compensate partially for the nonideal motion of a robot axis. The accelerometer is thus used to "observe" the actual joint behavior. In fact, at least one commercially manufactured robot utilizes a single axis accelerometer on its prismatic joint to reduce undesirable arm oscillations caused, in part, by imperfect mechanical linkages. Readers interested in learning more about the use of such sensors are encouraged to read the references at the end of this chapter, where linear optimal estimation theory is employed in an attempt to improve the performance of a robot [3, 4].

Recently the Pennwalt Corporation of King of Prussia, Pennsylvania, has begun to market a piezoelectric polymer (i.e., PVDF)-based accelerometer. This device has been shown to be far more rugged and sensitive than the more traditional devices. Moreover, its cost is about an order of magnitude less than the industry standard unit with virtually the identical frequency response.

3.9 PROXIMITY SENSORS

Up to this point, we have discussed the behavior and application of sensors that were used to measure the position, velocity, or acceleration of robot joints (or more accurately. their actuators) and were called collectively internal state sensors. A second major class of robotic sensor is used to monitor the robot's geometric and/or dynamic relation to its task. Such sensors are sometimes referred to as external state sensors. Machine or robotic vision systems represent an important subclass of this group of devices and are treated separately in Chapter 6. The remainder of the current chapter is devoted to non-vision-type sensors that either can be or have already been used to

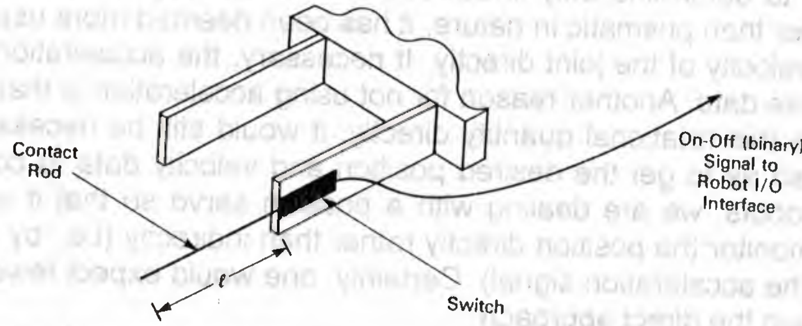


Figure 3-34 Simple contact rod proximity sensor mounted on one "finger" of a robotic gripper.

make the robot more aware of its external environment. Although some of these may utilize optical techniques as part of their sensing system, they are not properly classified as visual sensors, so we describe them in this chapter. Also, as will be seen, many arc still under development in various research facilities and are therefore not ready for use in an actual manufacturing environment. Since it is not possible to treat this subject in an exhaustive manner, readers desiring more information are referred to the references at the end of the chapter. Of particular note is an excellent summary report by D. J. Hall [5] of Carnegie-Mellon University's Robotic Institute which was quite helpful in preparing much of the remainder of this chapter.

In this section we describe a number of sensors used to tell the robot when it is near an object or obstruction. This can be done either by using a contacting or a noncontacting technique. Often, such sensors are called proximity devices, but the distinction between proximity and touch and/or slip is not clear-cut. That is, some proximity devices can also be used as touch (or tactile) sensors. We will consider only the proximity feature here, deferring the discussion of their application to touch and/or slip detection to a subsequent section.

3.9.1 Contact Proximity Sensors

The simplest type of proximity sensor is of the contacting variety. As Figure 3-34, shows, such a device consists of a rod that protrudes from one end and a switch or other linear position-monitoring element located within the body of the sensor. As the robotic manipulator moves, the sensor will become active only when the rod comes in contact with an object or an obstruction. When this occurs, the switch mounted inside the sensor will close (or open, if that is more convenient). The change of state of the switch, monitored through the robot's I/O interface, will cause an appropriate action to take place. Examples include an immediate (or emergency) halt if the device is used to sense obstacles or the branching to another part of the robot's program, thereby causing a particular operation to be performed (e.g., closing of the gripper). Such contact monitors can be placed anywhere on the robot's arm and/or wrist, and it is possible to utilize more than one. Thus simultaneous obstacle and object sensing is possible.

If the simple on-off switch is replaced by one of the linear position-sensing devices described in an earlier portion of this chapter, the "binary" contact proximity sensor becomes one that can detect actual position of the object (or obstacle). For example, a simple pot or an LVDT can be employed. Then once the protruding rod makes contact, further motion of the manipulator will push the rod into the sensor. If this rod is attached to the magnetic core of an LVDT or the wiper of a potentiometer, the motion will be converted into a voltage that will be proportional to the actual distance of the end of the rod (and hence the object) from some reference point on the robot (e.g., the end of the gripper). In addition, the approach velocity can be obtained from this information by performing either an analog or digital differentiation. Thus both distance and approach velocity can be monitored using such a contact sensor.

It is important to understand that a single contact proximity device cannot provide any information about the shape or nature of the object or obstacle (i.e., no object recognition capability is possible.* It will be seen in a later section, however, that some

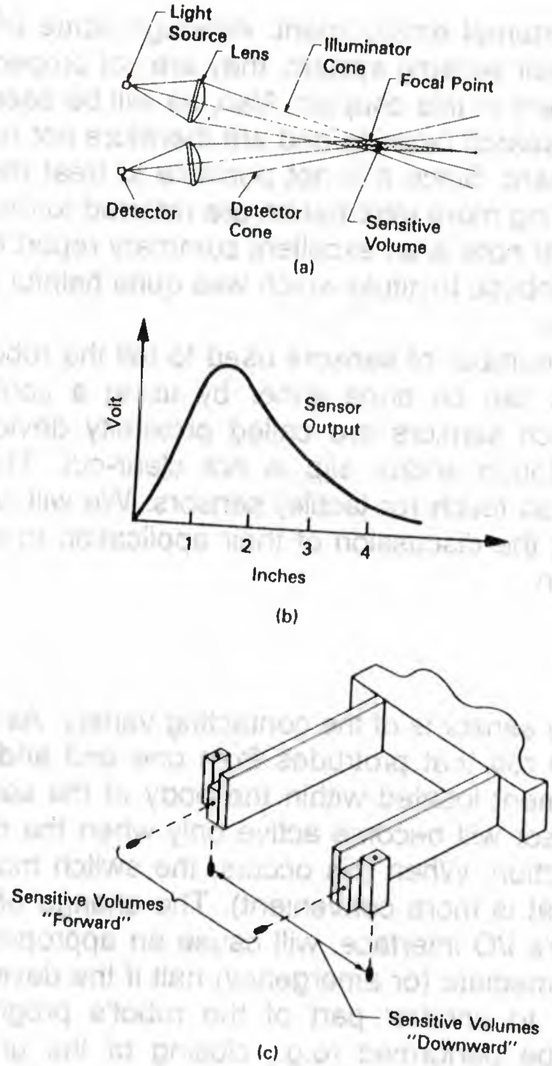


Figure 3.35 Reflected light sensor. (a) light source-detector assembly; (b) detector output voltage as a function of distance; (c) two-dimensional sensor array mounted on a parallel jaws gripper. (With permission of A.K. Bejczy and Jet Propulsion Laboratories, Pasadena, CA. From reference [6].)

degree of object recognition can be obtained by utilizing arrays of these devices.

3.9.2 Noncontact Proximity Sensors

In contrast to the devices described above, a much larger class of proximity sensor does not require any physical contact at all in order to produce a signal that can be used by a robot to determine whether it is near an object or obstacle. These noncontact devices depend on a variety of operating principles in order to make the proximity determination. For example, reflected light, ultrasound, or variation in capacitance, inductance, or resistance have all been used. We now briefly describe a number of such sensors.

3.9.3 Reflected light sensors

One of the simplest types of proximity sensors that uses light reflected from an object and has been used experimentally on a robot gripper is shown in Figure 3-35a. The sensor consists of a source of light and a photodetector separated by about 8 mm and tilted symmetrically toward one another. This, together with lenses mounted in front of the assembly, produces focused incident and reflected beams. Figure 3-35b shows the photodetector voltage as a function of object distance from the (detector) lens. Figure 3-35c indicates that several of these sensors can be placed on a robotic gripper. In this way, proximity in several directions can be monitored simultaneously (e.g., ahead and below the robot's hand).

Although the maximum sensor output will occur when an object (or obstacle) is at the focal point, Figure 3-35b reveals a basic difficulty with this device. That is, two different object positions produce the same voltage except when the object is located exactly at the focal point. Since a one-to-one correspondence between position and detector voltage does not exist, additional logic or hardware is required to eliminate the ambiguity. For example, if the robot is moving and the sensor signal is increasing, it is clear that the object is on the fur side of the focal point (i.e., has yet to reach this point, and so the output Corresponds to the larger of the two position values). If, however, the signal is decreasing, the focal point has been passed and the smaller distance should be used. Several sensors placed at angles can also be utilized to eliminate this ambiguity. In addition, Jet Propulsion Labs (JPL) has embedded fiber optic filaments inside the fingers of the robot so that the effective voltage characteristic of the sensor is monitonically decreasing with distance.

Besides this difficulty, other problems with the sensor exist. For example, ambient light will shift the curve in Figure 3-35b up or down depending on the intensity. The problem has been solved at JPL by pulsing the light source at a 6kHz rate. However, a more difficult and perhaps impossible problem to overcome is that the sensor is sensitive to the reflectivity of the object or obstacle. A highly reflective surface will obviously produce a larger output voltage than one that is less reflective. Thus it might be necessary to "calibrate" the sensor to each object so that the maximum output voltage could be found. Then, knowing the characteristic of the detector, position could be determined relative to this maximum value. Alternatively, careful control and/or an a priori knowledge of the surface reflectivity would be required.

Even with these measures implemented, it would still be difficult to use the sensor for absolute position monitoring. The major reasons are that the device would be quite sensitive to variations in light-source output, drift in the detector characteristics (due to ambient-temperature fluctuation), and environmentally caused changes in the reflectivity of the object. Thus all that could be reasonably expected would be to sense the proximity of an object to the robot's gripper within a band of distance. A threshold detecting circuit might be utilized to achieve this.

It is rather disheartening to realize that what at the outset appeared to be a simple and ideal noncontact proximity sensor has such great problems associated with it that its application to robots is not likely. It is for this reason that other devices are normally used to monitor proximity in a manufacturing environment.

3.9.4 Fiber optic scanning sensors

Fiber optics have been used to develop several different types of noncontact proximity sensors. As reported by Fayfield [7], there are at least three systems for utilizing this important technology in the robotics and/or the manufacturing fields. With regard to Figure 3-36, one employs transmitted light, Whereas the other two make use of reflected light. It is important to understand that it is not possible, in all cases, to obtain reliable absolute position information. The devices can only ten whether or not a part is present.

In the opposed or beam break configuration (Figure 3-36a), the object is detected when it actually interrupts the beam of light. Such an optical interrupter depends on the object being opaque and is, obviously, not useful where parts are made of transparent or translucent materials. By employing high-gain amplifiers and noise-reduction schemes, these sensors can detect objects as close as a few mils and as far away as several inches. However, they are limited to informing the robot that something is or is not present. That is, absolute position information cannot be obtained. In addition to this, the receiver fiber bundle alignment is fairly critical. Thus, anything that would tend to misalign it from the emitter bundle would obviously affect the sensor's effectiveness. Finally, it would be necessary to use units with different gaps and/or lengths, depending on the type and size of the object to be sensed.

A second type of fiber optic proximity sensor is referred to as a retroreflective device since it employs a reflective target placed some distance from the body of the unit (see Figure 3-36b). An opaque object entering the area between the end of the fiber bundle and the target is sensed since reflected light reaching the receiver is considerably reduced in intensity. This is also true for parts made of translucent materials because the incident beam of light and that reflected from the target are both attenuated when they pass through such an object. The use of thresholding circuits on the receiver side of the sensor is important in both of these cases. The retroreflective scheme utilizes a bifurcated fiber bundle so that incident and reflected light is carried by the same set of fibers. Clearly, this eliminates potential alignment difficulties associated with the previous technique. However, the need for a separate target somewhat restricts the use of such a sensor to parts detection only. It is clear that unless an unforeseen (and unpredicted) obstacle happens to disrupt the light from the reflecting target, it will not be sensed by a retroreflective mode fiber optic sensor.

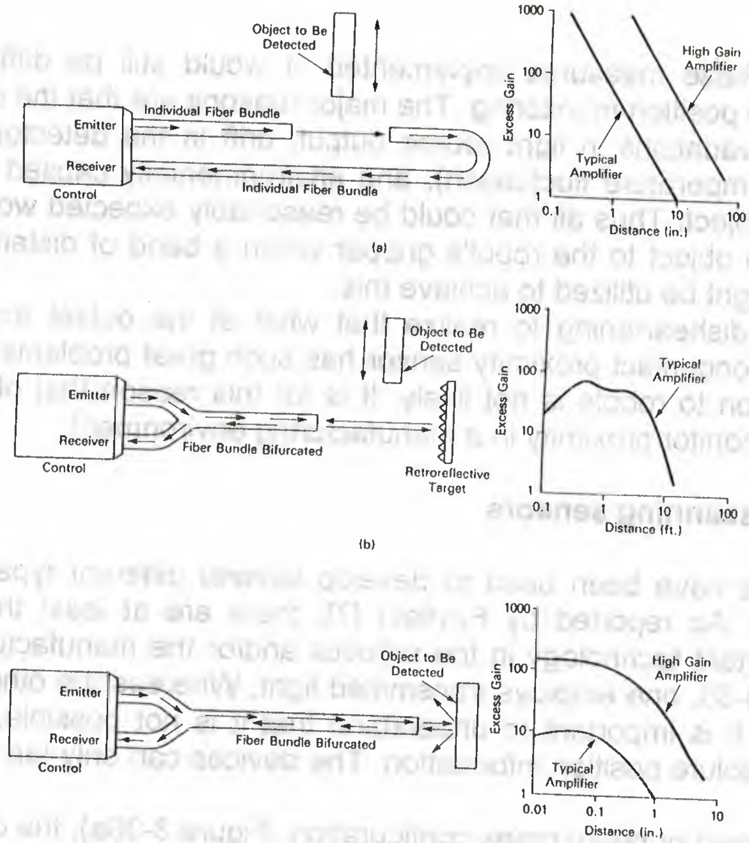


Figure 3-36 Three possible configurations of a fiberoptic scanning sensor: (a) opposed or beam break; (b) retroreflective; (c) diffuse. (Redrawn with permission of Production Engineering, Cleveland OH. From reference [7].)

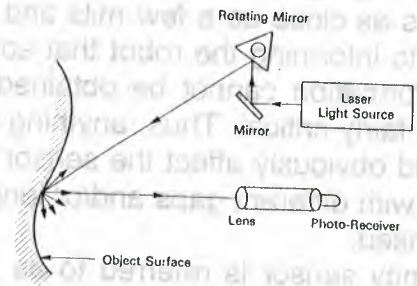


Figure 3-37 Scanning laser proximity sensor. Object scanning is accomplished by having a motor rotate a mirror at a constant angular velocity. (With permission of M. Ueda. From reference [8].)

The last fiber optic proximity scheme is shown in Figure 3-36c. Here a bifurcated fiber bundle is again used, but there is no retroreflective target. The sensor actually can measure the amount of light reflected from an object up to a few inches away from the fiber bundle. Since most materials reflect some light, this "diffuse" device can be used to detect transparent and translucent objects. As in the case of the reflected light proximity detector described above, some degree of absolute position monitoring is possible under ideal conditions. However, all of the difficulties with that type of sensor that were described previously are also present with the fiber optic version.

3.9.5 Scanning laser sensors

A considerably more involved and costly proximity sensor is shown in Figure 5.8.4. Consisting of a laser light source, two mirrors, one of which is rotated by an ac motor, and a lens-photo-receiver assembly, this scanning laser device has been used to permit an industrial robot to arc-weld curved objects [8]. The incident light beam from the laser (helium-neon) is "swept" across the object surface by the action of the motor-driven triangular mirror. Note that this occurs three times for each motor revolution. A lens mounted in front of a photodetector (e.g., a phototransistor) permits light reflected from only one point on the object's surface to be acquired. Distance from the sensor to this point is determined by synchronizing the ac motor voltage with a high-frequency clock. The number of clock pulses from the time this voltage is zero until the photodetector receives reflected light is a measure of the distance. T racking (in the case of the welding application, for example) is achieved by mounting the sensor on the end effector of the robot, thereby allowing the entire sensor to bc moved to different locations in space. Black, transparent, or extremely shiny objects cause problems for this proximity technique.

3.9.6 Ultrasonic sensors

Ultrasonics has been used to provide ranging and imaging information for many years. For example, naval vessels have used sonar sensing systems to detect submerged submarines since the early 1940s. Also, since the late 1970s ultrasonic imaging has been used to provide "pictures" of various human organs without subjecting patients to more objectionable forms of radiation (e.g., x-rays). The advent of Polaroid Corporation’s Sonar Sensing element brought ultrasonic ranging to the nonprofessional photographic market. As employed on their instant camera, the Polaroid sensor projects a low (energy)-level electrostatically generated ultrasonic pulse and measures the time for the reflected beam or "echo wave" to return to the sensor. This information is used to measure the distance to the object and then to automatically adjust the camera's focus accordingly. In recent years, this and other similar detectors have been adapted to robots.

Although there are several different sonar sensing techniques, robots normally utilize devices that produce short bursts of a sinusoidal waveform whose frequency is above the audio range (e.g., 40 kHz) [9]. From the block diagram in Figure 3-37, the operation of such a sensor can be understood. When an initiate signal is given, the transmit (sinusoidal) oscillator is enabled for 1 ms. This causes 40 cycles of energy at

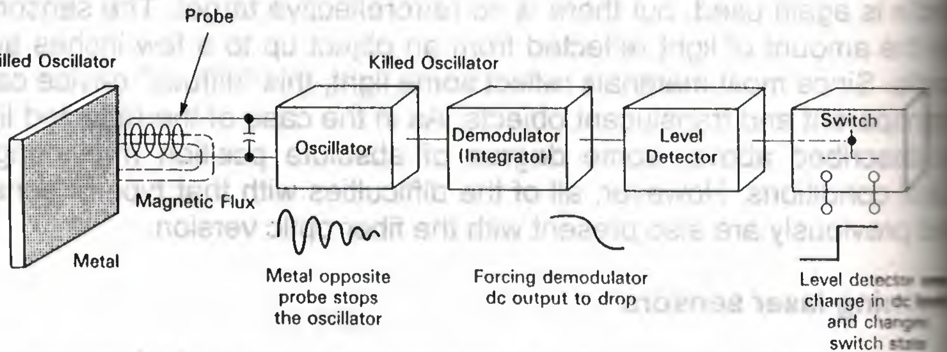


Figure 3-38 Killed oscillator, eddy current, noncontact, proximity sensor. By removing the level detector and switch, the device can monitor absolute distance. (Redrawn with permission of *Machine Design*. From reference [13].)

40 kHz to be transmitted. In addition, a timing or sampling window enables the AND gate for a specific period of time, thereby permitting any reflected pulses (from the echo wave) to be counted. This count is proportional to the distance of the object from the transducer. Thus the maximum range of the sensor can be varied by adjusting the sampling window. Errors due to spurious signals are prevented by using a narrowband filter tuned to the frequency of the transmitted beam (e.g., 40 kHz). The direct wave from the transmitter is not counted due to the action of the blanking pulse, which keeps the counter disabled until after the transmit oscillator is off (i.e., disabled).

In the case of the Polaroid device, the burst actually consists of four distinct high frequencies: 50, 53, 57, and 60 kHz. This prevents the surface topology or material of the object being scanned from "looking" like a matched termination to the incident ultrasonic energy which would eliminate or "cancel" the echo wave, thereby rendering the object effectively "invisible" to the sensor.

The idea behind using a group of frequencies is that the matching effect is frequency sensitive. Thus some energy will always be reflected. The "penalty" one pays for this is that the sensing electronics used in conjunction with such a device will be more complex since they must be able to handle four different frequencies rather than a single one as described above. The accuracy of the Polaroid sensor is reported to be about 1% (of its range).

Arrays of Polaroid sonar sensors have been used by the National Bureau of Standards to create a safety "curtain" of sonar energy about an industrial robot [10]. If this curtain is broken by someone attempting to enter the workspace, the robot is immediately halted and must be reset before it can resume operation. This scheme prevents an accident even when an intruder is entirely within the work envelope. Note that permitting the robot to continue moving once the sonar shield is reestablished would not safely handle such a situation.

3.9.7 Eddy-current sensors

Another type of sensor used as a proximity switch and for determining the accuracy and repeatability of commercial robotic manipulators operates on the eddy-current principle. A typical device of this class utilizes a sensing coil to induce high-frequency (eddy) currents in a ferrous or nonferrous (e.g., aluminum) conductive target. The amplitude of the sensor's generated oscillation depends on the distance between the metal surface and the coil, and this in turn determines the amount of magnetic coupling in the overall circuit. Consequently, position can be obtained by monitoring the amplitude.

One technique for doing this employs a "killed oscillator," [13] as shown in Figure 3-38. The presence of a metal target near the coil in the sensor's probe causes the oscillator's amplitude to drop since the induced eddy currents represent a loss mechanism and thus produce damping or "killing" of the sinusoidal waveform. A demodulator, essentially an integrator, responds to such a change by producing a smaller dc output. In a proximity switch application, a thresholding circuit is used to detect when the level drops below some predetermined value, at which point the switch's state is changed. By adjusting the threshold value, a robot manipulator can be stopped at a desired distance from a part or object. This scheme can also be used to

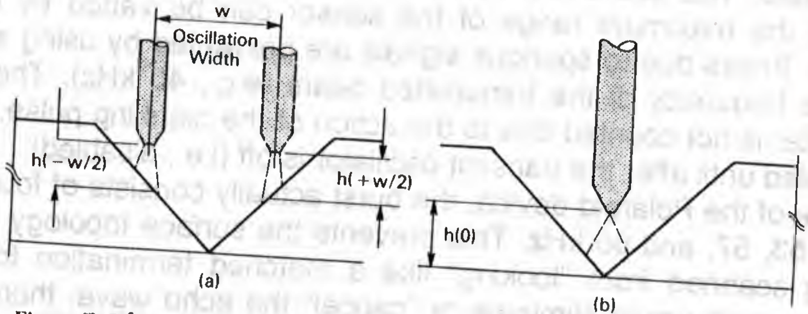


Figure 3.39 Through-the-arc "resistive" (position) sensing for automated seam tracking in an arc welding application. (Redrawn with permission of G.E. Cook. From reference [14].)

prevent an inadvertent collision between the end effecter and another piece of machinery within the robot's workcell.

3.9.8 Resistive sensing

A problem encountered in the robotic application of arc welding is keeping the welding tool (also referred to as a "torch" or "gun") tip at a specified constant distance from the seam that is to be welded. By doing this and also keeping constant the speed with which the gun is moved, the uniformity and strength of the weld can be controlled. In addition, a strong weld can be ensured by having the robot accurately "track" this seam. A technique that has been developed to meet these requirements is called through-the-arc resistive sensing. The fundamental principle underlying such a sensory technique is that for a constant voltage applied to the welding tool, the arc's resistance (or more correctly, the current) is a measure of the height of the torch tip above the surface that is to be welded. Inductive current monitoring is utilized because welding normally produces large currents (e.g., 100 to 200 A).

In the case of gas metal arc welding (GMAW), more commonly Called the metal inert gas (MIG) technique, a blanket of inert gas (e.g., argon, helium, or carbon dioxide) protects the welding torch's electrode, as well as the material being welded, from exposure to the air and, hence, rapid oxidation. The electrode consists of a wire continuously supplied from a drum. The composition of this wire varies depending on the nature of the welding application because the electrode's metal is used as a filler in the MIG process. It is found that for this type of welding, the relationship between the arc current I and voltage V is given by

$$V = R(h-L)I$$

where R = average resistivity per unit length of the electrode wire L = arc length h = height of the tool tip from the metal surface.

Normally, V is a constant, so that I is an inverse function of h. Thus, by adjusting the robot's position, the arc current can be kept fairly constant. It has been found, for example, that in the MIG process? It can be vary by about 1 to 1.5% for each 40mil change in h.

The problem of automated seam tracking has been solved by a modification of the foregoing method (see Figure 3-39). In this through-the-arc position sensing, the welding tool is deliberately moved back and forth a small distance across the seam between the two pieces of metal [14]. This action is often called weaving and is available as a selectable motion, often with a variable oscillation amplitude, on many robots (e.g., a Unimation PUMA 550). If the center of the seam is being properly tracked, the arc current at the maximum and minimum points on the weave (or oscillation) will be the same. However, if the gun has moved away from the seam's center, these two currents will differ, and this difference signal (or error) can be used to realign the manipulator. The vertical height h can also be controlled by sampling the arc current at the center of the torch oscillation and then comparing this current with some reference value, which has been determined in advance (i.e., off-line) and depends on the type of material being welded. Again, the difference between the reference and actual currents produces an error signal that can be used to readjust the robot's position above the welding surface.

A similar seam and height tracking procedure is possible for the other major type of arc welding [i.e., the tungsten inert gas (TIG) process, also called gas tungsten are welding (GTAW)]. In addition, other weld tracking systems have been developed, but these are usually of the contact variety. We will discuss them in a subsequent section when tactile devices are described.

3.10 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).

3.10.1 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 are welding seam trackers.

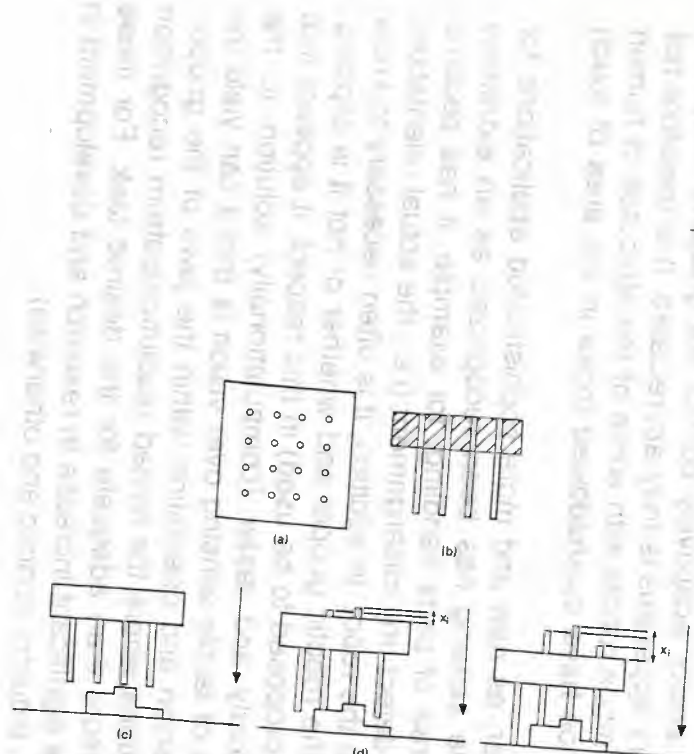


Figure 3-40 Three-dimensional proximity rod tactile sensor: (a) top view; (b) side view; (c) sensor descending; (d) sensor in partial contact with object; (e) sensor in full contact with object. (Courtesy of W.B. Heginbotham, Nottingham, England. From Page, C.J. and Pugh, A., "Novel Techniques for Tactile Sensing in a 3-D Environment," *Proc. of 6th Int'l Symp. on Industrial Robots*, University of Nottingham, England, March 24–26, 1976.)

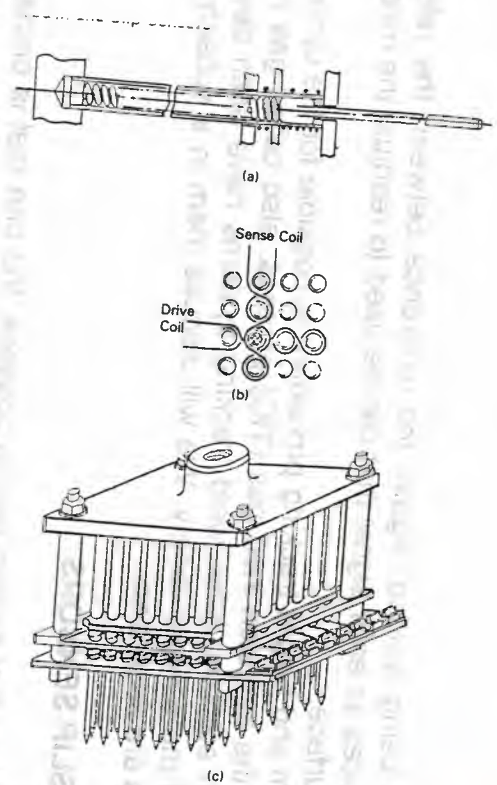


Figure 3-41 The sensing principle of a 64 (8 × 8) element proximity rod tactile sensor: (a) single spring loaded ferrite cylinder element; (b) top view of the sensing and drive coil configuration; (c) drawing of the actual sensor. (Courtesy of W.B. Heginbotham, Nottingham, England.)

3.10.2 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 [21] is shown in Figure 3-40, 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 3-40c). 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 3-40e). 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 3-40d). 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 3-40e). 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 empolyed 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.

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 3-41). 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 [22]. Obviously, many of the other position-sensing

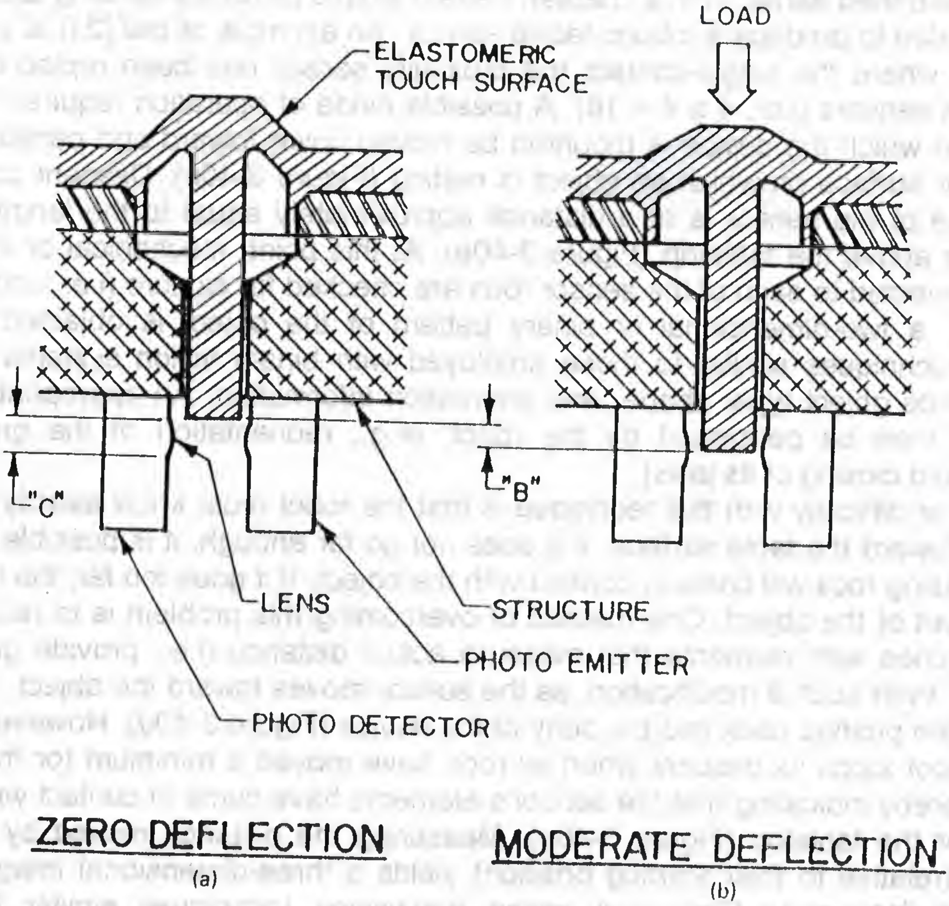


Figure 3-42 One element of a photodetector tactile sensor: (a) no load applied to the sensor's surface; (b) a load results in partial obstruction of the light beam from the photo emitter causing a change in the photodetector output. (Courtesy of J. Rebman and the Lord Corp., Industrial Automation Division, Cary, NC.)

methods discussed in earlier sections of the chapter could also be used. However the more expensive ones (e.g., the LVDT) would not be practical since each rod would require a separate position sensor.

3.10.3 Photodetector Tactile Sensors

Among the noncontact proximity sensors discussed, it will be recalled that one used the "beam break principle." Over the last decade, several tactile sensors have been developed utilizing this technique [23]. In 1983, the Lord Corporation described a commercial device [24] shown in Figure 3-42. [Actually, only one sensor element is indicated. In reality the sensor would, of course, consist of an array of such elements, (e.g., 8 x 8 for the Lord LTS 100). It can be seen that the portion of the sensor that comes in contact with the abject to be sensed is covered with an elastomer (a rubber-like material). In addition, a piece of this material extends through the sensor structure. Mounted on the back of the body of the device is a photo emitter-detector assembly (sac Figure 3-42a). When the object comes in contact with the touch surface, if the elastomer is compressed a minimum distance, the material extending through the body breaks the beam of the photosensor (see Figure 3-42b). Obviously, a thresholding circuit can be used to provide binary information about the object, that is, whether or not each element of a sensor composed of such devices is in contact with a part. In a manner similar to that described for the proximity rod contact sensor, two-dimensional information about size, shape, and part orientation can be obtained.

It is also possible to determine information about the relative deflection at each array point, that the voltage output from a photodetector varies with the incident light intensity. Thus, by monitoring the actual signal from the individual photodetectors, the voltage level can be related to distance traveled by the sensing element. Depth (e.g., three-dimensional) information is limited since the overall travel distance is quite small. For example, the elastomer used in the Lord LTS 100 will deflect a maximum of 2 mm. However, the voltage variation at each array point can be related to the pressure or force being applied by the robotic gripper. Clearly, this is a desirable attribute of such a sensor. The above-mentioned device will sense a force of 1 lb applied to any single sensing site at a full mechanical deflection of 2 mm.

At least two potential difficulties occur with such a sensor. The first has to do with mechanical hysteresis in the elastomer. This implies that the rubber will not return to its original position after it has been compressed. For a binary device, proper thresholding of the photodetector signal level will probably minimize the problem. However, this effect will create severe problems with a sensor that is supposed to provide absolute voltage-level information (as is the case with a device that also gives pressure or force data). The severity of the problem is reduced somewhat. however, by the small travel distance (e.g. 2 mm).

The second problem with using this type of tactile sensor has to do with ruggedness in a manufacturing environment. Since the elastomeric surface must actually come in contact with the object being grasped by the robot, it is quite likely that significant wear will take place. This means that unless the rubber is carefully protected, it will have to be replaced quite frequently. Depending on the actual application, this may or may not be an acceptable solution.

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 quality.

CHAPTER 4. COMPUTER CONSIDERATIONS FOR ROBOTICS SYSTEM

4.1 INTRODUCTION

The purpose of this chapter is to provide an understanding of computer architecture of robotic system. It helps to gain appreciation of the practical consideration that comprises the selection of a computer system from both the hardware and software point of view.

4.2 ROBOT PROGRAMMING

As discussed in previously Chapter, the most sophisticated robot control systems have a programming capability that allows for elemental decision making, a capability needed to coordinate a robot's actions with ancillary devices and processes (i.e., to interface with its environment). Branching is the ability of the software to transfer control during program execution to an instruction other than the next sequential command. At a specific point in a task cycle, the robot will be programmed to anticipate a branching signal—a special electrical signal sent to the controller by a designated internal or external sensor. If such a signal is received, the program will follow a predetermined path or function (branching). If no signal is received, the program will continue to follow the main path. Thus a robot interacting with a group of machine tools will perform a given sequence of operations, depending on which steps have been completed. For example, after a raw part is loaded onto a press, the program will look for a branching signal. If the signal is received, the program will branch to a pause, causing the robot to wait while an ancillary machine works on that part. After the machine has completed the prescribed work, an external completion signal is sent to the controller by a sensor located on that ancillary machine. Then the robot is directed to take the part out of the press and transfer it to another machine. Decision making can also be used to correct an operational problem. For example, a program may have a branch to a taught subprogram for releasing a jammed tool.

Robot languages provide flexibility to the user in defining the task to be performed. Not only do they permit the motion of the task to be defined but they also provide the user with the ability to imbue intelligence in the control program. In its simplest forms, this intelligence may check binary sensors and change a location, or make a simple decision based on sensory information to handle an exception. As the capability of the language increases, the intelligence of the algorithm controlling the robot in a specific application can also increase. Thus corrections based on sensory inputs (such as vision or tactile sensors} are possible along with communication with other computers and data bases.

Historically, the initial applications of robots were relatively simple and accordingly, their controllers did not require or provide sophisticated sequence control. Typically, the following sequence was all that was needed:

- Move to a specified location in space
- Control the state of a gripper
- Control the state of output lines
- Provide sequence control based on the state of input lines

As applications became more complex, and computer technology more advanced, techniques were developed to take advantage of the newer computer architectures.

In the following section, techniques for robot control sequencing will be presented from three appropriately more progressive perspectives (fixed instruction sequence control, robotic extensions to general purpose programming languages, and robot-specific programming languages). This is followed by a summary of robot programming languages and two examples illustrating these methods are presented. The section concludes with a discussion of how points in space are taught or "demonstrated" to a robot.

4.2.1 Robot Control Sequencing

Robot sequencing can be accomplished in a variety of ways. As discussed in Chapter 1, there are certain features of functionality required by a robot control system in order to facilitate both the training (programming of the sequence of events) and its use with ancillary equipment. To someone familiar with general-purpose programming languages? It is obvious how certain aspects of this functionality can be easily provided by a computer language. What may not be as obvious is that most of the important functions needed for manipulator control and simple interfacing can be implemented by dedicated sequencers? These sequence controllers accept commands (possibly given by the setting of switches) and record the robot's joint positions. The sequencing of the manipulator is achieved by "playing back" the desired states at a later time. In a certain sense, these sequencers also possess the power of programming languages but without all the explicit commands and data structures associated with a formal programming language.

To contrast various "programming" methods, all of which permit the user to define the sequence of operations of a manipulator, three distinct implementations will be discussed. Specifically, they are:

- Fixed Instruction Sequence Control
- Robotic Extensions of General-Purpose Programming Languages
- Robot-Specific Programming Languages

The first is a relatively simple method which makes use of a fixed event sequence in each instruction. The second is based on extensions of programming languages which add robot-specific functions (or subroutines) to the standard library, or in which robot-specific commands have been added to the structure of the language. The third is a language tailored specifically to the programming or training of robots.

4.2.2 Fixed instruction sequence control

In this mode of implementation, the sequence of the robot's operation is defined by means of a "teach pendant" which provides the ability to position the tool point of the manipulator in space by means of buttons or a joystick. Additional controls allow the trainer to define the state of the gripper (open or closed) and the state of each of the output lines (on or off) as well as time delays and simple branching based on the state of input lines. By saving joint position, and other state data, a sequence of events can then be defined.

To better understand the nature of a fixed instruction sequence controller, the implementation used on the Mark I controller from United States Robots will be examined. In general, each program step consists of a series of actions. These are:

- Check the status of input lines
- Check for a subroutine call
- Perform a robot motion
- Delay a specified time interval
- Set the state of the gripper (open or closed)
- Set the state of output lines

To understand how this relatively simple structure can provide sufficient program control, and for the sake of discussion, let us assume that the controller already has a number of programs stored in its memory. A specific program is first selected (by number) utilizing a series of thumbwheel switches. To begin the sequence of actions defined by the program, a "start" switch is depressed which causes the first instruction to be obtained from memory. First, a logical "AND" of a subset of the input lines is performed against a "mask" stored in memory. It should be understood that the program will wait indefinitely until the specified input line(s) are asserted. Next, if the step is a subroutine (another series of program steps), then it is executed and the following program step is obtained from memory (note that the motion and subsequent steps are not performed in this case). If no subroutine call was indicated? Then the robot controller causes the manipulator to move to a point in space defined by a set of joint variables stored in memory. Once this location is reached, the remaining actions (for the current program step) are executed. These include waiting a specified delay time, opening or closing the gripper, and the final action, which is the setting of the state of the output lines to a value defined in the programming sequence. Following this, the next program instruction (step) is fetched from memory and decoded as defined previously. After all the steps of a particular program are executed, the sequence repeats from the first step. That is, the controller keeps executing the program indefinitely.

Due to the nature of the fixed sequence of actions for each program step, it may be necessary to program additional steps to properly sequence the manipulator. For example, it is necessary to provide a delay to ensure gripper activation prior to arm motion. This is due to the fact that it takes a finite time for a gripper to reach its final state after its activating mechanism receives its control signal. Therefore, the trainer might want to insert a delay (on the order of a few hundred milliseconds) prior to the execution of any other manipulator motion. Since the action sequences of a program step without a subroutine call are check inputs, perform motion, delay, set gripper state, and set output line states, one easily sees that it is possible for the next program step to cause a motion (if the input conditions are satisfied immediately) before the gripper's state has stabilized. To accomplish a delay prior to the motion of this subsequent step, it is necessary to program an additional step in which no motion occurs but which makes use of the delay in the sequence of actions.

While this type of programming may require substantial human activity, it is still able to produce the desired results (i.e., sequencing a manipulator through a set of motions). The key to both successful and efficient programming of this type of controller is knowing the sequence of actions and how to take advantage of them.

As the complexity of the tasks being performed by robots increased, the demands for more advanced motion control and decision capability also increased, thereby requiring more sophisticated programming methods. In some cases, the simple sequencing controls could be expanded by adding more functionality to the teach pendant by means of multiple levels and added control switches. Besides increasing the complexity of the teach pendant, this approach also increased the programming time and required skill level of the trainer.

An outgrowth of such complex sequence controllers is a "menu-driven" programming system that permits the training of the robot using a fixed set of functions. The menu system differs from the "fixed instruction" sequence control in that instructions specific to each function are generated.

One major advantage of a menu system, however, is that it may be easily extended to accommodate new functions and even provide interfaces to external sensors such as vision. It should be apparent that this concept can also be extended to a robot-specific language by adding a terminal interface and the typical language functionality such as syntax checking of instructions prior to execution (or during compilation).

Although extensions of fixed instruction sequence control could certainly have provided additional capability, they lacked flexible program control and data structures.

4.2.3 Robotic extensions of general-purpose programming languages

Another step in the evolution of robot programming was the incorporation or a language. The use of a general purpose programming language with extensions provides the user with the control and data structures of the language. The robots

Table 4-1

	AL	AML	HELP	JARS	MCL	RAIL	RPL	VAL
Language Modalities								
Textual	×	×	×	×	×	×	×	×
Menu		×						
Language Type								
Subroutines				×			×	
Extension					×			
New language	×	×	×			×		×
Geometric Data Types								
Frame (pose)	×			×	×	×		×
Joint angles		×		×				×
Vector	×	×		×	×			
Transformation	×			×	×			×
Rotation	×			×	×			
Path						×		
Control Modes								
Position	×	×	×		×	×	×	×
Guarded moves	a	a	a					
Bias force	×							
Stiffness/compliance	×			b				
Visual servoing	c			c		c	c	c
Conveyor tracking				×	×			
Motion Types								
Coordinated joint between two points	×	×	×	×		×	d	×
Straight line between two points	c			×	×	×	d	×
Splined through several points	×		×	×		×	d	×
Continuous path ("tape recorder" mode)								
Implicit geometry circles					×			
Implicit geometry patterns					×			
Signal Lines								
Binary input	0	64	f	0	242	6	32	32
Binary output	0	64	f	2	242	10	32	32
Analog input	64	0	f	0	242	0	32	0
Analog output	4	0	0	0	242	0	64	0
Display and Specification of Rotations								
Rotation matrix	g			h				
Angle about a vector	×			h				
Quaternions								
Euler angles	i	×		×		×		×
Roll-pitch-yaw	j		×		×			
Ability to Control Multiple Arms								
Multiple arms	×		×		×			
Control Structures								
Statement labels		×	×	×	×		×	×
If-then	×	×	×	×	×	×	×	×
If-then-else	×	×	×	×	×	×	×	
While-do	×	×	×	×	×	×	×	
Do-until	×	×		×		×	×	

	AL	AML	HELP	JARS	MCL	RAIL	RPL	VAL
Case	×			×		×	×	
For	×	×		×		×	×	
Begin-end	×	×		[k]	[l]			
Cobegin-coend	×		[m]		[l]			
Procedure/function/ subroutine	×	×	×	×	×	×	×	×
Successful Sensor Interfaces								
Vision	×	[n]	×	×	×	×	×	×
Force	×	×	×					
Proximity								
Limit switch	×	×		×	×	×	×	×
Support Modules								
Text editor	[p]	×	[o]	[o]		×	×	×
File system	[p]	×	[o]	[o]		×	×	×
Hot editor		×						
Interpreter	×	×	×					
Compiler	×			×	×		×	
Simulator	×	[q]			×			
MACROs	×		×		×			
INCLUDE statement	×	×						
Command files				×			×	
Logging of sessions	×							
Error logging	×							
Help functions	×	×						
Tutorial dialogue		×						
Debugging Features								
Single stepping		×	×			×	×	
Breakpoints	×	×				×	×	
Trace		×	×		×		×	
Dump	×		×		×		×	

Source: Reprinted courtesy of the Society of Manufacturing Engineers. Copyright 1983 from the ROBOTS 7/13th ISIR Conference Proceedings.

[a]Using force-control or limit-switch action.

[b]Currently being implemented at Jet Propulsion Laboratory.

[c]Uses visual inputs to determine set points but does not specifically perform visual servoing.

[d]Relies on the VAL controller.

[e]Currently being implemented at Stanford University.

[f]Custom for each system.

[g]AL displays rotations as a rotation matrix.

[h]Normally, JARS does not display these forms; however, the user may write a routine to print them because JARS has the forms available internally.

[i]AL accepts directly the specification of an orientation by three Euler angles (or by an angle about a vector).

[j]AL orientations could also be specified by roll–pitch–yaw angles.

[k]Since it is a language based on subroutines added to Pascal, JARS has all the structures of Pascal.

[l]MCL can invoke tasks in parallel using INPAR.

[m]HELP permits the simultaneous activation of several tasks.

[n]Reported by the IBM T. J. Watson Research Center, Yorktown Heights, New York; not commercially available.

[o]JARS and HELP use the systems support features of the RT-11 operating system.

[p]AL uses the support features of the PDP-10 operating system.

[q]A simulator has been developed at the IBM T. J. Watson Research Center, Yorktown Heights, New York.

specific operations are handled by Subroutines or functions. Clearly this implies that the training of a robot now requires a person well versed in the concepts of computer programming.

Various permutations of this concept are possible, including the use of subroutines as compared to extensions of languages. The extensions to the language include robot-specific commands (and possibly new data types) in addition to the existing set of commands (and data types) while leaving the general syntax and program flow intact.

An advantage of using an extension of a general-purpose programming language is that the designers can concentrate on the problem at hand, designing a robot instead of spending time designing a sequencer, providing editing capabilities, and so on. The actual implementation may make use of a compiled or interpreted language depending on the nature of the base language chosen to be extended and the objectives of the design team. One other advantage in extending a language is that more sophisticated cell control can be handled by the robot controller. In this case, it now has more power to perform nonrobot input/output and has the ability to perform certain man-machine interfaces, e.IA.7 statistical and error reporting.

An example program for the United States Robots' MAKFR 22 Sahara robot is illustrated in Table 4-1. (This example is treated in detail in Section 4-1.) It is interesting to note that this is the form used to program most Scara robots from Japan.

This programming method (as compared to the fixed instruction technique) makes use of program control, specifically the FOR-NEXT loop and the STOP statements. One should also observe that there are statements that do not cause robot motion and the sequence of events is chosen by the programmer or trainer. Thus it is seen that some of the constraints imposed by the fixed event instruction are removed.

As the available technology became more sophisticated and manufacturing requirements grew, the limited flexibility of the language extension approach became obvious. This provided the impetus for the development of robot-specific languages.

4.2.4 Robot-specific programming languages

A major motivating factor that led to the development of robot-specific programming languages was the need to interface the robot's control system to external sensors in order to provide "real-time" changes to its programmed sequence based on sensory information. Other requirements such as computing the locations for a palletizing operation based on the geometry of the pallet, or being able to train a task on one robot system and perform it on another (with minor manual adjustment of the points) also were an impetus. Additionally, requirements for offline programming, CAD/CAM interfacing, and more meaningful task descriptions led to various language developments.

Table 4-2 shows a complete terminal session of a Westinghouse/Unimation robot using VAL 1. This example, discussed more fully in Section 4-2, shows an entire environment for the training of the robot. As shown in the table, the program is retrieved from a mass storage device, then listed, and the fixed positions defined in the program are displayed. Finally, the program is executed and output, indicating the current cycle, is displayed on the terminal. As the listing indicates, this language clearly provides more capability for complex robot control than that of the fixed instruction sequencer or the extended language examples described previously.

Section 4-2, presents various commercial and research robot programming languages and a table that compares program control, robot specific mathematics, and input/output capability for each language. Once again, it should be noted that regardless of the complexity of the programming language, the objective is to define a sequence of operations that are needed to obtain successful control of the robot .

4.2.5 Languages Selected Summary of Robot

Currently, a large number of robot languages are available, although no standards for these exist. The more common languages include:

- AL
- AML
- RAIL
- RPL
- VAL

Brief descriptions of each of these are given below This summary is adapted from a paper by Gruver et al. [9].

AL

AL was the second-generation robot programming language produced at the Stanford University Artificial Intelligence Laboratory, an early leader in robot research. Based on concurrent Pascal, it provided constructs for control of multiple arms in cooperative motion. Commercial arms were integrated into the AL system. This language has been copied by several research groups around the world. Implementation required a large mainframe computer, but a stand-alone portable version was marketed for industrial applications. It runs on a PDP 11/45 and is written almost entirely in OMSI Pascal [9]. In the AL system, programs are developed and compiled on a PDP-10. The resulting p-code is downloaded into a PDP-11/45, where it is executed at run time. High-level code is written in SAIL (Stanford Artificial Intelligence Language). The run-time system is written in PALX. The PDP 11/45 has a floating-point processor, no cache memory, a single terminal, and 128 kilobytes of RAM memory. Two PUMA 600 s and two Stanford Scheinman arms were controlled at the same time by this language.

AML

A manufacturing language (AML) was designed by IBM to be a well structured,

semantically powerful interactive language that would be well adapted to robot programming. The central idea was to provide a powerful base language with simple subsets for use by programmers with a wide range of experience. An interpreter implements the base language and defines the primitive operations, such as the rules for manipulating vectors and other "aggregate"
objects that are naturally required to describe robot behavior. A major design point of the language was that these rules should be as consistent as possible, with no special-case exceptions. Such a structure provides a ready growth path as programmers and applications grow more sophisticated. AML is being used to control the RS/1 assembly robot, a Cartesian arm having linear hydraulic motors and active force feedback from the end effector. The computer controller on the RS/I assembly robot consists of an IBM series/1 minicomputer with a minimum of 192-kilobyte memory. Peripherals include disk and diskette drive, matrix printer, and keyboard/display terminals. A subset of AML was employed on the Model 7535 robot that was controlled by the IBM personal computer. However, the features of this version are not included here since the 7535 is no longer being marketed by IBM.

RAIL

RAIL was developed by Automatix, Inc. of Bilerica, Massachusetts as a high level language for the control of both vision and manipulation. It is an interpreter, loosely based on Pascal. Many constructs have been incorporated into RAIL to support inspection and arc-welding systems, which are a major product of Automatix. The central processor of the RAIL system is a Motorola 68000. Peripherals include a terminal and a teach box. RAIL is being supplied with three different systems: vision only, no arm; a custom-designed Cartesian arm for assembly tasks; and a Hitachi process robot for arc welding.

RPL

RPL was developed at SRI International to facilitate development, testing, and debugging of control algorithms for modest automatic manufacturing systems that consist of a few manipulators, sensors, and pieces of auxiliary equipment. It was designed for use by people who are not skilled programmers, such as factory production engineers or line foremen. RPL may be viewed as LISP cast in a FORTRAN-like syntax.

The SRI Robot Programming System (RPS) consists of a compiler that translates RPL programs into interpretable code and an interpreter for that code. RPS is written mostly in Carnegie-Mellon's BLISS-11 and cross-compiles from a DEC PDP-10 to a PDP-11 or LSI-II. The programs written in this language run under RT-11 with floppy or hard disks. The RPL language is implemented as subroutine calls. The user sets up the subroutine library and documents it for people who must write RPL programs. Previously, SRI operated the Unimate 2000A and 2000B hydraulic arms and the SRI vision module with this language.

VAL

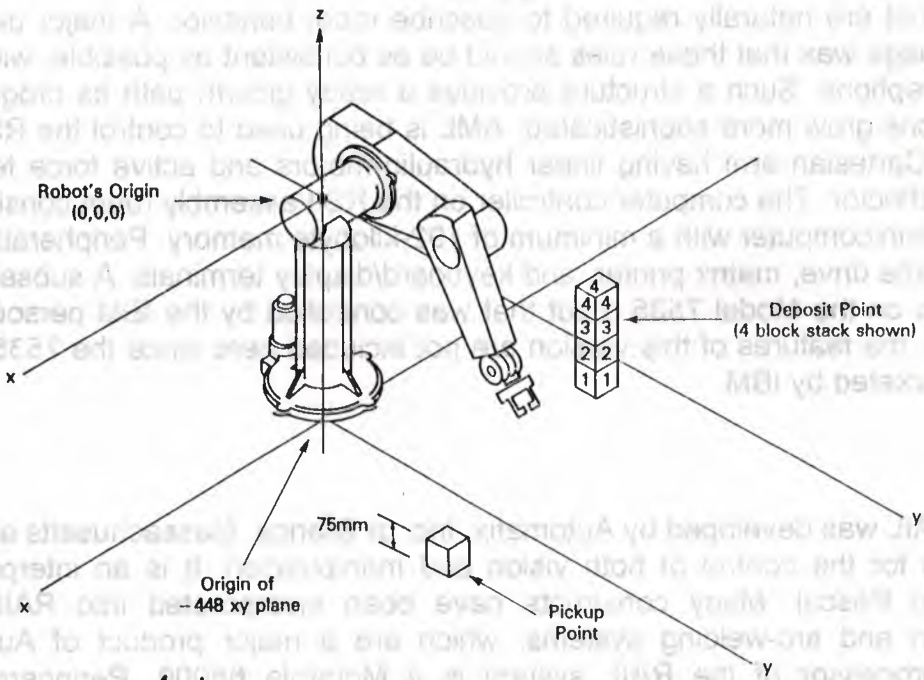


Figure 4-1 Workspace for VAL programming example. The pickup and deposit points are on the *xy*-plane offset (in *z*) from the robot's origin by −448 mm.

VAL is a robot programming language and control system originally designed for use with Unimation robots. Its stated purpose is to provide the ability to define robot tasks easily. The intended user of VAL will typically be the manufacturing engineer responsible for implementing the robot in a desired application.

Eight robot programming languages are compared in Table 4-1. Prior programming knowledge is helpful but not essential. VAL has the structure of BASIC, with many new command words added for robot programming. It also has its own operating system, called the VAL Monitor, which contains the user interface, editor, and file manager. The central monitor contains a DEC LSI-I I/ 03, or more recently, the LSI-11/23. In a Puma 55() robot, each of the joints is controlled by a separate 6503 microprocessor. The monitor communicates with the user terminal, the floppy disk, the teach box, a discrete I/O module, and an optional vision system. VAL is implemented using the C language and the 6502 assembly language. It has been released for use with all PUMA robots and with the Unimate 2000 and 4000 series. The languages described above as well as three others, HELP, JARS, and MCL, are compared in Table 7.6.1 and have been adapted from Gruver et al. [9].

Sample Programs

The following examples illustrate the use of two different robot programming languages, VAL and one employed on a particular Scara-type manipulator.

EXAMPLE 4.1 VAL Example

Assume that it is desired to pick up identical objects from a known location and then stack the objects on top of each other to a maximum stacking height of four. Figure 4-1, shows the application.

Let us consider this application and its implementation in the VAL programming language. Table 4-2, is a listing of a session on the terminal, which includes loading and listing the program, viewing the value of the stored locations, and finally, executing the program.

The dot (.) in the leftmost column is the prompt, which tells the user that VAL is ready to accept a command.

TABLE 4.2. LIST OF A VAL TERMINAL SESSION

```
.LOAD STACK
        .PROGRAM STACK
        .LOCATIONS
OK
.LISTP STACK

.PROGRAM STACK
```

```
1. REMARK
2. REMARK  THIS PROGRAM PICKS UP PARTS FROM A FIXED
3. REMARK  LOCATION CALLED PICKUP, THEN DEPOSITS THEM AT A
4. REMARK  LOCATION CALLED B. IT IS ASSUMED THAT 4 PARTS
5. REMARK  ARE TO BE STACKED ON TOP OF ONE ANOTHER.
6. REMARK
7.      OPENI

8. SET B = DE POSIT
9. SETI COUNT = 0.
10. APPROS PICKUP, 200.00
11. MOVES PICKUP
12. CLOSEI
13. DEPARTS 200.00
14. APPRO B. 200.00
15. MOVES B
16. OPENI
17. DEPARTS 200.00
18.  SETI COUNT = COUNT + 1
20.  TYPEI COUNT
20.  REMARK COUNT INDICATES THE TOTAL NUMBER OF ITEMS STACKED
21.  IF COUNT EQ 4 THEN 20
22.  REMARK MOVE THE LOCATION OF B UP BY 75.00 MM.
23.  SHIFT B BY 0.00, 0.00, 75.00
24.  GOTO 10
25.  SPEED 50.00 ALWAYS
26.  READY
27.  TYPE *** END OF STACK PROGRAM ***
.END
.LISTL

              X/JT1 Y/JT2 Z/JT3 O/JT4 A /JT5T

DEPOSIT  - 445.03     130.59      - 448.44     -87.654      88.890       -
180.000
PICKUP            163.94      438.84      - 448.38      178.006      88.896
        -180.000

EXEC STACK

        COUNT =   1.
        COUNT =   2.
        COUNT =   3.
        COUNT =   4.

*** END OF STACK PROGRAM ***
```

PROGRAM COMPLETED: STOPPED AT STEP 28

robot controller, LOAD STACK, tells the system to recall the program and any location data from the disk. The system response is on the next three lines, indicating successful completion of this request. The following command to the controller is LISTP STACK, which tells VAL to list the program which is called STACK. This particular version also delimits the program listing by printing .PROGRAM STACK at the beginning and .END at the end. Two more commands that are used in the table are (1) LISTL, which commands the controller to print all the locations that the controller knows about (in this case there are two such locations, DEPOSIT and PICKUP), and (2) EXEC STACK, which tells the controller to execute the program called STACK, which is stored in its memory. Following the EXEC command is the output generated by the program STACK. This output is the value of the variable COUNT as the program is executed. Note that the value of COUNT is used to terminate execution of the program when the desired number of items have been stacked.

Examination of the program listing shows that each line has a number associated with it (i.e., I through 27). These numbers are used to identify a line so that the program may be edited. VAL has an editor that allows the user to create programs and store them in the controller. Once stored, a program may be modified by referring to its line numbers. The modifications include inserting, deleting, or modifying lines.

The operation of the robot based on the program steps will now be described .

• Lines I through 6 are comments.

• Line 7 tells the gripper to open immediately and then wait a small amount of time to ensure that the action took place.

• Line 8 equates the location of the variable B to a defined location called DEPOSIT. This step is necessary since the value of B will be modified each time a new item is stacked.

• Line 9 sets an integer variable called COUNT to zero. The variable COUNT is used to terminate the program when the proper number of items have been stacked (i.e., 4 items).

• Line 10 has a label (10) associated with it. It commands the robot to move from wherever it is along a straight line to a location 200 mm above the point called PICKUP. At the end of the motion, the approach vector of the gripper will be pointing downward. Recall that the approach vector is defined so that moving along it causes objects to go toward the inside of the gripper.

• Line 11 tells the robot to move its gripper in a straight line toward the position defined by PICKUP. In this example, the motion will be along the approach vector since the gripper is pointing downward. The position defined by PICKUP is such that when motion ends, the object will be inside the gripper's jaws.

• Line 12 commands the system to close the gripper and wait a sufficient amount of time for the action to occur. In some cases it may be necessary to add an additional delay if that provided by the command is insufficient.

• Line 13 tells the manipulator to move along its approach vector in the direction opposite from which it originally came to a point 200mm above the pickup point.

• Line 14 tells the manipulator to move to within 200mm of point B. aligning its approach vector downward.

• Line 15 commands the manipulator to move in a straight line until its tool point is coincident with location B.

• Line 16 tells the gripper to open so that the part can be deposited. This also includes some time delay for the action to occur. As stated previously, additional delay may be necessary to compensate for the actual valves and mechanics used to implement the gripper and to permit the manipulator to settle to the desired location.

• Line 17 tells the manipulator to move back along the approach vector so that it is 200 mm above location B.

• Lines 18 and l9 increment the variable COUNT and display its value.

• Line 20 is a comment.

• Line 21 is a test to see if COUNT is equal to 4. If so, go to the statement with label 20; otherwise, go to the next line.

• Line 22 is a comment.

• Line 23 modifies the location defined by B so that its z coordinate is increased by 75.0 mm.

• Line 24 forces the program to go to label 10.

• Line 25, which is labeled, tells the controller to reduce the speed of motions to 50%.

• Line 26 tells the controller to move the manipulator to its ready position, which is defined as all of the links in a straight line pointing upward.

• Line 27 tells the controller to print a message to the terminal.

From the description of the program, one can easily see the power implemented by the instructions. Commands exist to cause the manipulator to move in a straight line and to manipulate position data. (Note that the "S" in the statement indicates that straight-line motion is desired.) For example, the variable B. Which represents a location (i.e., a set of six joint variables) is modified by a single statement in line 23. Similarly, the commands APPROS and DEPARTS are quite interesting because they actually define positions relative to a variable but do not make it necessary for the user to define the actual positions for each move that the robot has to make This concept is quite important for robot training, since we have really defined only two positions, PICKUP and B. However, we can move to many positions relative to them. Using this approach, if it is necessary to modify either of the points (PICKUP or B), the changes made to them will automatically be reflected in the intermediate points (selectively by the robot s path planner), which are defined solely on these two positions.

EXAMPLE 4-2, Scara Programming Example

The MAKER 22 is programmed in a language similar to BASIC, with robot specific extensions. For example, positions in space may be referenced by a single-variable name of the form Pxxx, where xxx is a three-digit number from 000 to 999. In order that position variables may be referenced by an index, it is possible to catenate the P with an integer variable such as A and refer to the point PA. Whatever the value (from 000 to 999) specified by the programmer. A will then reference the actual position variable. Certain operations may be performed on these position points, such as addition and subtraction. Additionally, provisions exist to multiply or divide a position by a scalar. Only two types of moves are provided in the language: M0V which causes the manipulator to move in a joint-interpolated fashion; and CP. which causes the robot to move in a continuous-path fashion Whenever a CP command is encountered. The controller will move the manipulator from its current location to the point which is the argument of the command while also looking ahead for the next CP command and its argument. The occurrence of the next such command tells the controller to continue moving toward this next such command tells the controller to continue moving toward the next specified position once it has come close to the location defined by the previous CP command. This process continues until the end of the program or a MOV command is encountered. It is clear that if one wanted the manipulator to follow a specific path, all that would be necessary is to define a sufficient number of points for the path and then write a program that uses CP moves to connect them.

The example that we explore illustrates the use of topics discussed in the previous paragraphs. It is desired to cause the MAKER 22 to move in a straight line. For our discussion, we will assume that two positions have been defined previously, P1 and P2*, and that we wish to have the manipulator move in a straight line starting from PI and ending at P2.

TABLE 4-3. MAKER 22 PROGRAMMING EXAMPLE

10: "STRAIGHT LINE"	Label with a comment
N = 10	Number of intermediate points plus 1
P100= P1	Copy P1 to P100
P101 = P2 - P1	P101 is distance to be moved
P101 = P101 / N	Incremental distance
MOV P100	Set manipulator at first point
For L = 1 TO N	Beginning of loop
P100 = P100 + P101	Compute intermediate point
CP P100	Do CP move to point
NEXT L	End of loop
STOP	

Table 4-3, shows a listing of the program and comments defining the purpose of the instructions. The program in Table 4-3, takes the difference between the initial and terminal points of the line and divides by the number of intermediate points plus 1 to compute an incremental distance. It then instructs the manipulator to move to the first point, P100. After attaining this position, it computes intermediate points by adding P101 to P100 and then instructs the robot to move in a continuous-point fashion connecting the 10 points to form an approximation to a straight line. Note that the last point is P2.

It should be apparent that the robot programming language for the MAKER 22 does not contain as high a level of expression as indicated in the example using VAL. This is obvious if one recognizes that a straight line is achieved with one instruction using VAL whereas it requires the entire program in Table 4-3, to perform the identical maneuver with the Maker 22. However, the same functionality, that is, the ability to move in a straight line, is provided by both languages.

4.3 Demonstration of Points in Space

To program a servo-controlled robot, a skilled operator often breaks down the assigned task into a series of steps so that the manipulator/tool can be directed through these steps to complete the task (a program). This program is played back (and may be repeated several times, i.e., it can be used as a subroutine) until the task cycle is completed. The robot is then ready to repeat the cycle. Thc robot's actions may be coordinated with ancillary devices through special sensors and/or limit switches. These, in conjunction with the controller, send "start work" signals to, and receive "completion" signals from other robots or interfacing devices with which that robot is interacting.

A servo-controlled robot can be "taught" to follow a program which, once stored in memory, can be replayed, causing the controller to be instructed to send power to each joint's motor, which in turn, initiates motion. This teaching process may require that the operator "demonstrate' points in space by causing the end effecter to move (using one of a number of possible methods) to a series of locations within the work cell.

The robot can also be taught its assembly tasks from a CAD/CAM data base Here, the desired points in space are down loaded from such a data base, rather than being taught (on the robot) by an operator. This has the advantage of not occupying the robot for teaching of points and also permits the optimization of the path using simulation techniques. In addition, it is also likely that within the next few years artificial intelligence (AI) techniques will permit robot teaching to he more generalized. For example, AI will allow the robot to place filled bottles in a case or pallet, without having to be explicitly taught a predetermined pattern and/or having specific points actually demonstrated by an operator or down loaded from a CAD/CAM system. Before discussing this topic, however, we will consider more standard techniques of demonstrating points to a robot.

There are several methods currently in use. The method employed depends on the manufacturer's specifications, control system software, and the robot's computing/memory capabilities. Teaching typically involves one of the following methods: continuous path, via points, or programmed points. Each of these is now briefly discussed.

4.3.1 Continuous path (CP)

With the CP method, the operator releases all joint brakes and enables an automatic sampler. The manipulator is then manually moved through each of the positions required to perform the task. The controller "remembers" or stores the coordinates of all the joints for every position. In this manner, complex three dimensional paths may easily be followed. Teaching may be done at a speed different from that speed needed for real-time operation (i.e., playback may be set at other speeds, allowing for different cycle times). This method requires minimal debugging, allows for continuous-path programming, and requires minimal knowledge of robotics. However, a thorough understanding of the assigned task is a prerequisite, and editing requires reprogramming from the error point. This method is typically used with robots employed in spray-painting and arc welding applications .

4.3.2 Via points (VP)

Teaching with the VP method does not require that the operator physically move the manipulator; rather, it is remotely controlled by either a computer terminal or, more commonly, a teach pendant—a device similar to a remote control box with the additional capability to record and play back stored commands. The teach pendant is plugged into the controlling computer during programming (the on-line method), and the operator then presses the appropriate buttons to position the arm, with small incremental motion for precise positioning. When the correct position is achieved, a switch is activated to inform the computer to read and store positions for all joints. This process is repeated for every spatial point desired to be "taught." Essentially, only the endpoints of the motions are demonstrated.

The VP method is often employed to program discrete points in space (through which the end effecter is required to pass) and is most commonly used for point-to-point robots. The teach pendant is most commonly used for heavy-duty robots and in those lightweight robots that have sophisticated control systems.

There are more advanced systems that allow for the movements and endpoints to be recorded in an unspecified order. This enables new programs to be created by calling out the points in a sequence that differs from the original order of input, thus facilitating programming and editing. These systems also allow the programmer to define velocity and acceleration or deceleration between points. However, such advanced systems have an inherent danger; that is, the path resulting from a new sequence of movements may inadvertently bring the end effector in contact with nearby machinery. For this reason, manufacturers recommend that once the program is complete, the program should be played back at a very slow speed to minimize the possibility of damage to the robot or other equipment.

4.3.3 Programmed points (PP)

The PP method is also an on-line system. The robot operates via a prerecorded program (i.e., without manual intervention), with the program sequence having been set up externally. Applications of the PP method of using decision making include orienting (i.e., aligning workpieces in designated positions) for assembly operations and material-handling work using conveyers. In addition to the techniques used for programming a robot as described above, there is a new methodology emerging. This is discussed next.

4.3.4 Artificial Intelligence and Robot Programming

The discipline known as artificial intelligence (AI) is becoming more practical as new developments in computer hardware and software evolve. Higher memory density, faster processors, and new languages are bringing the tools of artificial intelligence to practice. There are "expert systems" development environments that execute on nominally priced personal computers, and these are already having an impact in many areas previously the exclusive domain of the human thought process. Experience is showing that in a complex equipment maintenance milieu, in certain classes of medical diagnosis, theorem proving, biochemical analysis, and a plethora of other fields, AI is contributing to productivity. The much touted nationalized Japanese fifth-generation computer project is directed toward creating AI techniques that will reduce software production to a blue-collar job. Whether or not the Japanese will succeed is yet to be determined, but even if the goal is not fully reached, there will be significant technological fallout from the effort.

In the programming of robotic systems, the use of AI techniques is certain to have an impact because of the availability of data base information that can be used to plan a robot's task efficiently. Although there is no integrated system available today, laboratory demonstration such as the assembling of simple structures from randomly presented and available parts is already accomplished. More over, a number of laboratory facilities are currently implementing AI/expert systems in a variety of mobile robots. Intended for use in the nuclear power industry and by the military, these devices are being employed as testbeds for practical results in the areas of autonomous navigation, collision avoidance, maintenance and repair, assembly, reconnaissance, and perimeter monitoring.

SUMMARY

In this chapter we have discussed many topics relevant to computer considerations for robotic systems. The picture presented here is a snapshot of numerous technological considerations that are changing rapidly, and thus the specific material in the chapter may be quickly outdated. The general topics treated here will not become outdated, however, and for this reason one must develop a general set of methods to evaluate new advances in robotic software, communications, cell controllers, and other robotic computer-related subjects. Although the specific robot languages or the specific interface protocol may change, the role that these technological components play will be more or less consistent.

CHAPTER 5. ROBOTIC APPLICATIONS:

5.1 INTRODUCTION

In its relative infancy, the state of the art of robotic applications is, in some ways, paralleling the development of digital computers. When they were first introduced, computers were used for tasks that had previously been performed by people (with perhaps the assistance of some type of manual aid, such as a slide rule or mechanical calculator). This was a natural application, for it was obvious that the new device would be able to perform such jobs much faster and even more reliably than people could perform them. However, as time progressed, it was recognized that tasks that had heretofore been rejected as being impossible to undertake because of excessive manpower and/or time requirements were now possible to attempt. Thus problems that were "not practical" to solve were handled with relative ease. Besides being able to solve such problems, it became apparent that there were many applications for the computer that had never been thought of before its development. In a sense, what happened was that people took off their "blinders" and allowed their imaginations free reign. The result of this has been that computers are now applied in many areas other than the more traditional "number crunching" that was initially envisioned as the major use. The fields of control (of large-scale systems), learning and teaching devices, handling of large data bases, and artificial (or perhaps more descriptive, "autonomous") intelligence come to mind, to name but a few nontraditional applications. But where do we stand with robots?

As already mentioned in earlier sections of this chapter, the first applications of the robot have been in areas where human beings have traditionally been working. Although there have been some significant technological advances in the design of robots (i.e., the hardware) since the first one was developed more than 20 years ago, the manipulators currently being manufactured are, as a general rule, rather simple (e.g., most lack the ability to sense their external or working environment). As a result, the state of the art in robot applications is probably where the computer was when it was used primarily for "computing." It has taken a much longer time for the blinders to be taken off when talking about robots than it did with computers. One can cite a number of possible reasons for this, including the problems of recessions, fear of people losing their jobs, and the lack of a major scientific breakthrough comparable to the development of the transistor and later, the integrated circuit. Also, some of the first big users and/or developers of computers were in government, the military, and the universities. These three entities, which were responsible for developing many of the unique computer applications, have only recently entered the robot field in a large way. (The program at the National Bureau of Standards, having been started in the 1970s, is a notable exception.) The industrial sector has been the major user, and as might be expected, the need to produce a "good bottom-line result" has prevented or at least significantly reduced the risk taking required to produce new ideas (i.e., applications) and developmental research by manufacturers. The recent emergence of robot

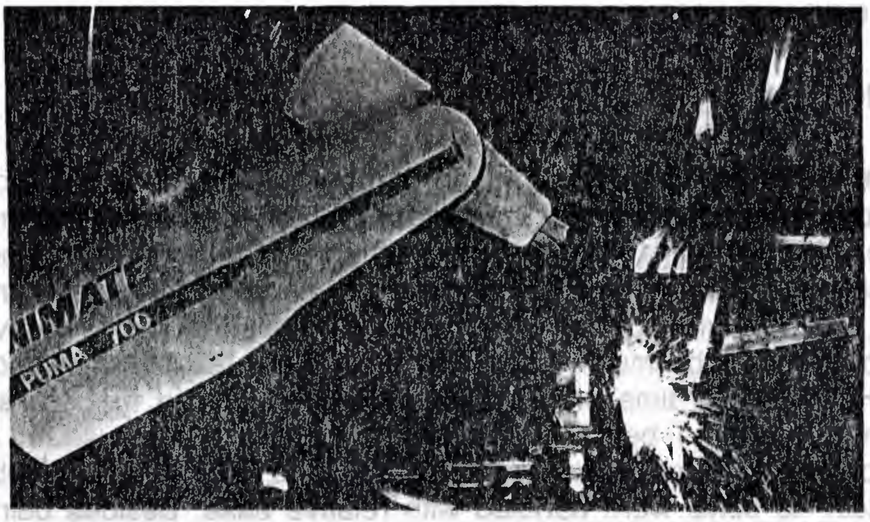


Figure 5-1 A PUMA 700 series robot performing a spot welding operation on an automobile part. (Courtesy of Unimation, Inc., a Westinghouse Company, Danbury, CT.)

programs supported by both the military and state and federal government may indicate that this situation is beginning to change, however. As a consequence, it is to be assumed that over the next few years, nontraditional robotic applications will begin to appear which will, in part, contribute to the development of the fully automated factory or factory of the future.

In the first part of this section we briefly summarize some of the more traditional uses for robots, some of which have already been mentioned in earlier sections of this chapter. In the concluding portion of the section we indicate some of the more futuristic applications that have been proposed by some workers in the field.

5.2 Current Robotic Applications

In the preceding two sections we encountered a number of applications of today's industrial robots. For example, it was indicated that welding, grinding, and spray painting account for the majority of applications of the current generation of robots.

5.2.1 Welding

Welding is one of the major uses for an industrial robot. Actually, two distinct types of welding operations are readily and economically performed by robots: spot and arc welding. In the former case, the robot is taught a series of distinct points. Since the metal parts that are to be joined may be quite irregular (in three dimensions), a wrist with good dexterity is often required (e.g., three degrees of freedom). This permits the welding tool to be aligned properly at the desired weld point without the gun coming into contact with other portions of the part. Typically, the welding tools carried by these robots are large and reasonably heavy. Also, it is usually necessary for the manipulator to have a long reach. As a consequence, large point-to-point servo-controlled robots (either hydraulically or electrically actuated) such as those produced by Cincinnati Milacron (i.e., the T-566), Yaskawa (i.e., the Motoman L3), or General Motors Fanuck (GMF) are normally used for this purpose. The automobile industry is a heavy user of this type of robot (see Figure 5-1). Since the weld points are pretaught, sensory information is generally not required in order to energize the welding gun. It is, however, possible to utilize the increased motor current that results when the tool makes contact with the part to initiate the welding operation.

The second type of welding application, arc welding, is also utilized extensively by the auto industry. Here, an often irregularly shaped seam or a wide joint must be made. In this case, a continuous-path servo-controlled robot that is often specifically designed for this single application is most usually the choice (e.g., the Unimation Apprentice robot). If the parts to be welded can be accurately positioned and held in place, the complex three-dimensional path can be pretaught and no external sensors may be necessary. At present, a number of manufacturers include a position sensor that is placed in front of the welding tool and can therefore provide information concerning irregularities in the weld path. Several manufacturers provide additional sensory feedback among them Automatix and GE. Where a wide joint is to be handled, the robot can be programmed to produce a weave type of motion. This ensures that the weld covers the entire gap. A major advantage of a robotic welder is that the arc time (a

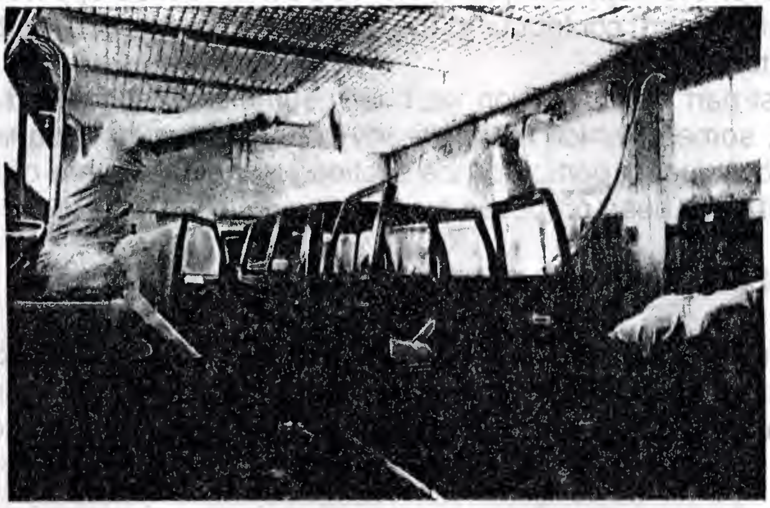


Figure 5-2 Spray painting application at a GM plant in Baltimore, MD. (Used with permission from General Motors Corporation, Detroit, MI.)

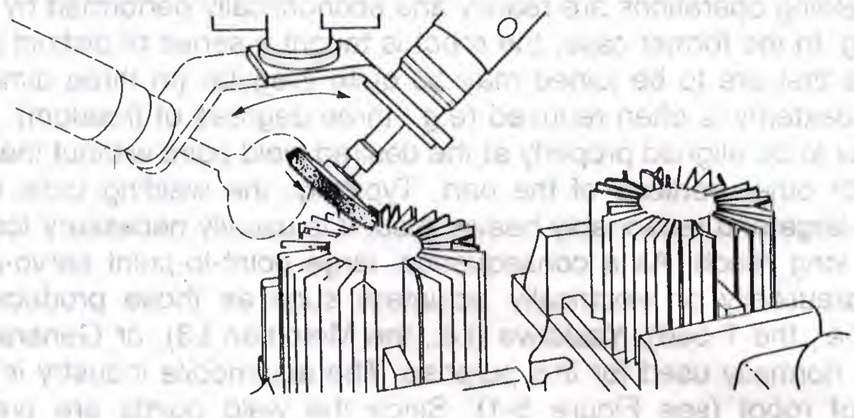


Figure 5-3 Robot used to perform a grinding operation. Depicted here is the smoothing of the top part of large heat sinks. (Courtesy of Unimation, Inc., a Westinghouse Company, Danbury, CT.)

critical parameter in determining the weld's strength) can be carefully controlled.

5.2.2 Spray painting

The spray-painting operation is one that human beings should not perform, both because of the potential fire hazard and the fact that a fine mist of paint (both lead and modern plastic based) is carcinogenic. As such, this task is a natural application for a robot and so it is not surprising that there are a large number of manipulators that perform only this particular job. Another advantage in using a robot for spray painting is that the resultant coating will be far more uniform than a human being could ever produce. This results in a higher-quality product, less reworking of parts, and considerably less paint being used (reductions of 406hG are often achieved). Robots employed for this purpose are usually capable of performing both straight-line and continuous-path motions (see Figure 5-2).

Programming a spray-painting robot is usually performed by the best human operator. His actions are then mimicked by one or more robots. The spray painting application generally does not require the use of external sensors. However, it is necessary that the part to be painted be accurately presented to the manipulator.

5.2.3 Grinding

As a result of arc welding two pieces of metal, a bead is formed at the seam. Where a smooth surface is required for appearance sake (such as on auto bodies) or for functionality (e.g., to maintain necessary tolerance of parts), it is usually necessary to perform a grinding operation. This is also a natural task for a robot since the manipulator can use the same program that was employed in the arc welding operation. All that must be done is to remove the welding tool anal replace it with a rotary grinder (see Figure 5-3).

Another important grinding task is on metal castings. Here the robot is taught the correct shape of the Casting using continuous-path programming. The grinder then removes any undesired high spots and corrects areas of the casting that are too large. A third robotic grinding application is that of deburring. Here the unwanted material that remains around the back side of a drilled hole is ground away to leave a smooth surface. For increased productivity, it is especially important to be able to perform this task automatically after the holes have been drilled automatically (perhaps by a robot).

In these grinding applications, there is always some uncertainty in the dimensions of the part being worked on. As a result, sensory information is often needed to permit the robot to more accurately "feel" the actual contour of the part. This is especially important in the case of smoothing of the arc weld bead. Relatively simple touch sensors that provide this information are currently available. For example, the Swedish company ASEA uses such a sensor with its IRB60 robot.

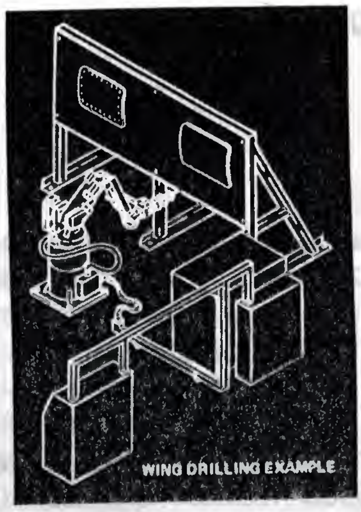


Figure 5-4 A Cincinnati Milacron T³ robot drilling holes in an aircraft wing. (Courtesy J. Coshnitzke, of Cincinnati Milacron, Cincinnati, OH.)

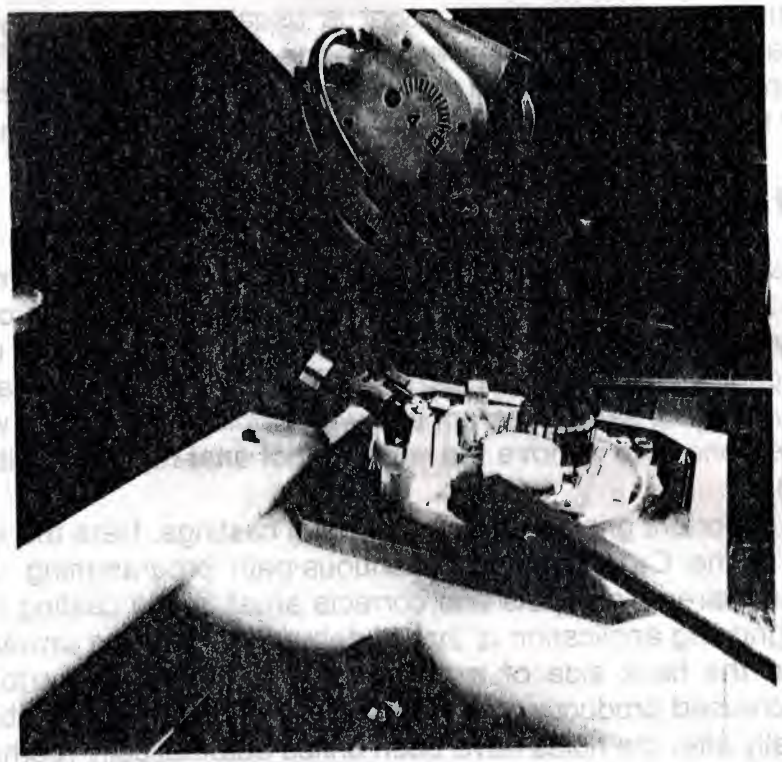


Figure 5-5 The assembling of smoke detectors is accomplished using several United States Robots, Inc., Maker 100, five-axis, servo-controlled robots. (Courtesy of G. Heatherston and U.S. Robots, Inc., a Square D. Company.)

5.2.4 Other applications involving a rotary tool

In addition to the rotary grinding or deburring applications, robots are also currently used for drilling holes, routing, polishing, nut running, and driving of screws. In the first two cases, preprogramming of either points or paths can be performed when extreme accuracy is not required. However, where exact placement of drilled holes is needed (e.g., in the structural components of aircraft), it may be necessary to utilize a template (see Figure 5-4). The difficulty with doing this is that unless the robot wrist has some "give" (i.e., compliance), any misalignment of either the part or the robot itself will result in a damaged template and/or an inaccurately placed hole. This problem is overcome by means of a compliant wrist which permits the drill bit to be aligned in the template hole even if there is a positional error.

5.2.5 Assembly operations

Human beings are capable of assembling a group of diverse parts to produce either a finished product or a subassembly because of their ability to utilize good eye-hand coordination in conjunction with the important sense of touch. However, these jobs may be extremely tedious because of their repetitious nature. As such, assembly operations represent an attractive application of robots. For example, consider the assembly of smoke detectors shown in Figure 5-5. Here, although not shown, a group of servo-controlled robots (e.g., U.S. Robots' Maker 100) is actually used. First, the finished printed circuit board is acquired and then is loaded into the bottom portion of the plastic case. Next, a 9-volt battery (with its terminals reversed to increase shelf life) is inserted into the battery compartment. Finally, the top portion of the plastic case is placed onto the finished bottom assembly. It should be noted that this last operation also requires that the robot exert a downward pressure so as to ensure proper locking of the two parts of the case. The finished detector is then stacked in a carton utilizing a palletizing program.

Other assembly applications performed by robots include putting together scissors, pliers, and other simple hand tools, the fabrication of small electric motors, and the assembling of electrical plugs and switches. In most of these examples, the robot is taught the desired points and the sequence of operations. The only external sensory information that is normally utilized is whether or not a part or subassembly is at a particular location within the work cell. (Such an indication can be obtained using simple optical interrupters or mechanical switches, as discussed).

As mentioned above, some applications depend on the robot wrist being compliant. This is especially important in certain assembly operations, for example, insertion of shafts or rods into small clearance holes or the screwing of a screw into a threaded hole. To prevent the binding and/or bending of the rods or cross threading of the screws, an RCC is often used between the end effecter and the robot's wrist flange. Alternatively, force and/or tactile feedback can be utilized to provide better external sensing capability, thereby permitting the robot to adapt better to any positional errors caused by either the devices which hold and/or position (i.e.. "present") parts or by the

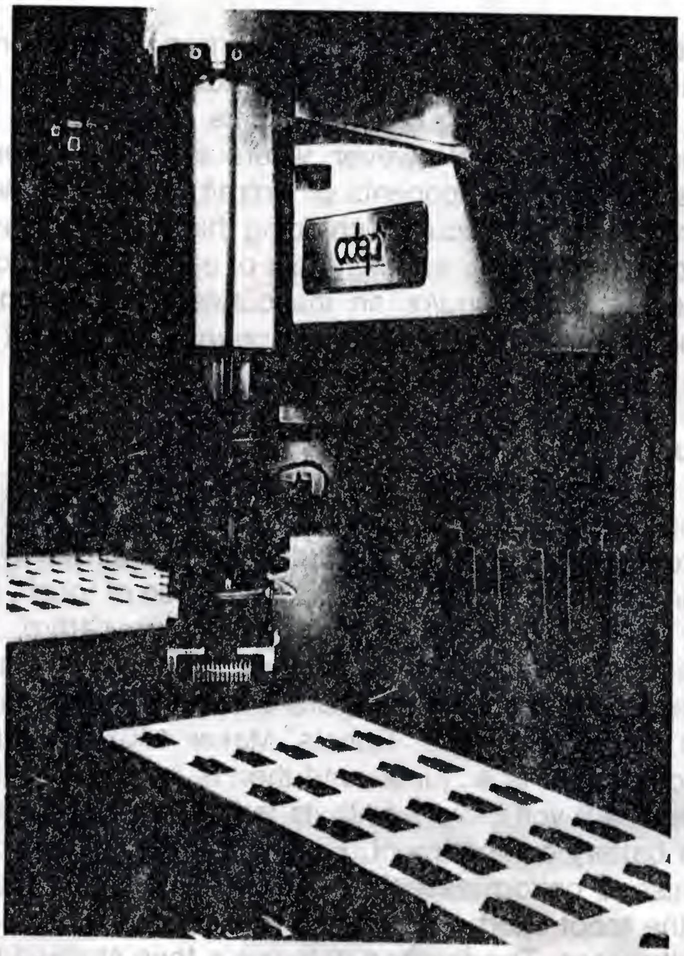


Figure 5-6 A vision system is used with an Adept robot to perform printed circuit board assembly. (Courtesy of Adept Technology Inc., San Jose, CA.)

robot itself. However, such sensing is, for the most part, not well developed, so most assembly applications arc currently geared toward those that either do not require external sensing or else can be performed with an RCC device.

A number of assembly applications do not require the use of a compliant wrist (e.g., electronics assembly). In this case it is necessary to insert a variety of electronic components (e.g., resistors, capacitors, etc.) into a printed circuit (PC) board. As the leads on these components are easily bent, extremely accurate placement before insertion is usually required. Although human beings can perform these operations, the work is tedious and repetitious, with the result that mistakes are often made. Thus a robot is a good choice for this task. However, the high degree of accuracy demands that the manipulator be equipped with an external sensor (e.g., a vision system, see Figure 5-6). Although vision peripherals tend to reduce system throughput, it is expected that such applications will become more common as the cost of vision hardware and software drops and the systems themselves become faster. In fact, this is already happening.

SUMMARY

In this fairly detailed, nontechnical introduction, we have attempted to give the reader an understanding of what an industrial robot is and what it is not, where it is applicable and where it is not, and finally, how such devices have evolved and how they may cause another industrial revolution to occur. In particular, the reader has been introduced to most of the terminology associated with these devices and has been shown how to categorize them either by geometry of their major axes or by the type of control utilized. In addition to tracing the development of robots historically, the economic and sociological consequences of these forms of automation have been discussed. Finally, the current and possible future applications of robots have been presented.

It should be apparent from the material contained in this chapter that there exist a wide variety of manipulators and that they can perform a large number of tasks. Moreover, as vision and tactile sensors are incorporated and the controllers become "smarter." the complexity of these tasks will no doubt increase. Applications that were not originally envisioned and involve more than just replacing a human worker with a robot will then be feasible. To be sure, there will be an impact on some workers, who, unfortunately will be displaced by these machines. However, it is expected that in the longer term, more jobs will be created as new and expanded industries are developed as a direct consequence of this new, more flexible form of automation, the robot.