5. Sensors and Intelligent Robots

5.1. ARTIFICIAL INTELLIGENCE AND AUTOMATED MANUFACTURING

Artificial intelligence (AI) is concerned with the intelligent characteristics present in human behaviour. A human being with his intuition and learning capability understands language, reasons and solves problems. Imparting all these intelligent characteristics to the machines is really difficult. In automated manufacturing, attempts have been made to design machines that can perform tasks intelligently. AI techniques are therefore the key to automation in the field of manufacturing. The principles of AI viz. learning, reasoning and planning have been employed primarily in three broad application areas:

1. Development of machine vision and image processing
2. Development of computer systems to enable understanding natural (written and spoken) language
3. Development of expert systems to make decisions and perform tasks without any hindrance.

The basic problem in AI is the technique of data or knowledge representation and its manipulation. Two most important techniques are:

1. Logic formation or developing logic programming in which inferences may be made from the facts. PROLOG and LISP are some of the suitable programming languages for solving problems in the field of AI research.
2. Semantic networks in which nodes represent the objects or situations and lines or arcs represent the relationship between different nodes. The network of nodes helps to represent knowledge and information.

Once knowledge is gathered and represented, a search for an optimal solution to reach the desired goal is attempted. There are different search techniques and the techniques depend on the situations to be handled.

In the manufacturing scene, the primary application of natural (written) language lies in the development of data base management systems (DBMS). At present, records of (say) materials, shipping of goods and sales etc, are all handled in DBMS which requires some specialized programming language. Plant managers cannot use natural language, or type at the 'front end' of the computer simple questions like, 'what is the present position of inventory' or 'How many machines were sold on June 15, 1990?' and get a prompt reply. However, spoken language can be processed by microphones and the signal patterns are compared to the stored recognizable voice signals. The computer system is trained to listen to these spoken words and can recognize the command. Instead of using a keyboard, one can also give commands to computer in spoken language. Though their applications are as yet limited to the laboratory stage, such computers have great potential.

Expert systems on the other hand allow the computers to make decisions. Expert systems use hundreds of rules and employ them to make their judgement. They provide or refine knowledge and can generate many intelligent actions by way of reasoning. One of the classic expert systems was Stanford's MYCIN which advises medical practioners to use antibiotic drugs during treatment of their patients. MYCIN uses rules gathered from the expertise of doctors. The rules use IF and THEN statements. For example:

$$\text{IF} \qquad \left.\begin{array}{l}(1)\ \text{this is}\cdots\\ (2)\ \text{this is}\cdots\\ (3)\ \text{this is}\cdots\end{array}\right\}\quad \text{say, organism characteristics}$$

THEN there is an evidence of ... (organism identified) Advice From MYCIN Expert System:

[Rec. 1:] My preferred therapy recommendation is

—(a specific antibiotic)

In the field of robotics, Charles Rosen of Standard Institute used AI and machine intelligence in his project SHAKEY that was a mobile robot with range sensing devices and camera for vision. Solving an assembly task using robot vision and other suitable sensors is another practical example of AI that attempts to perform scene analysis and identify objects in the field of camera view. Learning from experience is an important attribute of machine intelligence. Learning strengthens knowledge base. Development of world modelling and task level robot languages are some of the attempts in this direction. However, applicability of natural language for machines, and intelligent machines are still long-term research issues.

5.2. AI AND ROBOTICS

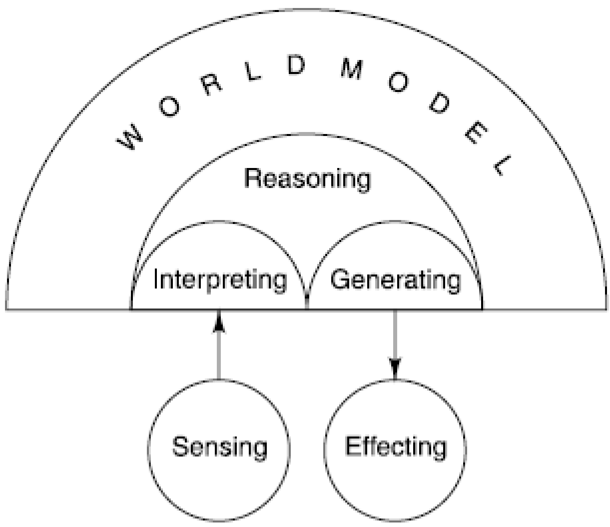
A manufacturing process is basically a stochastic process. An operator uses his arms, hands, senses and brain to perform operations like grasping, holding, orienting, inserting, aligning, fitting, screwing and turning of workpieces of various shapes and sizes. In small and medium batch production, programmable automation is adopted. Robots are important aids in programmable automation. For precision, a robot has to interact with the environment around it in a manner similar to man. Therefore, a robot should be intelligent if it has to emulate human capabilities.

An intelligent robot has servoed arm and end-effector, has sensors and adaptive control functions with the help of a computer. Adaptive control is necessary to correct the errors in position and orientation of the workpieces and the end-effector. An intelligent robot must determine cause and effect phenomena; so it must detect the faults and minimize their effects.

What is important for a smart robot is the need for brain and sensory systems so as to be able to think and then move and manipulate. The thinking processes such as brain function are performed by a computer. Sensing and effecting are the body functions that can be performed using the basic laws and axioms of physical science, production engineering, electrical engineering and computer science and engineering. In order to accomplish a task, both the brain and the body functions are to be coordinated. So a smart robot must have artificial intelligence that will differentiate the robot from just another machine.

Figure 5.1 indicates a simplified model of Artificial Intelligence (AI) and Robotics. The model emphasizes the intelligent functions performed.

Figure 5.1 Simplified model of AI/Robotics



Sensing includes seeing, hearing, touching, smelling and measuring. The sensors gather and produce information. They have little ability to reason about it.

Effecting can be done by actions. The action can be accomplished by manipulators using body, arm, wrist, hands, fingers, legs, wheeled vehicles (mobile robot) and with various means of communication. For example, effecting can be done by communicating sounds, displaying pictures and graphics.

The components of interpreting, generating and reasoning are necessary to acquire knowledge about the environment. These components in fact, recognize, locate and assemble the objects and may direct the changes in the environment.

Interpreting information is a means to understanding the environment. However, interpreting information in the proper context is necessary. Generating function is a means to influence the environment. Reasoning is a means to cope with unforeseen, incomplete, uncertain and, perhaps, conflicting information to act or react to the environment.

5.3. NEED FOR SENSING SYSTEMS

Human operators are constantly receiving and acting upon large amounts of sensory feedback. It is via such sensory feedback that human operators are able to make sense of their surroundings and perform both simple and complex tasks. It also enables human operators to make sense of uncertain situations as they are encountered, and adapt themselves accordingly. More than this, the human operator is capable of combining various types of sensory feedback to perform the most complex of tasks.

It may appear that if industrial robots are to be capable of performing the same types of tasks without constant human supervision, they too must be equipped with sensory feedback. The feedback from various sensors is then analyzed via a digital computer and associated software. The use of sensing technology to endow machines with a greater degree of intelligence in dealing with their environment is an active topic of research and development in the robotics field. A robot that can 'see' and 'feel' is easier to be trained in the performance of complex tasks. There are many reasons for employing sensory feedback in robots. The first is to provide positional and velocity information concerning the joint, arm and end-effector status, position, velocity and acceleration. Usually this type of feedback is provided continuously and becomes an integral part of the physical robot control system. This identifies a category of sensors known as internal sensors. Other reasons are to prevent damage to the robot itself, its surroundings and human operators and to provide identification and real time information indicating the presence of different types of components and concerning the nature of tasks performed. These involve various sensory devices to suit the needs of the particular robot task being carried out and the characteristics of the working environment. They are collectively known as external sensors and are interfaced to the robot control system via input/output facilities provided by the robot control system hardware and software.

As the tasks to which industrial robots are applied become more demanding, there is an increasing need for sensory feedback. Sensory feedback is needed whenever there is uncertainty and unpredictability in the tasks being performed.

5.4. SENSORY DEVICES

A sensor is a transducer used to make a measurement of a physical variable. Any sensor requires calibration in order to be useful as a measuring device. Calibration is the procedure by which the relationship between the measured variable and the converted output signal is established.

Care should be taken in the choice of sensory devices for particular tasks. The operating characteristics of each device should be closely matched to the task for which it is being utilized. Different sensors can be used in different ways to sense the same conditions and the same sensors can be used in different ways to sense different conditions. Five important characteristics of any sensing device are:

Range This refers to the minimum and maximum change in input signal to which the sensor can respond. The sensor should possess a wide operating range.

Response The sensor should be capable of responding to changes in the sensed variable in minimum time. Ideally, the response should be instantaneous.

Accuracy The accuracy of the measurement should be as high as possible. The output of the sensing device should properly reflect the input quantity being measured or sensed.

Sensitivity It refers to the change in the output exhibited by the sensor for a unit change in input. The sensitivity should be as high as possible.

Linearity The sensory device should exhibit the same sensitivity over its entire operating range.

The following are the other considerations.

1. The device should not disturb or have any effect upon the quantity it senses or measures.
2. The device should be suitable for the environment in which it is to be employed.
3. Ideally, the device should include isolation from receiving excess signals or electrical noise that could give rise to the possibility of misoperation or damage of the sensor, circuit or system.
4. Also important are the physical size, cost and ease of operation.

5.5. TYPES OF SENSORS

Sensors in robotics may be broadly classified as contact and non-contact sensors.

As the name implies, the former class of sensors respond to physical contact, such as touch, slip, force and torque. Non-contact sensors on the other hand rely on the response of a detector to variations in acoustic or electromagnetic radiation. The most prominent examples of non-contact sensors are range, proximity, and vision sensors.

5.5.1. Contact Sensors

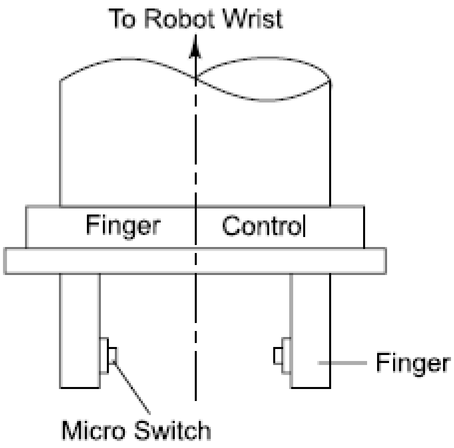
Contact sensors are devices used in robotics to obtain information associated with the physical contact between a manipulator hand and objects in workspace. Contact information can be used, for example, for object location and recognition and to control the force exerted by a manipulator or a given object. Contact sensors can be subdivided into two principal categories: binary or touch sensor and analog or force sensors. Binary sensors, also called touch sensors, are basically switches that respond to the presence or absence of an object. These sensors provide a binary output signal which indicates whether or not contact has been made with the object. Analog sensors, also called force sensors, on the other hand provide an output signal proportional to a local force. They indicate not only that contact has been made with the object, but also the magnitude of the contact force between them.

5.5.1.1. Touch Sensors

Touch sensors are used to indicate that contact has been made between two objects without regard to the magnitude of the contacting force. Included within this category are simple devices such as limit switches, microswitches, etc.

In the simplest arrangement, a switch is placed on the inner surface of each finger of a manipulator hand as illustrated inFig. 5.2.

Figure 5.2 Robot hand with microswitches



Multiple binary touch sensors can be used on the inside or outside surface of each finger to provide further tactile information. The latter use is for providing control signals useful for guiding the hand throughout the workspaces and is analogous to what humans do in feeling their way in a totally dark room.

5.5.1.2. Position and Displacement Sensors

Position and displacement sensors are used as components of the robot control system. The control structure of a robot needs to know the position of each joint in order to calculate the position of the end-effector thus enabling the successful completion of the programmed task. The movements of the joints can be either linear or angular (rotary) depending on the type of robot.

Future developments in the robotics field may allow the use of external sensors to actually measure the end-effector position in relation to its surroundings, thereby dispensing with the need to calculate it from the joint positions. But, for the present, internal position sensors remain the most accurate and reliable way of determining the end-effector position within a robot control structure. There are two main types of position sensors: absolute and incremental, the latter being also called displacement sensors. Some common devices which are used as position sensors are discussed below.

Potentiometers Potentiometers are analog devices whose output voltage is proportional to the position (linear or angular) of a wiper. Potentiometers may be either linear or angular. Figure 5.3 illustrates a typical angular potentiometer, which has a resistive element and a rotating wiper. When voltage is applied across the resistive element, the output voltage between the wiper and the ground is proportional to the ratio of the resistance on one side of the wiper to the total resistance of the resistive element, which essentially gives the position of the wiper. The function of a potentiometer can be represented by the function,

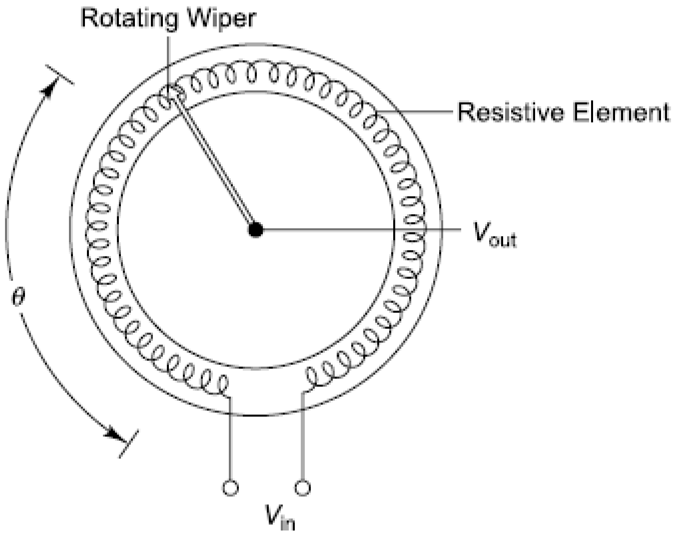
$$V_o = K \cdot \theta$$

where V_o = output voltage

K = voltage constant of the potentiometer in volts per radian (for angular pot) or volts per mm (for linear pot)

θ = position of the pot in radian or mm

Figure 5.3 A potentiometer



Potentiometers are relatively inexpensive and easy to apply. However, they are temperature sensitive, a characteristic that also affects their accuracy. The wiper contact is another limiting factor, being subject to wear and producing electrical noise.

Encoders Encoders which are non-contact type position sensors are classified into two basic types: incremental and absolute. Unlike potentiometers which give analog signals, encoders give digital signals directly.

In a simple incremental encoder, a disc is encoded with alternating transparent and opaque (light and dark) stripes aligned radially; a photo transmitter (light source) is located on one side and a photo receiver (photo cell) on the side of the disc. As the disc rotates, the light beam from the transmitter is alternately passed and broken which is detected by the receiver, whose output is a pulse train having frequency proportional to the speed of rotation of the disc. There are usually two sets of photo transmitters and receivers aligned 90° out of phase to provide direction information. By counting the number of pulses and by adding or subtracting based on the direction, it is possible to use the encoder for position information with respect to a known starting position.

Absolute encoders with which position can be known in absolute terms (i.e., not with respect to a starting position) employ the same basic construction as incremental encoders except that there are more tracks of stripes and a corresponding number of transmitters and receivers. The stripes are usually aligned to provide a binary number proportional to the shaft angle and the angle can be read directly from the encoder without any counting. The resolution of an absolute encoder depends on the number of tracks (n) and is given by,

$$\text{resolution} = 2^n$$

A 3-bit encoder is illustrated in Fig. 5.4. The binary code at each of the eight (0–7) radial positions is generated by a unique combination of ON's and OFF's of photocells. The resulting code is shown in Table 5.1 (1 represents ON and 0 represents OFF). In practice, malfunctions may occur if the photo cells become skewed from the radial line. At the transition points, between numbers 1 and 2, 3 and 4, and 5 and 6, a number of photocells change their states at a time, which can be another source of malfunction. To overcome this problem, the natural binary code is modified so that at any transition position, a change in only one binary digit is necessary. The resulting code is known as the Gray Code. Gray coded disc increases reliability but requires the use of additional decoder circuitry.

Figure 5.4 An encoder

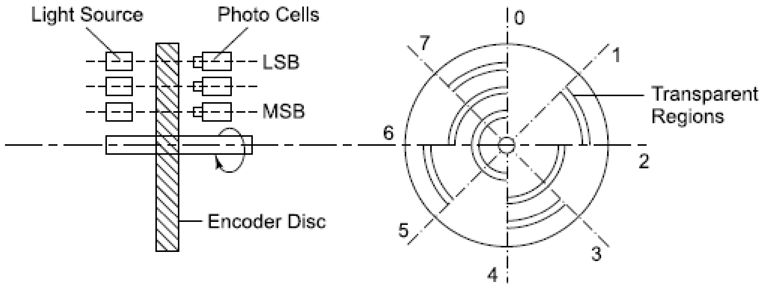
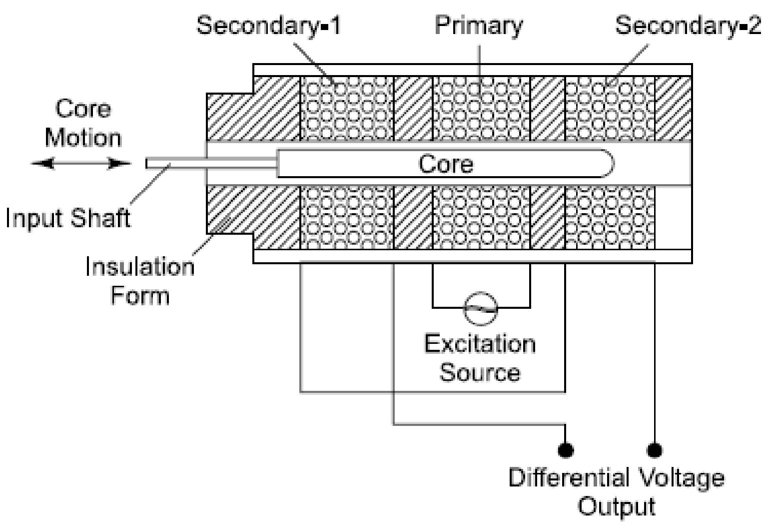


Table 5.1 Decimal Numbers and their Corresponding Binary and Gray Codes

Decimal number	Natural binary code	Gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

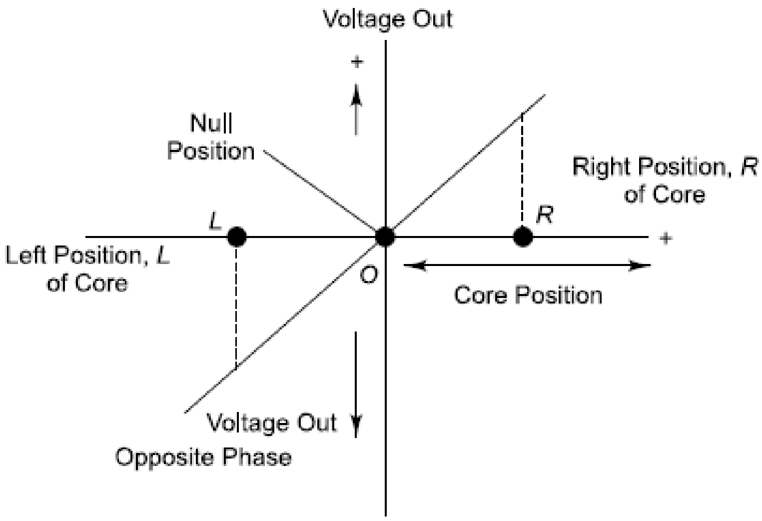
LVDT The linear variable differential transformer (LVDT) is another type of position sensor, whose construction is shown in Fig. 5.5. It consists of a primary, two secondaries, and a movable core. The primary is excited with an a.c. source.

Figure 5.5 LVDT construction



When the core is in its exact central location, the amplitude of the voltage induced in secondary-1 will be the same as that in secondary-2. The secondaries are connected to cancel phase, and the output voltage will be zero at this point. Figure 5.6 illustrates the nature of output voltage as the core is moved to the left or to the right. The magnitude of the output voltage is shown to be a linear function of core position, and the phase is determined by the side of the null position on which the core is located. Finally, the a.c. output of LVDT can be converted to d.c. using rectifiers.

Figure 5.6 LVDT output voltage versus core position



5.5.1.3. Force and Torque Sensors

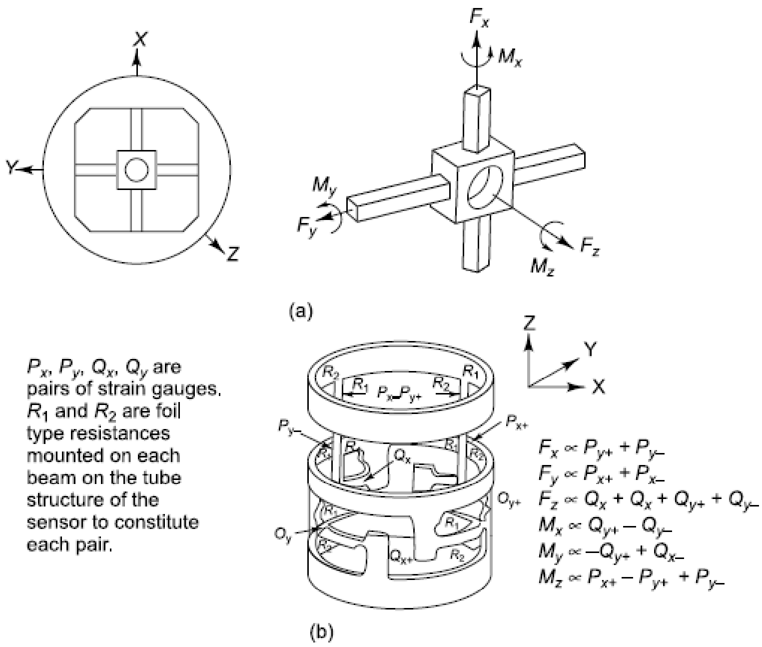
The capacity to measure forces permits the robot to perform a number of tasks like grasping parts of different sizes in material handling, machine loading and assembly work applying the appropriate level of force for the given part. Force being a vector quantity must be specified both in magnitude and direction.

Force sensing in robotics can be accomplished in several ways. A commonly used technique is wrist sensing, in which the sensors (like strain gauges) are mounted between the tip of a robot arm and the end-effector. Another technique is joint sensing, in which the sensors measure the cartesian components of force and torque acting on a robot joint and adds them vectorially. For joints driven by d.c. motor, sensing is done simply by measuring the armature current for each of the joint motors. Finally, a third technique is to form an array of force sensing elements so that the shape and other information about the contact surface can be determined.

Wrist Sensors The purpose of a force sensing wrist is to provide information about the three components of force (F_x, F_y and F_z) and the three moments (M_x, M_y and M_z) being applied at the end of the arm. Based on sensory information and calculations carried but by the robot controller to resolve the forces and moments into their six components, the robot controller can obtain the exact amount of forces and moments being applied at the wrist, which can be used for a number of applications. As an example, an insertion operation (e.g., inserting a peg into a hole in an assembly operation) requires that there should be no side forces being applied to the peg. Another example is where the robot's end-effector is required to follow along an edge of an irregular surface. The robot equipped with a force sensing wrist plus the proper computing capacity could be programmed to accomplish these kinds of applications. The calculations required to utilize a force sensing wrist are complex and require considerable computing time.

Wrist sensors are small, sensitive, light in weight and relatively compact in design. Most wrist force sensors function as transducers for transforming forces and moments exerted at the hand into measurable deflection or displacements at the wrist. As an example, the sensor shown in Fig. 5.7(a) uses eight pairs of semiconductor strain-gauges mounted on four deflection bars—one gauge on each side of a deflection bar. Since the eight pairs of strain gauges are oriented normal to the *X*, *Y* and *Z* axes of the force coordinate frame, the three components of force, *F* and three components of the moment, *M* can be determined by properly adding and subtracting the output voltages respectively. An SRI design of force-torque wrist sensor as shown in Fig. 5.7(b) can measure the three components of force and the three components of torque in a cartesian coordinate frame.

Figure 5.7 Strain gauge type wrist force sensors



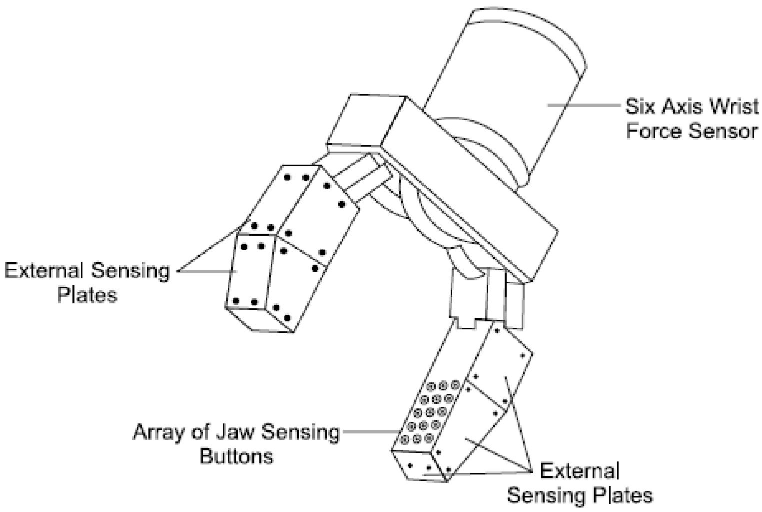
Joint Sensors Many robots are powered by d.c. servo motors, for which the output torque is linearly related to the armature current in the motor. Thus, it is possible to measure the torque of each robot joint (indirectly) by inserting a suitable series resistor in one lead of each servo motor and measuring the voltage across it which is proportional to the motor current and, hence is related to motor torque.

Motor current measurement is simple and inexpensive, but is not without drawbacks. The accuracy is affected by any friction in the motor bearings, associated gears and joint bearings. The measurements not only reflect the forces being applied at the tool but also the forces and torques required to accelerate the links of the arm and to overcome the friction and transmission losses of the joints.

Tactile Array Sensors During the past few years, considerable efforts have been devoted to the development of tactile sensing arrays capable of yielding touch information over a wider area than afforded by a single sensor. A tactile array sensor is a special type of force sensor composed of a matrix of force sensing elements. The force data provided by this type of device may be combined with pattern recognition techniques to describe a number of characteristics about the impression in contact with the array sensor surface, like (i) the object's presence, contact area, shape, location and orientation, (ii) the pressure and pressure distribution, etc. Tactile array sensors can be mounted in the fingers of the robot gripper or attached to a work table as a flat touch surface.

Figure 5.8 shows a robot hand in which the inner surface of each finger has been covered with a tactile sensing array. The external sensing plates are typically binary devices, which are fitted to the outer jaw surfaces. The inner face has a matrix of touch buttons to sense the workpiece. The outer sensors may be used to detect unavoidable obstacles and prevent damage to the manipulator. The inner mounted tactile sensor array is useful in getting information about an object before it is acquired at a particular location. The array may be composed of multiple individual sensors. The force on each sensor acts against washer displacing a vane that controls the amount of light received by a phototransistor from an LED. The tactile array can be made of artificial skin.

Figure 5.8 Robot hand with tactile sensors



Various plastic materials and synthetic rubber are used to produce materials with elastic properties, which are called elastomers, and can be used to make skin for a robot gripper. Conductive elastomers can provide feedback signals which are proportional to the acting forces. This device which is often called artificial skin, is typically composed of an array of elastomer pads as shown in Fig. 5.9(a). As each pad is squeezed by any object pressing against the surface, its electrical resistance changes in response to the amount of deflection in the pad, which is proportional to the applied force. By measuring the resistance of each pad, which is easily transformed into electrical signals, information about the shape of the object pressing against the array of sensing elements can be determined. As the number of pads in the array is increased, the resolution of the information obtained improves. Figure 5.9(b) indicates two perpendicularly intersecting electrodes with conducting materials in between. Change of resistance due to pressure causes electrical current to flow.

Figure 5.9 Artificial skins

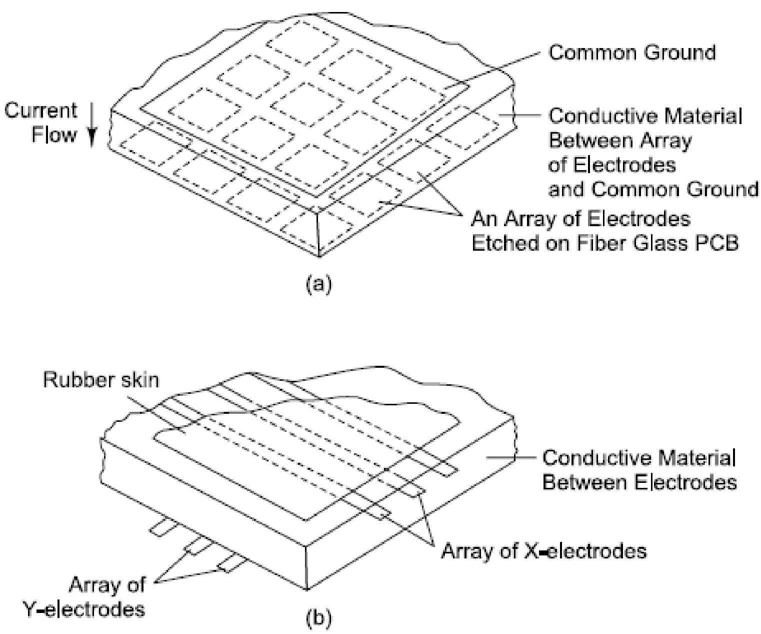
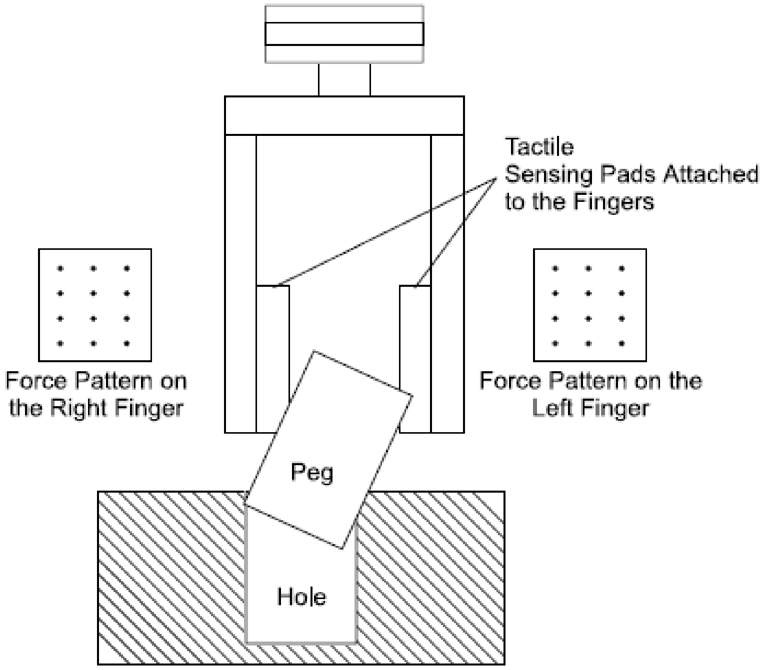


Figure 5.10 shows a possible use of tactile array sensors to measure the force and moments being applied to an object by the robot in an application in which a robot is inserting a peg into a hole. The figure illustrates the situation showing the likely pattern of forces on the surface of a tactile array sensor.

Figure 5.10 Tactile array force sensors for assembly



5.5.1.4. Slip Sensor for Robot Grippers

Figure 5.11 indicates a scheme of a sensor based robot gripper. The fingers are closed on the object to secure gripping by means of a lever actuated with the wrist of a robot. A full bridge with electrical strain gauges on the lever measures strain due to the effort required to close the finger and the gripping force can be determined with proper calibration. A specially mounted, rubber padded and spring loaded wheel in contact with the upper surface of the object measures the degree of slip through the positional rotation of a potentiometer. Dead weights are placed on the weighing pan to induce slip between the fingers and the object being gripped. The movement of the slider point of the potentiometer varies with the slip and analog voltage signal obtained is digitized and fed to a microprocessor. When there is a slip, the microprocessor detects it and sends a high value signal through an open collector buffer to the I/O module of the robot controller. The system components comprising slip sensor, force sensor, and robot controller are shown in Fig. 5.12.

Figure 5.11 Slip sensor-based robot gripper

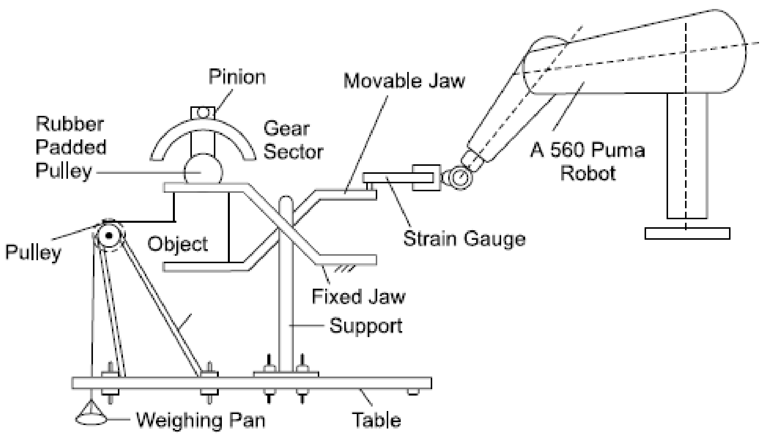
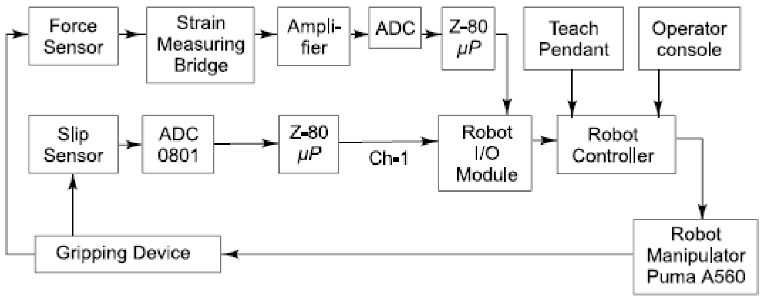


Figure 5.12 *Slip sensor system components*



5.5.2. Proximity and Range Sensors

The presence of an object can be sensed by a proximity sensor. Photoelectric proximity sensors may control the motion of a manipulator arm. Range sensors are used to sense and measure the distance between the objects and the sensing device and they may be used even to locate the workpiece in the robot workcell. Proximity and range sensors may be located on the end-effector or wrist.

There are various techniques that may be employed for designing proximity sensors. They include optical devices, acoustics, eddy currents, magnetic fields etc.

A photoelectric proximity sensor that senses the presence of an object without making physical contact is indicated inFig. 5.13. It consists of a solid state LED which acts a transmitter of infrared light and a solid state photodiode which acts as a receiver. Both are mounted in a small package. The sensing space is approximately the intersection of two cones in front of the sensor. If the reflectance and incident angle are fixed, the distance may be measured with suitable calibration. When the received light exceeds a threshold value, it corresponds to a predetermined distance.

Figure 5.13 *Proximity sensor*

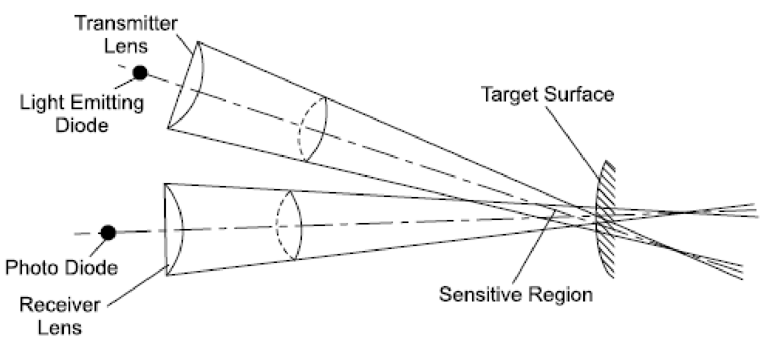
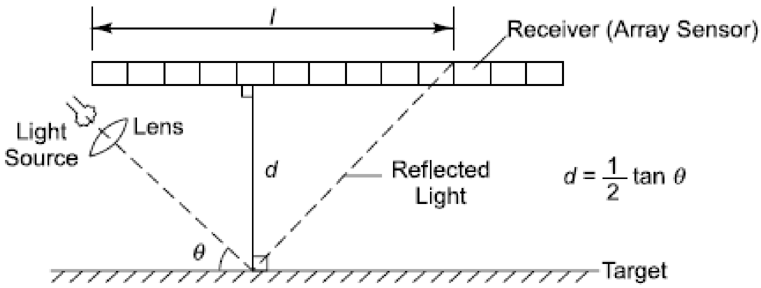


Figure 5.14 indicates a proximity sensor that locates a part. The distance between the target and the array of light sensors is given by d = 1/2 tan θ. The surface of the target is parallel to the sensing array.

Figure 5.14 Proximity array sensor



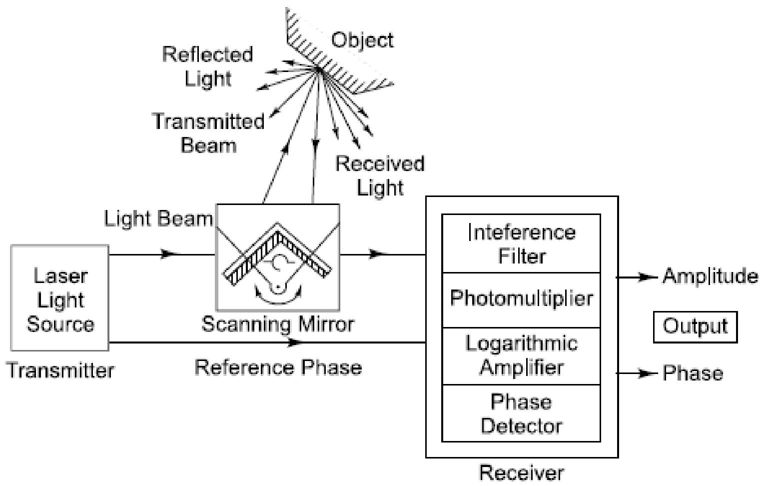
Proximity sensors based on electrical fields are commercially available. The sensing device when brought near the object creates an alternating magnetic field in a small region and this field induces eddy currents through the conducting object. The eddy currents produce their own magnetic field which interacts with the primary field. This changes the flux density and indicates the presence of the object.

Proximity sensor using magnetic field is composed of a reed switch and a permanent magnet. The sensor when brought near the target object activates the reed switch as the magnetic circuit is completed between the object and the sensor.

5.5.2.1. Range Imaging Sensor

Bats use range sensor for the purpose of navigation. A typical range imaging sensor uses a laser scanner which is classified into two basic schemes: One based on transmitting a laser pulse and measuring the time of arrival of the reflected signal and the other based on transmitting an amplitude modulated laser beam and measuring the phase shift of the reflected signal. The transmitted beam and the received light are essentially coaxial. The principle of range sensor is shown in Fig. 5.15.

Figure 5.15 Range imaging sensor

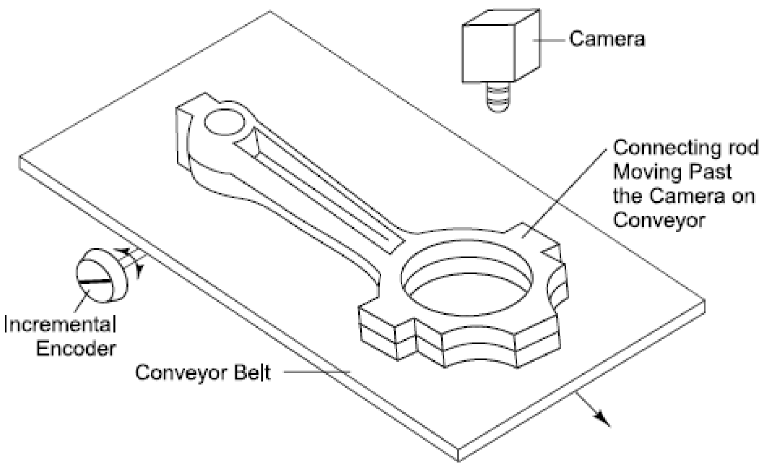


5.5.3. Electro Optical Imaging Sensors

Electro optical imaging sensors use solid state cameras interfaced with a vision processor. The cameras scan a scene, measure the reflected light intensities within a raster of say 128 × 128 pixels, convert these intensity values to analog and then binary electrical signals and feed the stream of information serially into the processer. The signals are stored in the computer's memory and processed in real time with consequent reduction of memory requirements.

Figure 5.16 indicates a connecting rod that moves past the viewing station and the camera can be triggered from the position sensor (incremental encoder) coupled to the moving conveyor belt. The other type of conveyor tracking sensor may also be used.

Figure 5.16 The connecting rod moving on a conveyor belt

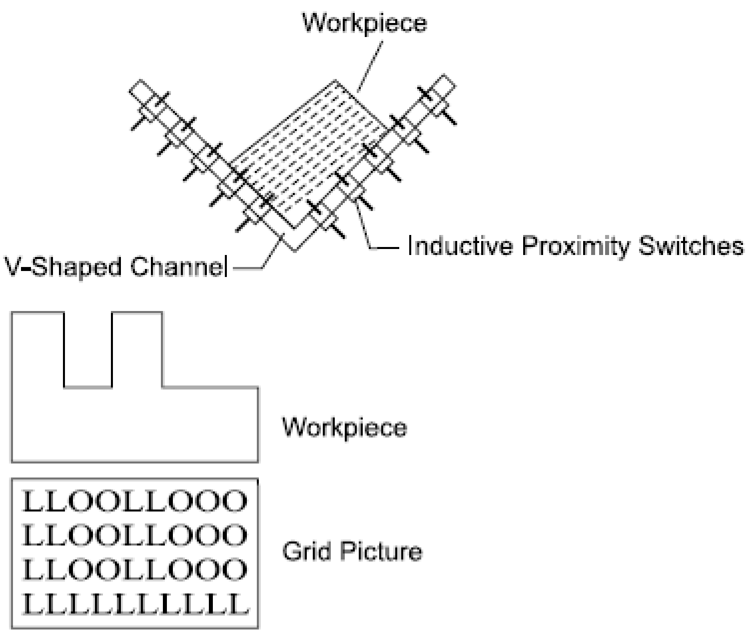


A robot works intelligently with the help of sensors and a good computer network. An intelligent robot can sense, effect, interpret, generate and reason to arrive at a logical decision. Unfortunately we are not in a position to understand the fundamentals of many intelligent functions that a man or even an animal has. The other important issue is economy. One has yet to leam to make sensible use of AI and Robotics. However, the challenge is still there, to understand the fundamentals, and design an intelligent and smart robotics system.

5.5.3.1. Inductive Proximity Switch

Figure 5.17 gives a coarse contour image of the metallic workpiece. The switches scale the workpiece in two planes simultaneously and a grid picture may be generated.

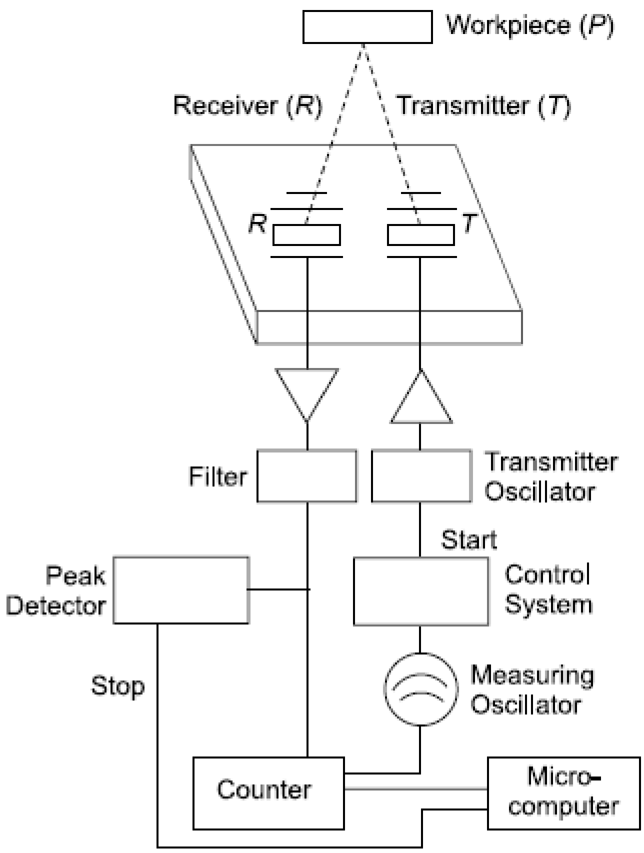
Figure 5.17 Contour image of the workpiece



5.5.3.2. Acoustic Sensor

Acoustic sensors can also recognize the workpieces like an optical system. A system for acoustic object recognition is shown in Fig. 5.18. A distant picture can be constructed from the transit time between the transmitted sound and received sound.

Figure 5.18 Acoustic transducer for object recognition



5.5.3.3. An Opto-electronic Sensor for Generating the Contour Picture of a Workpiece

Contour generation (pattern recognition) is an important step towards shape recognition. A linear array of opto-electronic sensors consisting of photo transistors acting as detectors and a point source of light for sensing brightness and darkness patterns in 2–D on the surface of the workpiece under transmitted illumination is shown in Fig. 5.19. A binary matrix representing the object pattern in 2–D is obtained. A computer software generates the contour of the workpiece from this binary matrix. A parallel beam of light falling on an object placed on a transparent sheet of glass creates a pattern of light and shadow corresponding to the 2–D object pattern. The light signals are converted to the corresponding voltage signals with the help of a suitably designed electronic circuit. A linear array of sixteen phototransistors has been employed to scan the object. The light falling on the base of each of the photo transistors makes it ON to provide a path for the current to flow from its collector to emitter. The current is amplified by an n-p-n transistor connected in Darlington configuration with each of the photo transistors. The voltage at the common collector of each element will be low or high depending on whether the transistor is 'ON' or 'OFF. The voltage signals from the different elements are fed to a microprocessor system through different I/O lines of PIO port.

Figure 5.19 Opto-electronic sensor (a) Schematic diagrams of sensor systems hardware for contouring (b) Network of entire sensor systems for contouring

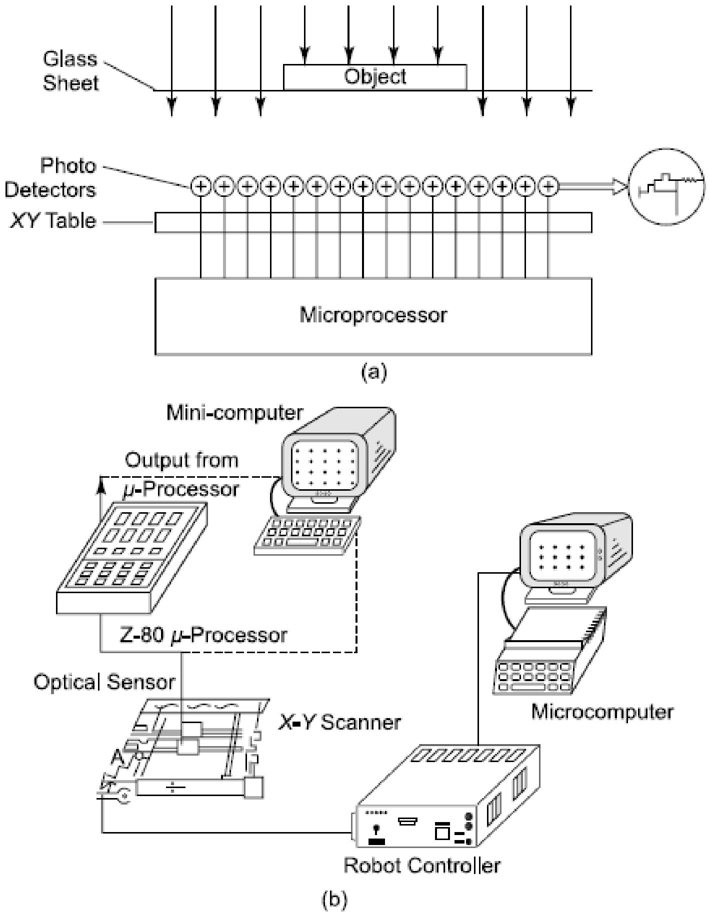
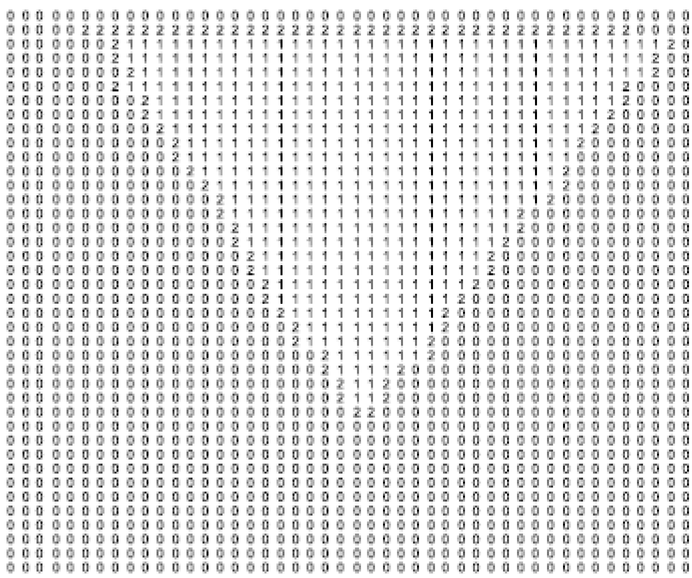


Figure 5.20 indicates the printed output of the triangular object in two dimensions. Once the contour picture is obtained, object features like area, perimeter, centre of area etc. can be found for 2-dimensional pattern recognition.

Figure 5.20 The 2-D pattern of a triangular shaped object



5.6 ROBOT VISION SYSTEMS

5.6. ROBOT VISION SYSTEMS

Robot vision, also termed computer vision or machine vision, is an important sensor technology with potential applications in many industrial operations. Many of the current applications of machine vision are in inspection. However, it is anticipated that this will play an increasingly significant role in the future of robotics. While proximity, touch and force-sensing play a significant role in the improvement of robot performance, vision is recognized as the most powerful of robot sensory capabilities, and sensors, concepts and processing hardware associated with robot vision are considerably more complex than those associated with the sensory approaches discussed earlier. Though computer vision is a multidisciplinary field, it is still in its early stages of development.

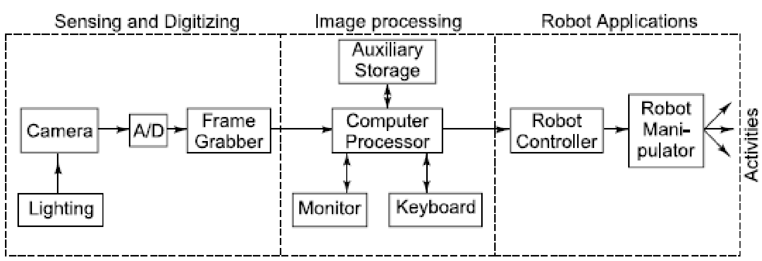
Robot vision may be defined as the process of extracting, characterizing and interpreting information from images of a three-dimensional world. The operation consists of three functions, broadly:

1. Sensing and digitizing—It is the process that yields a visual image of sufficient contrast that is typically digitized and stored in the computer memory.
2. Image processing and analysis—The digitized image is subjected to image processing and analysis for data reduction and interpretation of the image. It may be further subdivided into:

    - preprocessing
    - segmentation
    - description
    - recognition
    - interpretation

    Preprocessing deals with techniques like noise reduction and enhancement details. Segmentation partitions an image into objects of interest. Description computes various features like size, shape, etc. suitable for differentiating one object from another. Recognition identifies the object and finally interpretation assigns meaning to an ensemble of recognized objects in the scene.
3. Application—The current applications of robot vision include inspection, part identification, location and orientation. Research is ongoing in advanced application in complex inspection, guidance and navigation. The relationships between the three functions are illustrated in Fig. 5.21.

Figure 5.21 Machine vision system functions



The various areas of vision processing are grouped according to the sophistication involved in their implementation, such as low, medium and high level visions. The low-level vision refers to processes that are primitive in the sense that they require no intelligence on the part of the vision system. Sensing and processing will be treated as low level vision functions. Medium level vision refers to processes involving extraction, characterization and labelling components in an image resulting from low level vision. Segmentation, description, and recognition of individual objects are treated as medium level vision functions.

Finally, high level vision refers to the processes that attempt to emulate cognition. Interpretation will be treated as high level vision processing.

5.6.1. Low Level Vision

Low level vision includes basically sensing and preprocessing. The concepts and techniques required to implement these vision functions are described in this section. True vision is inherently a 3–D activity, but most of the work in vision is carried out using images of a 3–D scene with depth information being obtained by special imaging techniques such as the structured-lighting approach, or by the use of stereo-imaging.

5.6.2. Sensing and Digitizing

Visual information is converted to electrical signals by visual sensors. When sampled is spatially and quantized in amplitude, these signals yield a digital image.

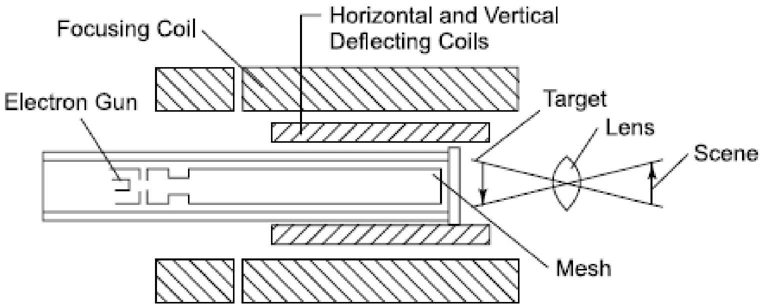
Image sensing requires some type of imaging device (such as a camera) and a digitizer which stores video frames in the computer memory. The initial step involves capturing the image of the scene with a vision camera. The light intensities corresponding to different portions of the scene are continuous analog values and must therefore be sampled and converted into digital form. The second step, digitizing, involves analog to digital (A/D) conversion by an A/D converter which is either a part of a digital video camera or the front end of a frame grabber. The frame grabber, representing the third step, is an image storage and computation device which stores a given pixel array. The frame grabber can vary in capability from one which simply stores an image to a more powerful frame grabber in which thresholding, windowing and calculations for histogram modification can be carried out under computer control. The stored image is subsequently processed and analyzed by the combination of the frame grabber and the vision controller.

5.6.2.1. Vision Cameras

The principal imaging devices used for robot vision are television cameras, consisting either of a tube (vidicon camera) of the solid state cameras (CCD, CID or silicon bipolar sensor cameras), and associated electronics. The principles of operation of the vidicon camera, a commonly used representative of the tube family of TV cameras and the charge-coupled device (CCD), one of the principal exponents of solid state image sensors have been considered. Solid state imaging devices offer a number of advantages over tube cameras such as lighter weight, smaller size, longer life and lower power consumption. However, the resolution of certain tubes is still beyond the capabilities of solid state cameras.

Figure 5.22 shows the scheme of a vidicon camera. An image is formed on the glass faceplate that has its inner surface coated with two layers of materials. The first layer is a transparent signal electrode film deposited on the faceplate of the inner surface. The second is a thin layer of photosensitive material deposited over the conducting film which consists of small areas of high density. Each area produces a decreasing electrical resistance in response to increasing illumination. A charge is created in each small area upon illumination. Thus an electrical charge pattern is generated corresponding to the image formed on the faceplate, which is a function of light intensity over a specified time.

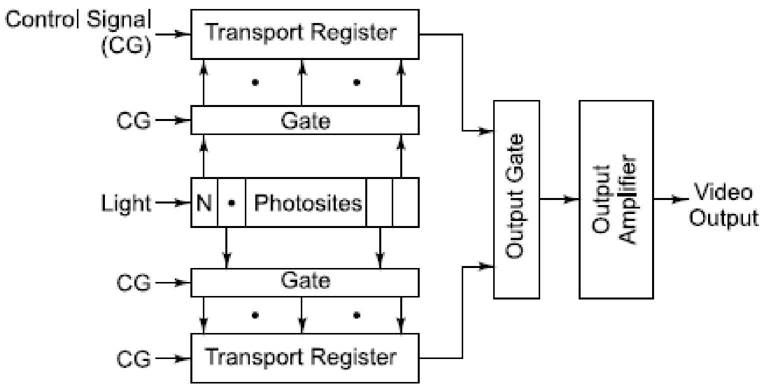
Figure 5.22 Scheme of vidicon camera



Once a light sensitive charge is generated, it is read out by scanning the photosensitive layer by an electron beam to produce a video signal, which is controlled by deflection coils mounted on the tube. The electron beam deposits enough electrons to neutralize the accumulated charge and the flow of electrons causes a current at the video signal electrode. The magnitude of the signal is proportional to the light intensity and the time for which the area is scanned. Raster scanning makes the scan time the same for all areas which is usually done by scanning the faceplate from left to right and top to bottom. To avoid flickering of the image, the frame is divided into two interlaced fields each scanned at twice the frame rate. The first field of each frame scans the odd lines, while the second scans the even lines. The camera output is a continuous voltage signal for each line scanned and is then sampled and quantized before being stored as a series of sampled voltages in the memory of a computer. The analog-to-digital (A/D) conversion results in a two-dimensional array of picture elements or pixels.

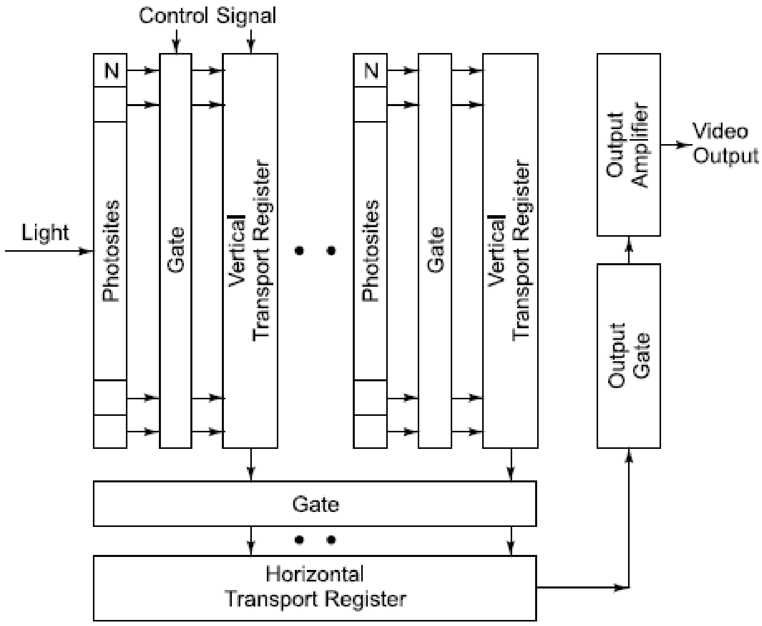
Another approach of obtaining a digitized image is by using a charge-coupled device (CCD) as shown inFig. 5.23 in which an image is projected onto the CCD which detects, stores and reads out the accumulated charge generated by the light pattern of the image. Light detection occurs through the absorption of light on a photoconductive substrate (e.g., silicon). CCD devices can be subdivided into two categories: line scan sensors and area sensors. The basic component of a line scan sensor is a row of silicon imaging elements called photosites. Image photons passing through a transparent polycrystalline silicon gate structure are absorbed in the silicon crystal, thus creating electron-hole pairs. The resulting photo-electrons are collected in the photosites with the amount of charge collected at each photosite being proportional to the light intensity at that location. A typical line scan sensor is composed of a row of photosites, two transfer gates to clock the contents of the photosites into so-called transport registers and an output gate to clock the contents of the transport registers into an amplifier whose output is a voltage signal proportional to the contents of the row of photosites.

Figure 5.23 CCD line scan sensor



Charge coupled area sensors shown in Fig. 5.24 are similar to the line scan sensors excepting that the photosites are arranged in a matrix format and there is a gate transport register combination between columns of photosites. The contents of odd-numbered photosites are sequentially gated into the vertical transport registers and then into the horizontal transport register. The content of this register is fed into an amplifier whose output is a line of video. Repeating this procedure for the even-numbered lines completes the second fields of a TV frame.

Figure 5.24 CCD area scan sensor



Line scan sensors obviously yield only one line of an input image and are ideally suited for applications in which objects are moving past the camera as on conveyor belts. A two-dimensional image is produced by the motion of the object in the direction perpendicular to the camera.

5.6.2.2. Illumination Techniques

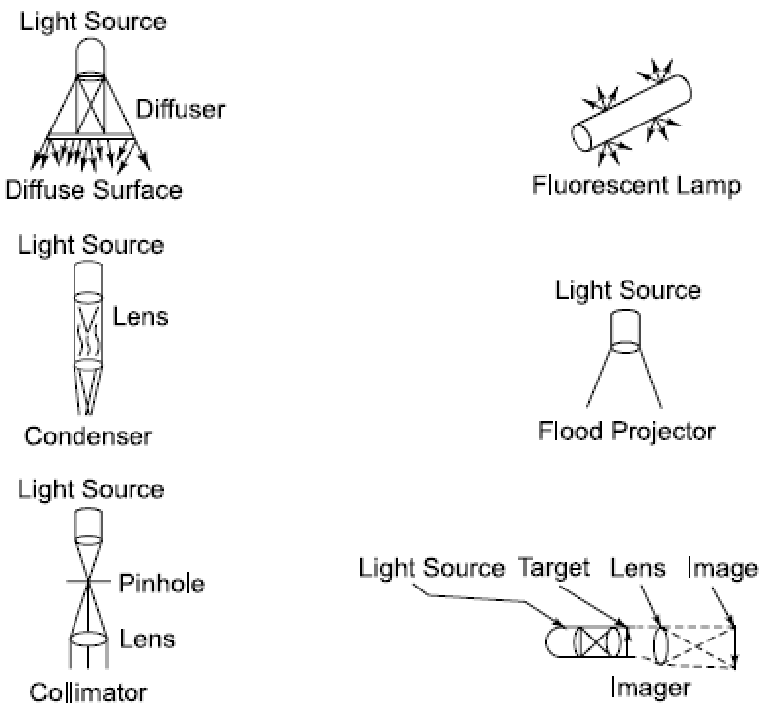
Proper illumination of a scene is an important factor that often affects the complexity of vision algorithms. Arbitrary lighting of the scene can result in low contrast images, specular reflections, shadows and extraneous details. A well-designed illumination system minimizes the complexity of the resulting image, while the information required for object detection and extraction is enhanced.

The basic types of lighting devices in robot vision may be grouped into the following categories:

1. Diffuse surface devices
2. Condenser projectors
3. Flood or spot projectors
4. Collimators
5. Imagers

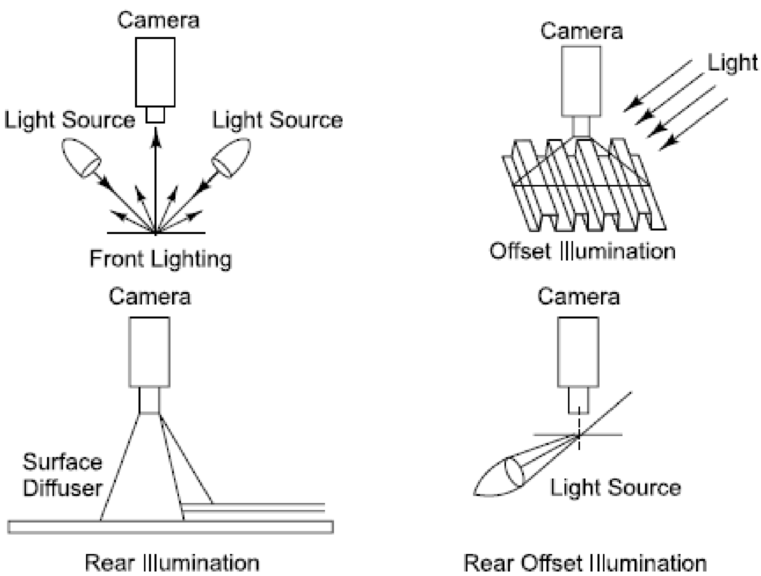
The illuminators are shown in Fig. 5.25(a).

Figure 5.25(a) Basic types of illuminators



There are two basic illumination techniques used in robot vision: front lighting and back lighting. In front lighting, the light source is on the same side of the scene as the camera and reflected light is used to create the image viewed by the camera. In back lighting the light source is directed at the camera and is located behind the objects of interest. Figure 5.25(b) illustrates the basic illumination techniques.

Figure 5.25(b) Basic illumination techniques



5.6.2.3. Analog-to-Digital (A/D) Signal Conversion

For vidicon tube camera, it is necessary to convert the analog signal for each pixel into digital form. Analog-to-digital (A/D) conversion process consists of three phases: sampling, quantization and encoding.

A given analog signal is approximated by the sampled digital outputs after a series of discrete-time analog signals are obtained. This is done by sampling the analog signal periodically at a proper sampling rate. Each sampled voltage level is quantized into a finite number of defined amplitude levels which correspond to the gray scale used in the system. The number of quantization levels is equal to 2^n, where n is the number of digits (bits) of the A/D converter.

The amplitude levels that are quantized must be changed into digital code, the process being termed encoding. This involves representing an amplitude level by a binary digit sequence.

5.6.2.4. Image Storage

The image following A/D converter is stored in the computer memory, typically called a frame buffer, which may be a part of the frame grabber, or, in the computer itself. Various techniques have been developed to acquire and access digital images. Frame grabber is an example of a video data acquisition device. A combination of row and column counters are used in the frame grabber which are synchronized with the electron beam scanning in the camera, so that each position of the screen may be uniquely addressed and the data is 'grabbed' via a signal sent from the computer to the address corresponding to a row-column combination. Such frame grabber techniques have become extremely popular and are used frequently in vision systems.

5.6.3. Preprocessing

Preprocessing deals with techniques like noise reduction and enhancement details. There are several approaches of preprocessing used in robot vision systems. The preprocessing approaches typical of the methods satisfying the requirements of computational speed and low implementation cost are discussed in this section.

5.6.3.1. Noise Reduction or Smoothing

Smoothing operations are used for reducing noise and other spurious effects that are introduced in an image as a result of sampling, quantization, transmission or disturbances in the environment during image acquisition and digitizing.

One straightforward technique for image smoothing is neighbourhood averaging in which a smoothed image is generated whose intensity at every point is obtained by averaging the intensity values of the pixels of the given image contained in a predefined neighbourhood of that point. One of the principal difficulties in this technique is that it blurs edges and other sharp details. This blurring can be reduced by the use of so called median filters in which the intensity of each pixel is replaced by the median of the intensities in a predefined neighbourhood of that pixel, instead of by the average.

5.6.3.2. Enhancement

One of the principal difficulties in many low level vision tasks is to be able to automatically adapt to changes in illumination, which often plays a central role in determining the success of subsequent processing algorithms. Several enhancement techniques are available which address these and other similar problems. Enhancement is a major area of digital image processing and scene analysis.

Histogram equalization or histogram linearization technique is ideally suited for automatic enhancement and is based on a transformation function that is uniquely determined by the histogram of the input image. However this method is limited in the sense that it is not applicable when *a priori* information is available regarding a desired output histogram shape. Hence the concept is generalized by developing an approach capable of generating an image with a specified intensity histogram. Histogram equalization is a special case of this technique, known as histogram specification.

The histogram equalization and specification methods are global in the sense that pixels are modified by a transformation function which is based on the intensity distribution over an entire image. While this global approach is suitable for overall enhancement, it is often necessary to enhance details over small areas, called local enhancement by devising transformation functions that are based on the intensity distribution, or other properties, in the neighbourhood of every pixel in a given image.

5.6.4. Higher Level Vision

Both medium level and high level vision will be discussed briefly in this section. The material is subdivided into four principal areas: segmentation, object description or feature extraction, object recognition and finally the interpretation of visual information.

5.6.4.1. Segmentation

As has been mentioned earlier, segmentation process subdivides a sensed scene into its constituent parts or objects. It is one of the most important elements of automated vision system, since it is at this stage of processing that objects are extracted from a scene for subsequent recognition and analysis. Segmentation algorithms are generally based on one of the two basic principles: similarity and discontinuity. The principal approaches in the first category are based on thresholding and region growing, and that in the second category are based on edge detection. These concepts are applicable to both static and dynamic (or time varying) scenes.

5.6.4.2. Thresholding

In its simplest form, thresholding is a binary conversion technique in which each pixel is converted into a binary value, either black or white. This is accomplished by utilizing a frequency histogram of the image and establishing what intensity (gray level) is to be the border between black and white. This method of using a histogram to determine a threshold is only one of a large number of ways to threshold an image. However, the method is used by many commercially available robot vision systems today.

Thresholding may be viewed as an operation that involves tests against a function T of the form:

$$T = T\,[x, y, p\,(x, y), f\,(x, y)]$$

(5.1)

where $f(x, y)$ is the intensity at the point (x, y), and $p(x, y)$ denotes some local property measured in the neighbourhood of this point. A thresholded image, $g(x, y)$ is created by defining

$$g(x, y) = \begin{cases} 1 & \text{if} \quad f(x, y) > T \\ 0 & \text{if} \quad f(x, y) \le T \end{cases}$$

(5.2)

so that pixels labelled 1 correspond to objects, while pixels labelled 0 correspond to the background assuming that the intensity of objects is *greater* than the intensity of the background. The opposite condition is treated with reversing the sense of inequalities. When T depends only on $f(x, y)$, the threshold is global. On the other hand, if T depends on both $f(x, y)$ and $p(x, y)$, the threshold is called local. If, in addition, T depends on the spatial coordinates x and y, it is called a dynamic threshold.

Global thresholds have applications in situations where there is a clear distinction between objects and the background and where illumination is relatively uniform. The back lighting and structured lighting techniques usually yield images that can be segmented by global thresholds; on the other hand, arbitrary illumination often yields images that require some type of local analysis to compensate for effects such as non-uniformities in illumination, shadows and reflections.

5.6.4.3. Region Growing

Once thresholding is established, the next step is to identify particular areas associated with objects within the image. Such regions usually possess uniform pixel properties computed over the area. As its name implies, region growing is a procedure that groups pixels or subregions into larger regions based on attribute similarities. The simplest approach is pixel aggregation, where one can start with a set of 'seed' points, and grow regions from these by appending to each seed point those neighbouring pixels that have similar properties, e.g., intensity, texture or colour. Two immediate problems are the selection of initial seeds that properly represent regions of interest and the selection of suitable properties or similarity criteria for including points in the various regions during region growing process. While the selection of a set of one or more starting points can often be based on the nature of the problem, the selection of similarity criteria is dependent not only on the problem under consideration but also on the type of image data available. Typically, region growing analysis must be carried out using a set of descriptors (to be discussed in the next section) based on intensity and spatial properties of a single image source. It is also important to note that descriptors alone can yield misleading results if connectivity or a adjacency information is not used in region growing process. Another important problem is the formulation of a stopping rule. Basically, we stop growing a region when no more pixels satisfy the criteria for inclusion in that region.

Region oriented segmentation process must meet certain conditions: (i) segmentation must be complete i.e., every pixel must be in a region, (ii) points in a region must be connected and (iii) the regions must be disjoint.

The procedure discussed above grows regions starting from a given set of seed points. Another alternative is to initially subdivide an image into a set of arbitrary, disjoint regions and then merge and/or split the regions satisfying the conditions just stated.

5.6.4.4. Edge Detection

Edge detection considers the intensity change that occurs in the pixels at the boundary or edges of a part. Once a region of similar attributes has been found, the boundary can be determined by a simple edge following procedure. For a binary image, the procedure is to scan the image from top left until a pixel within the region is encountered. For a pixel within the region, turn left and step; otherwise, turn right and step until the path has returned to the starting image, when the boundary is traversed. The contour-following procedure can be extended to gray level images.

The basic idea underlying most edge detection techniques is the computation of a local derivative operator. The first derivative at any point in an image can be obtained by using the magnitude of the gradient at that point, while the second derivative is given by the Laplacian operator.

Gradient Operator The gradient of an image $f(x, y)$ of location (x, y), denoted by $G[f(x, y)]$, is given by

$$G[f(x, y)] = [G_x^2 + G_y^2]^{1/2}$$

(5.3)

where,

$$G_x = \delta f/\delta x$$

(5.4)

$$G_y = \delta f/\delta y$$

(5.5)

The computation of the gradient is based on obtaining the first order derivatives $\delta f/\delta x$ and $\delta f/\delta y$. One approach is to use first order differences between adjacent pixels, i.e.

$$G_x = \frac{\delta f}{\delta x} = f(x, y) - f(x - 1, y)$$

(5.6)

$$G_x = \frac{\delta f}{\delta x} = f(x, y) - f(x -1, y)$$

(5.7)

A slightly more complicated definition involving pixels in a 3 × 3 neighbourhood centered at (x, y) is given by,

$$\begin{aligned} G_x = \frac{\delta f}{\delta x} &= [f(x + 1, y - 1) + 2f(x + 1, y) + f(x + 1, y + 1)] \\ &\quad - [f(x -1, y - 1) + 2f(x - 1, y) + f(x - 1, y + 1)] \end{aligned}$$

(5.8)

$$\begin{aligned} G_y = \frac{\delta f}{\delta x} &= [f(x - 1, y + 1) + 2f(x, y + 1) + f(x + 1, y + 1)] \\ &\quad - [f(x - 1, y - 1) + 2f(x, y - 1) + f(x + 1, y - 1)] \end{aligned}$$

(5.9)

It is noted that the pixels closest to (x, y) are weighted by 2 in these definitions. These definitions have the advantage over Eqs (5.6) and (5.7) of increased averaging, thus tending to make the gradient less sensitive to noise. G_x and G_y as given by the last pair of equations can be computed by using the masks shown in Figs 5.26(a) and 5.26(b), commonly referred to as Sobel operators.

Figure 5.26 Masks used for computing G_x and G_y

–1	–2	–1
0	0	0
1	2	1

(a)

–1	0	1
–2	0	0
–1	0	1

(b)

Based on gradient computations, one can generate an output image in numerous ways. The simplest approach is to let the value of output image at coordinates (x, y) be equal to the gradient of the input image at that point. Another approach is to create a binary image having values 0 or 1 depending on whether the gradient is less than or greater than a threshold value.

Laplacian Operator The Laplacian is a second order derivative defined as

$$L[f(x, y)] = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

(5.10)

The Laplacian, for digital images, is defined as

$$L[f(x, y)] = [f(x+1, y) + y(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x-y)$$

(5.11)

The above equation can be implemented based on the mask as shown in Fig. 5.27.

Figure 5.27 Masks to compute Laplacian

0	1	0
1	– 4	1
0	1	0

Being a second-order derivative operator, the Laplacian is typically sensitive to noise and is itself seldom used for edge detection. However, it is usually delegated the secondary role of serving as a detector for establishing whether a given pixel is on the dark or light side of an edge.

Edge detection algorithms are typically followed by linking and other boundary detection procedures designed to assemble edge pixels into a meaningful set of object boundaries.

5.6.4.5. Object Description

The description means extracting features from an object for the purpose of recognition. In vision applications, it is necessary to distinguish one object from another, which is usually accomplished by means of features that uniquely characterize the object. Some object features that can be used in robot vision include area, centre of gravity, diameter, perimeter etc. A feature, in this context, is a single parameter that permits ease of comparison and identification. The techniques available for extracting feature values for two-dimensional cases can be roughly categorized as those that deal with boundary features and those that deal with area features. The various features can be used to identify the object or part, and determine the part location and/or orientation.

Chain Codes These are used to represent a boundary as a set of straight line segments of specified length and direction, typically established on a rectangular grid using 4- or 8-connectivity, as shown in Fig. 5.28 where directions are given by the code chosen. Figure 5.29 shows a result using the 4-connectivity direction code, where the coding was started at the dot and proceeded in a clockwise direction. It is important to note that the chain code of a given boundary depends upon the starting point. However it is possible to normalize the code by a straightforward procedure.

Figure 5.28 *Directional chain codes (a) 4-directional chain (b) 8-directional chain*

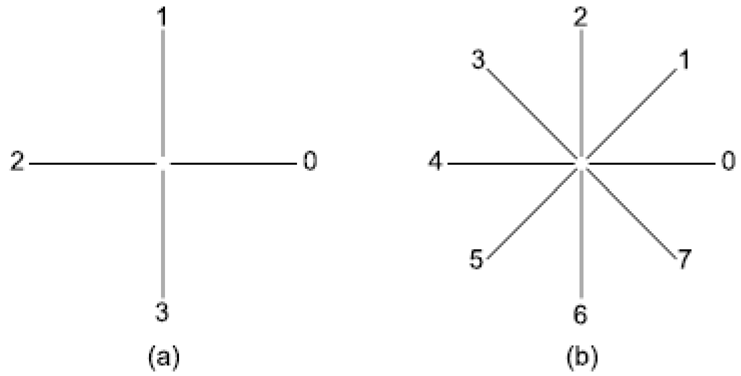
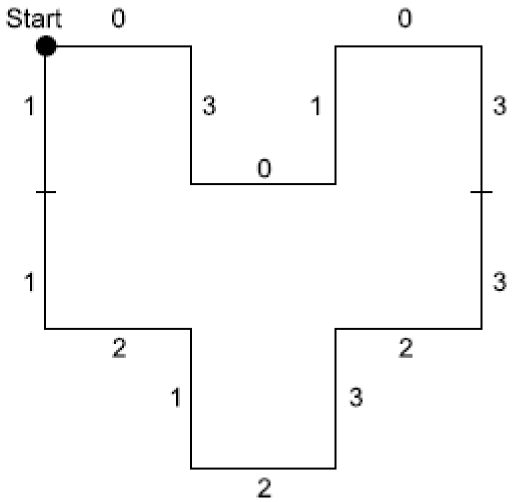


Figure 5.29 *Example of the chain code*



The major axis of a boundary is the straight line segment joining the two points farthest from each other. The minor axis is perpendicular to the major axis and of such a length that a box can be formed to just enclose the boundary. The ratio of the major to minor axis is called the eccentricity of the boundary, and the rectangle just described is called the basic rectangle. The major and minor axes are useful for establishing the orientation of an object. The eccentricity is also an important description of its shape.

The area of a region is defined as the number of pixels contained within its boundary. The perimeter of a region is the length of its boundary. The compactness of a region is defined as perimeter^2/area, which is a dimensionless quantity and minimum for a disc-shaped region.

A connected region is one in which all pairs of points can be connected by a curve lying entirely in the region. For a set of connected regions, some of which may have holes, it is useful to consider the Euler number, which is simply defined as the number of connected regions minus the number of holes.

5.6.4.6. Object Recognition

The next step in image data processing is to identify or recognize the object that the image represents, which is accomplished using the extracted feature information. The function of recognition algorithms is to identify each segmented object in a scene and to assign a label to that object, which must be powerful enough to uniquely identify the object. Object recognition techniques may be classified into two major categories:

- Template-matching techniques
- Structural techniques

In template-matching, the object is matched with a stored pattern feature-set defined as a model template that is obtained during the training procedure in which the vision system is programmed for known prototype objects. Training of the vision system should be carried out under conditions as close to operating conditions as possible. The system stores these model objects in the form of extracted feature values which can be subsequently compared against the corresponding feature values from images of unknown objects. These techniques are applicable if there is not a requirement for a large number of model templates. This procedure is based on the use of a sufficient number of features to minimize errors in the classification process.

In structural technique, relationships between features of an object are considered. Structural technique differs from decision-theoretic technique in that the latter deals with pattern on a quantitative basis, and mostly ignores interrelationships among object primitives. Structural methods, on the other hand, attempt to achieve object discrimination by capitalizing on these relationships; central to the structural recognition approach is the decomposition of an object into pattern primitives.

5.6.4.7. Interpretation

Interpretation process endorses a vision system with a higher level of cognition about its environment than that offered by any of the concepts previously discussed and encompasses all these methods as an integral part of understanding a visual scene. This is one of the most active research topics in robot vision. The power of a robot vision system is determined by its ability to extract meaningful information from a scene under a broad range of viewing conditions. Processing scenes requires the capability to obtain description and procedures for establishing relationships between these descriptors, even when they are incomplete.

5.6.5. Applications of Robot Vision System

Many of the current applications of machine vision are inspection tasks that do not involve the use of an industrial robot.

The use of machine vision in robotic applications falls into three broad categories as follows:

1. Inspection
2. Identification
3. Navigation

Inspection process is carried out by the vision system and the robot is used in a secondary role to support the application. The objectives of vision inspection include checking for surface defects, verification of the presence of components in assembly, measuring for dimensional accuracy, discovery of flaws in labelling during final inspection and checking for the presence of holes and other features in a part.

Identification is concerned with applications in which the purpose is to recognize and classify an object rather than to inspect it. Inspection implies that the part must be either accepted or rejected. In identification process, the part itself, or its position or orientation, is determined, followed by a subsequent decision and action taken by the robot. Identification applications include part sorting, palletizing and depalletizing and picking parts that are randomly oriented from a conveyor or a bin.

In navigational control, the purpose is to direct the actions of the robot and other devices in the robot cell based on visual input. An example is to control the trajectory of the robot's end-effector toward an object in the workspace. Industrial applications include part positioning, retrieving and reorienting parts which are moving along a conveyor, assembly, bin picking, seam tracking in continuous arc welding, and automatic robot path planning and collision avoidance using visual data. The visual data are just an important input in this type of task and a great deal of intelligence is required in the controller to use the data for navigation and collision avoidance.

Robot vision, coupled with the contact sensors discussed previously in this chapter, can immediately enhance a robot's applicability in manufacturing.

5.6.5.1. Visual Inspection of Electrical Power Plug

An electrical three pin power plug is taken as an example problem for inspection using robot vision system.

The robot vision system consists of an Adept vision system having the following components:

1. Solid state video camera with area vision capability
2. Vision processor and associated memories
3. Camera multiplexer for up to 8 cameras
4. TV monitor

Typical area vision specifications include,

1. Camera RS-170 format
2. 510 H × 492 V pixels
3. Accuracy 0.5% field of view
4. Ring type fluorescent light
5. Motorola 68000, 10 MHz multibus-based vision processor
6. Recognition speed up to 6 parts per second

The total inspection work is divided into two parts:

- recognition of the part
- finding the detailed geometrical features

The features are then checked with regard to their deviations so as to either accept or reject. So long as the part geometry lies within the limits of tolerance, the part is accepted, otherwise it is rejected.

5.6.5.2. Part Recognition

Template matching technique can be used for part recognition. In template matching technique, the prototype is trained and its features are used to recognize the object placed under the camera. During training, the geometrical dimensions of the cover and base of power plus are taught to the system. The flow chart is shown in Fig. 5.30 for the recognition of the object. The workpiece statistics (Figs 5.31, 5.32) include:

1. the topological positions of comers
2. the lengths and directions of lines or arcs
3. the radii and angular measurement of arcs

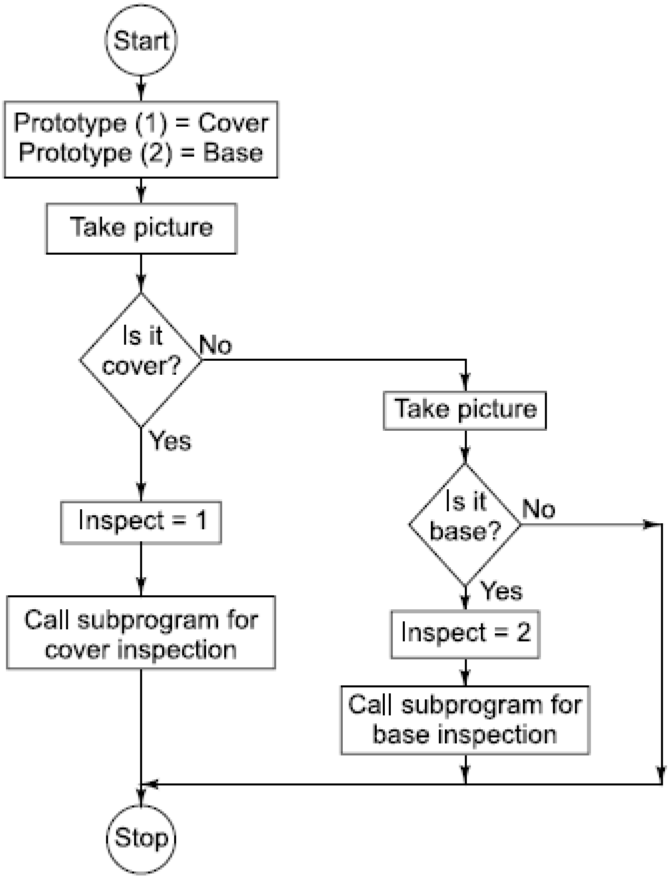


Figure 5.30 *Flow chart for recognition of electrical power plug components*

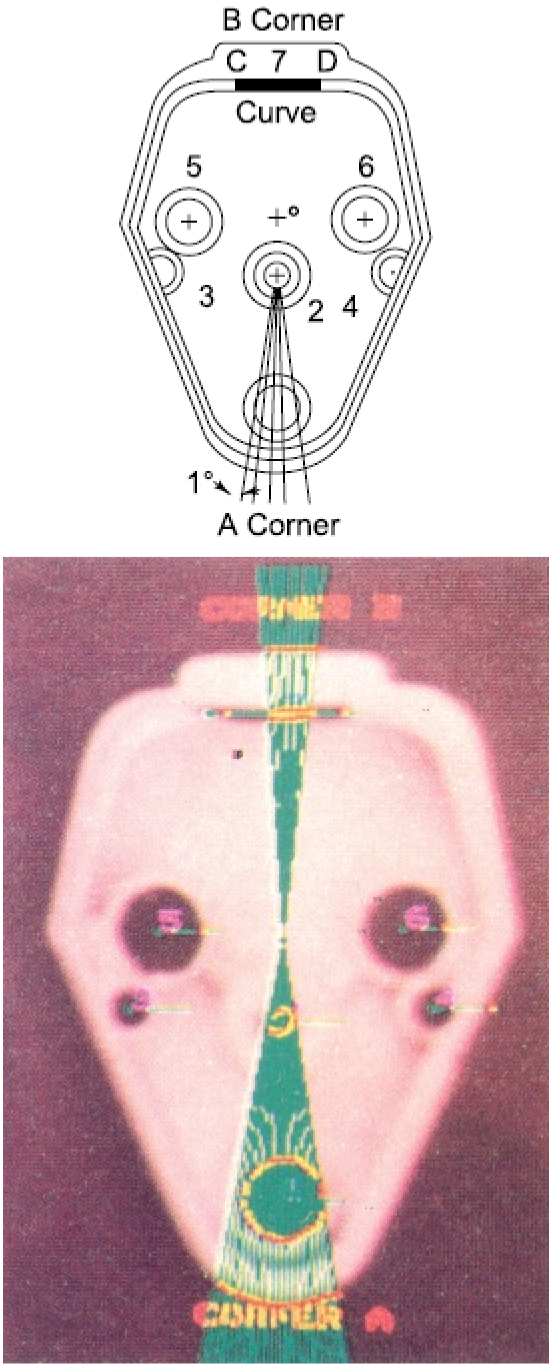
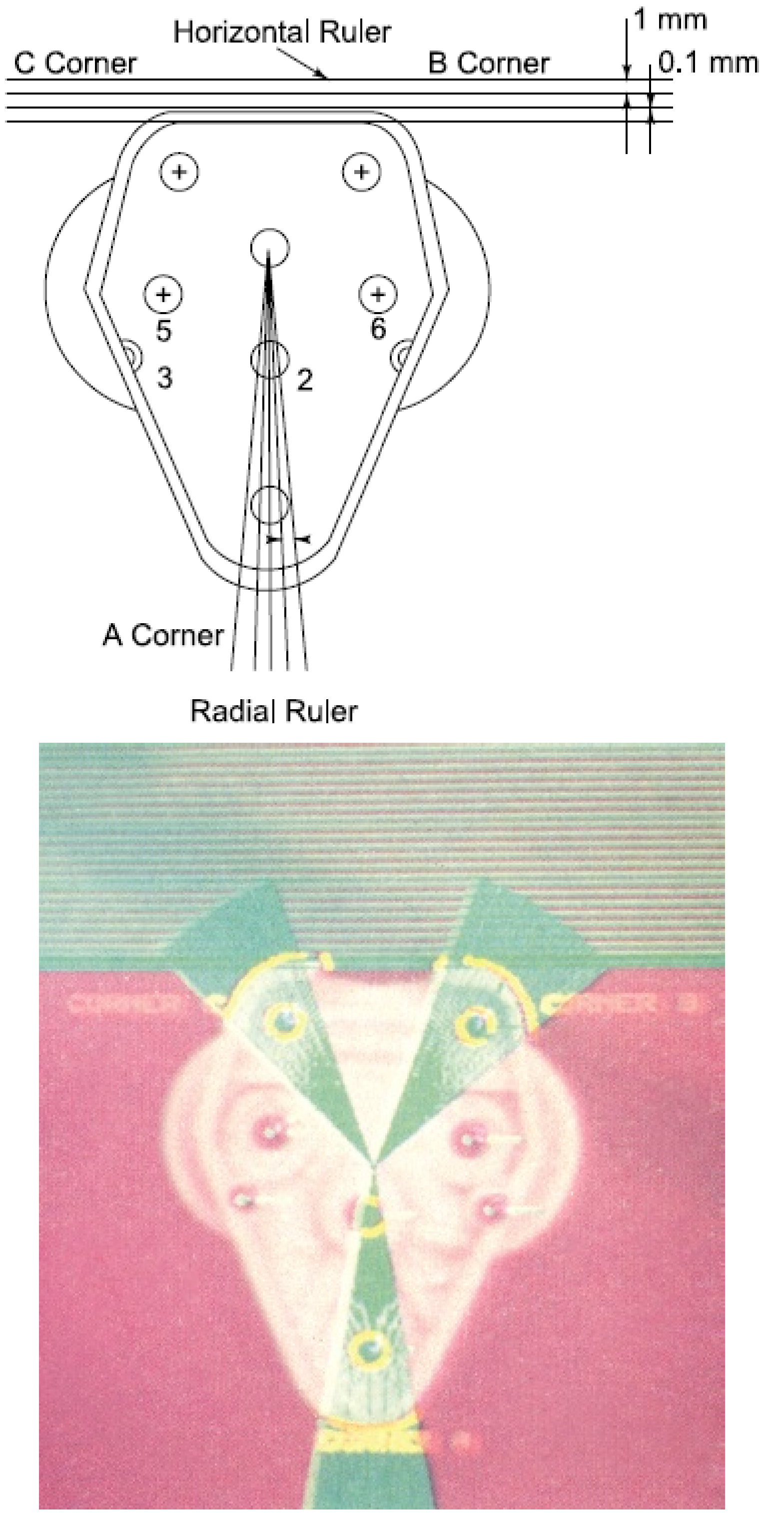


Figure 5.31 *Cover of power plug*

Figure 5.32 Base of power plug



5.6.5.3. Part Inspection

During part inspection, an algorithm, 'ruler' is used to find the distance between the starting point and transitions. A ruler is a general purpose image processing operator which searches for all edges along a directed line in an image. Ruler can be defined by simply specifying its starting and ending points (or start, length and angle) within the camera's frame of reference. Therefore, the ruler can have any length and orientation. If there are any transitions in the ruler, the distances between starting point and transitions are returned to the user-written program as an array of distances.

Ruler syntax in Adept Vision System is as follows:

VRULERI	(CAM, TYPE, DMODE, MAXCUT) ARR [I] MAGS [12]	
	= SHAPE, CX, CY, LEN, ANG.	
CAM	virtual camera to be selected	
TYPE	0	Standard (Default)
	1	Gray-level
	2	Fine-edge
DMODE	–1	No draw (default)
	0	Erase
	1	Draw solid
	2	Complement
	3	Draw dotted
MAXCUT	–1	As many as possible
ARR [ ]	Array with which the transitions detected by the ruler are placed	
I	First array element to be defined	
MAGS [ ]	Only for type 2 rulers	
I2	First array element to be defined	
SHAPE	1 (Straight line)	
Cx, Cy	Coordinates of the starting point of the ruler, mm	
LEN	Length of the ruler, mm	
ANG	Angle of the ruler, degrees	

VFEATURE command is useful in inspection to return specified information about the object most recently located. Syntax is,

VFEATURE (index)

Some of the index values are:

1 Found True/False

.

.

.

9 Verify percentage

10 Region area in pixels

17 Number of holes in the region

.

.

.

41 Region perimeter

The features that have been selected for inspection of cover shown in Fig. 5.31 are illustrated below:

- area of region (180 pixels)

- perimeter length (142)
- radius of holes (1 to 6): 3.14, 0.73, 1.35, 1.35, 3.30, 3.30
- distance between centre of hole 2 and centroid (6.48)
- distance between centre of hole 2 and other holes (1 to 7: 13.18, 0, 12, 12.23, 12.23, 24.25)
- distance between the point A comer and centroid (27.70)
- distance between B corner and centroid (23.14)
- curve diameter, CD, (9.78)

All dimensions are in millimeters except area of region. Area of region is in terms of number of pixels.

In order to find the distance between 'A comer' and centroid, 0, radial rulers are drawn at 1° interval. The rulers that have five transitions have been considered. The average of such five rulers covering the comer A has been accepted as a representative distance between comer A and the centroid.

In a similar way, the base of the power plug can be inspected and the parameters with reference to Fig. 5.32 have been shown below:

- area of region (170 pixels)
- perimeter length (141)
- radius of holes (1 to 8 : 1.48, 1.48, 0.89, 0.89, 1.4, 1.4, 1.4, 1.4)
- distance between centre of hole 2 and other holes: (13, 0, 12, 12, 12, 12, 20, 20)
- distance between centroid and A comer, B comer, C comer respectively (25.5, 22.5, 22.5)
- curve diameter (10)

In order to find the radius of curvature for holding cord sleeve (Fig. 5.32) horizontal rulers at a distance of 1 mm have been drawn and two transition points have been noted for the diameter of the semi-circular curved portion. The horizontal ruler passing through the mid portion of the semi circular curve is taken as the representative ruler and the distance between the two succeeding transition points are measured.

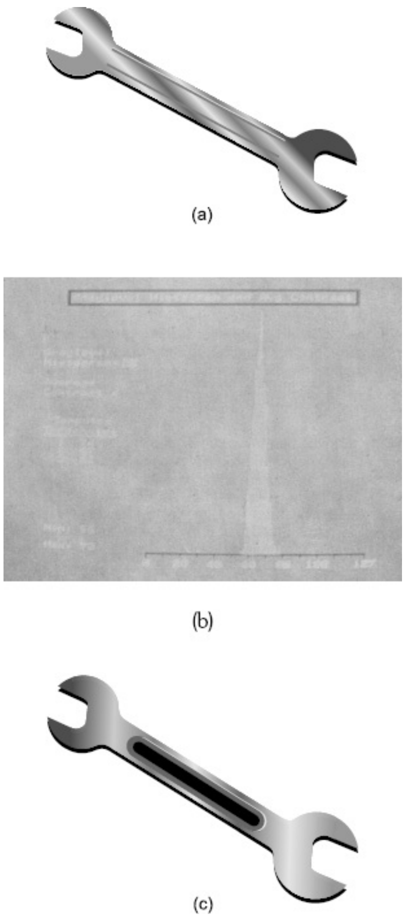
During inspection task, the part is recognized and the recognition program calls a subprogram to get the features of the recognized part. The program checks all the dimensions. If they are within specified tolerance values, the output is 'PART IS ACCEPTED' or else 'PART IS REJECTED'.

If the rulers are drawn and shown on the vision monitor, it takes more time to carry out an inspection task than if the rulers are not shown on the video screen. The time taken for inspection of cover and base are as follows:

	Part	Rulers shown (sec)	Rulers not shown (sec)
Time taken for inspection	COVER	2.0	1.952
	BASE	2.208	2.176

Before carrying on the inspection task, the part to be inspected requires recognition. In order to recognize the part, either gray or binary image is to be obtained. In order to get a binary picture, a thresholding level has to be fixed up. Figure 5.33 illustrates the steps in getting the binary image by thresholding. At first, gray image is obtained; the corresponding gray level histogram and average contrast is drawn and finally the thresholding level is set to obtain the binary image for subsequent image processing.

Figure 5.33 Illustrating gray image, thresholding and binary image (a) Gray image (b) Thresholding after gray level histogram (c) Binary image

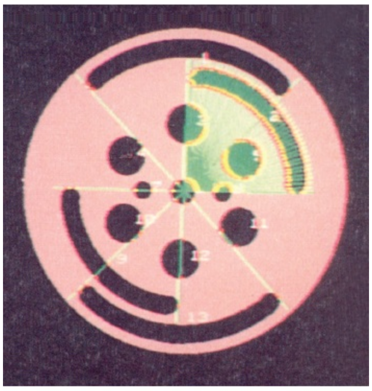


5.6.5.4. Recognition and Inspection of Encoder Disc

Figure 5.34 shows an electro-mechanical component known as shaft encoder disc used for positional feedback in a robot or an NC machine. It has two inner slots and two outer slots.

Figure 5.34 *Photographic views pf encoder-disc CNC System Turrent and Tooling System*

(a)



(b)

For proper operation of the encoder disc, the angular and radial positions of the slots and also the gap of the slots should be within tolerable limits. The slots can be easily inspected by the robot vision system using ruler lines. The 'VRULERI' command can draw a ruler line from one point to another point, and the transition points where the intensity changes are noted. In the present case, ruler lines are drawn from the centre of the disc anticlockwise at close angular intervals and the disc is placed in such a way that the first ruler line drawn at an angle (zero) does not pass through a slot. Starting from the zero datum, the lines are drawn and the first ruler line which passes through a slot gives the angle for the starting edge. All successive ruler lines are checked to see whether they pass through the slot, i.e. whether they have transition points within a specific length. The last line for which this condition is met, gives the angle for the other edge of the slot. The angles for the edges of two inner and two outer slots are thus determined. Such angles are given from the reference line corresponding to the starting edge for the first slot encountered in this process.

To inspect the gap all along the slot, ruler lines at certain angular intervals of 1° between the two edges of a slot are drawn and their transition points are noted. For all such lines, the maximum length for the inner transition point and the minimum length for the outer transition point are computed and shown on the display. The gap at any point on the slot cannot be less than the difference between the above two values. The minimum gap for a slot is thus determined.

In addition to the above inspection results, the area, perimeter, minimum and maximum radii of the disc and the number of holes are also checked. From the pictures of the disc taken, the minimum, maximum and average values of the above parameters are calculated, and shown in Table 5.2. These values are also used for recognizing the disc.

Table 5.2 Simple Inspection Result for Encoder Disc

1. Recognition Time = 0.592 Seconds					
2. Parameters for the disc (as a whole)					
Area (sq. mm)	Perimeter (mm)	Minimum radius (mm)	Maximum radius (mm)	No. of holes	Total hole area (sq. mm)
2099.126	162.4461	25.70215	25.97312	13	597.413

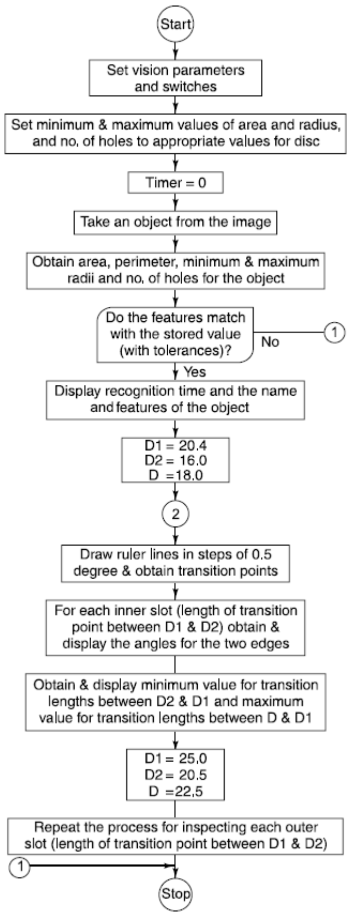
The display of encoder disc on the vision monitor is shown in Fig. 5.34.

Table 5.3 shows typical inspection results for the disc with the help of a vision software developed using V^+ language (VAL-II). The flow chart for the vision programme is illustrated in Fig. 5.35.

Table 5.3 Parameter for the Slots of Disc Obtained Using V^+ Language

Slot position	Angle for starting edge (degree)	Angle for other edge (degree)	Maximum distance for inner edge (mm)	Minimum distance for outer edge (mm)
Inner Slot 1	180.0	90.5	16.66	19.90
Inner Slot 2	181.0	271.5	16.64	19.91
Outer Slot 1	45.5	136.0	20.99	24.41
Outer Slot 2	226.5	316.0	20.98	24.45

Figure 5.35 Flowchart for the vision program for recognition and Inspection of encoder disc



5.6.5.5. Inspection of Solder Joints of Printed Circuit Boards

To ensure the performance of high-quality printed circuit boards, inspections are necessary for the solder joints. Two types of inspections are often done on the board. One is an electrical inspection. In terms of solder quality, this type of inspection can locate solder bridges between solder joints. An electrical test will not detect solder joints with insufficient or no solder even if the lead is in temporary or partial contact with the land and loose electrical contact exists. Missing solders cannot be detected by the tester in many cases.

Another type of inspection is visual inspection, that can locate solder defects such as solder insufficiencies, excessive solder, missing or no solder, and bent-over leads that could form a bridge after packaging and shipping of the boards. Visual inspection of solder joints can be an inefficient operation when performed entirely through human eyes.

The inspection of the solder joints can be performed by a vision system, operating on a two-dimensional black and white (binary) image of a portion of the board. The camera field of view on the board, is restricted to an area of approximately 30 mm × 30 mm. A sample PCB is taken. The inspection scene consists of a number of joints, varying from 30 to 35, depending on the location of the board. The solder joints are connected to traces which are connected to other solder joints in the scene or which terminate at the boundary of the scene. In detection of solder joints, it is necessary to detect the shape of the joint correctly without being influenced by the gloss and blur of the solder surface. The vision processor must be able to distinguish the joints from the traces and background board. This is accomplished through special scene illumination and filtering techniques. A ring tube fluorescent light is used around the board for uniform lighting. The solder trace is a dark-green colour. When an orange-red filter is placed in front of the camera lens, everything green in the scene is made to look dark. In this way traces are suppressed, and the vision processor is presenting with a scene in which traces have disappeared leaving only the solder joints.

The photograph as shown in Fig. 5.36 shows how the solder joints would appear to the vision processor without the special illumination and filtering where the traces are also visible. Figure 5.37 shows how the solder joints appear with the special illumination and filtering, as described above. After filtering, the traces are shown to be suppressed, leaving only the solder joints.

Figure 5.36 Photograph of PCB solder joints with traces

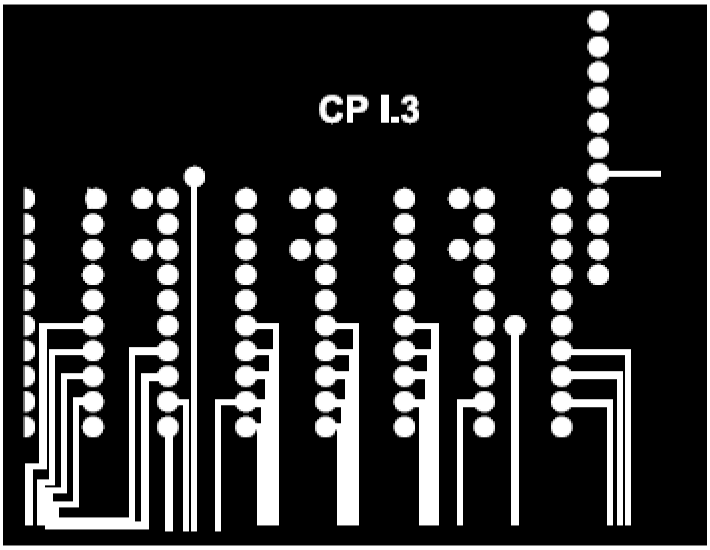
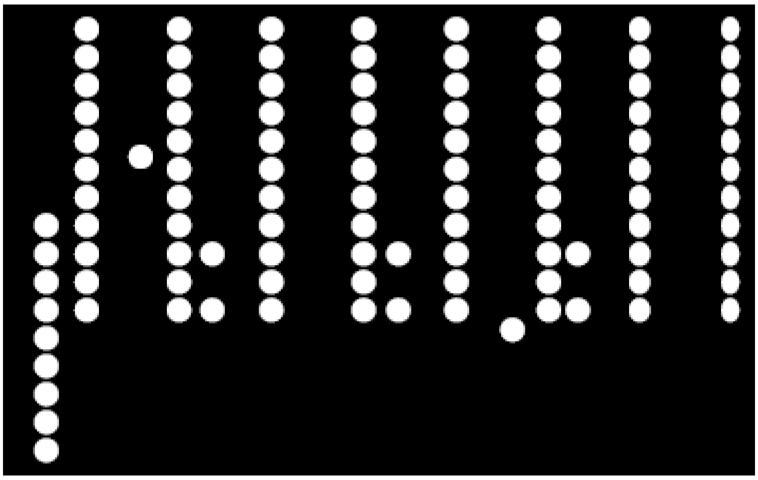


Figure 5.37 Photograph of PCB solder joints suppressing the traces



The figures also indicate normal solder, oversolder and undersolder in some of the joints. The inspection performed by the vision system is much faster when the solder joints are isolated in the scene as above than if the vision processor had to separate the joints from the traces before the actual inspection. With isolated solder joints, the vision system is programmed to begin inspection of first joint, moving to next joint in a row. After completion of inspection of all solder joints in a row, it starts inspecting the joints in next row. Each joint is treated as a separate region (object), the features of which are measured and compared to its predetermined values.

The following object features are used in the solder inspection:

1. Presence of hole (gap) in a solder joint.
2. Size of a solder joint measured by its area and size of a bounding box that just fits around the solder joint.
3. Shape of solder joint, measured by a non-dimensional roundness parameter, expressed by a quantity equal to the area of

an object divided by its perimeter squared, and normalized such that for a circle the parameter equals unity (1). Anything more irregularly shaped than a circle would have a value less than 1. In addition, the minimum and maximum radii of the solder joints are also obtained as a measure of the shape.

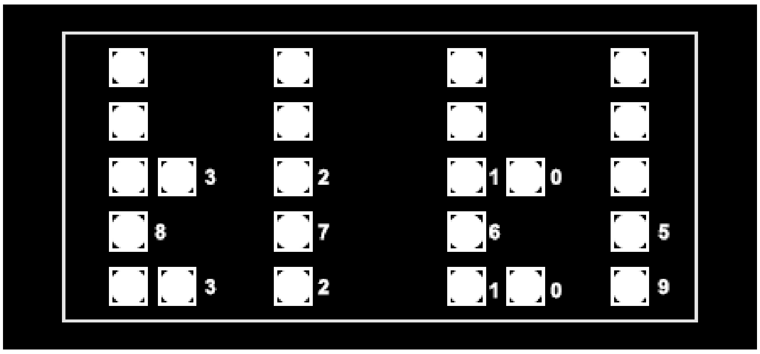
The above features are measured for a number of solder joints. A solder joint having roundness parameter greater than 0.9 is taken as 'good' shape, and their minimum, maximum and average values are determined. These values are shown in Table 5.4. These values are considered as standard accepted values for solder joints for the specimen printed circuit board (PCB).

Table 5.4 Feature for PCB Solder Joints Having Good Shape (Roundness Parameter > 0.9)

Total no. of joints taken = 30			
Parameter	Minimum	Maximum	Average
Area (sq. mm)	1.8349	2.2872	2.0450
Perimeter (mm)	4.6579	5.7786	5.1759
Minimum Radius (mm)	0.6638	0.7932	0.7433
Maximum Radius (mm)	0.7864	0.9848	0.8634
x-Bound (mm)	1.4558	1.7660	1.6127
y-Bound (mm)	1.4716	1.7661	1.6204

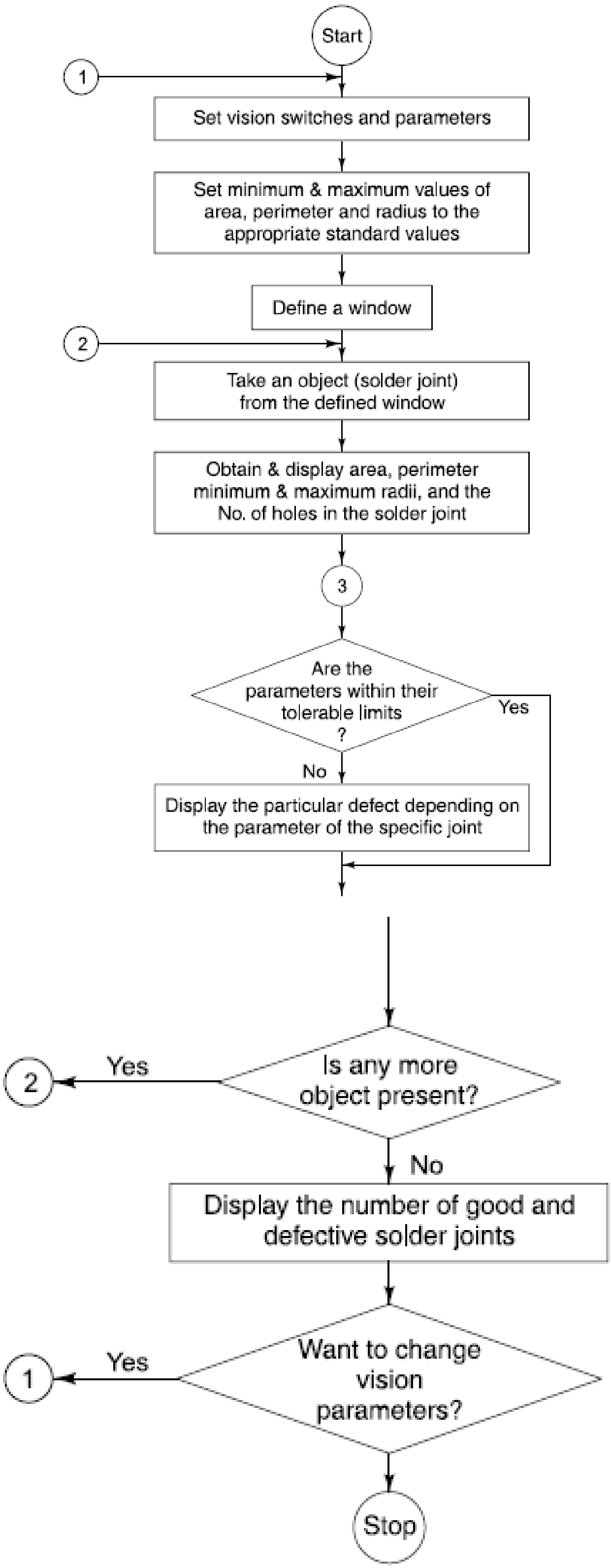
The vision system is programmed in V+ language to move from joint to joint in a scene, and measure these features for each joint. The presence of a hole in a solder joint is always an error condition. If a solder blob has too large a bounding box, there may be a bridge between the two joints. A box, too small, indicates missing solder or a spot of spurious solder. An irregularly shaped solder joint indicates missing solder or a solder bridge. Typical illustrations of a missing solder joint and a defective solder are given in photographic views of Fig. 5.38.

Figure 5.38 Detective solder joints



The flow chart of vision programme developed for the inspection of solder joints to detect the defective solder joints is illustrated in Fig. 5.39.

Figure 5.39 Flowchart for the vision program for inspection of PCBsolder joints



When a large number of PCBs are to be inspected in an automated fashion, a robot interfaced with a vision processor can be employed to provide flexible handling and positioning the board in front of a stationary camera. The robot can also position the circuit board as directed by the vision processor. When the board is properly located under the field of view of the camera, the vision processor takes a picture of the board area and inspects with the help of a suitable vision software. Each solder joint is inspected within the area in the scene individually. The inspection results for the board are displayed on the display unit. An operator can locate quickly those solder joints that need to be touched up. Adept SCARA robot with SXGS vision processor provides a good facility for the inspection of solder joints of printed circuit boards.

5.7. DESIGN AND CONTROL OF SENSOR INTEGRATED DEXTEROUS ROBOT HAND

5.7.1. Sensors and Robotic Prehension

Design of four-fingered tendon actuated gripper and a direct-linked robot hand have been described inSec. 4.10. The present illustration depicts the development of sensory systems and gripper control systems. Contact and noncontact sensors have been employed to grip the objects securely by the robot grippers.

The contact sensors used are

- Force sensors
- Torque sensors
- Touch sensors
- Position sensors
- Slip sensors

and the non-contact sensors include:

- Electro-optical imaging sensors (CCD cameras)
- Proximity sensors
- Range imaging sensors

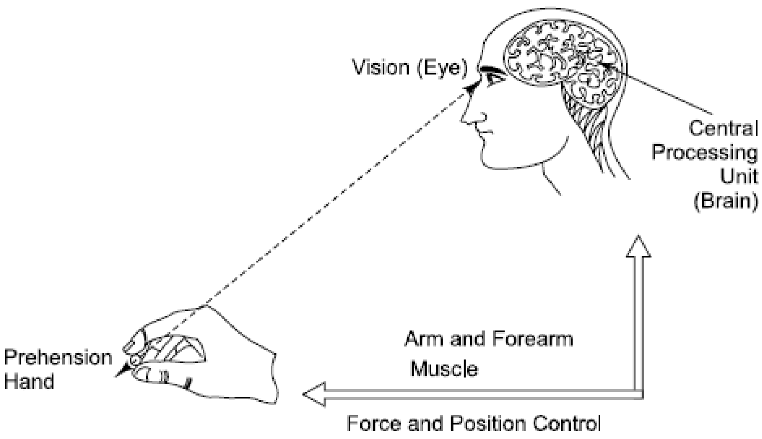
The act of sensing, in general, is performed in two steps:

a. Transducing—converting energy or physical condition to be sensed into a signal (usually electrical).
b. Signal conditioning—improving the signal-to-noise ratio, averaging, filtering and data conversion.

5.7.1.1. Robotic Prehension

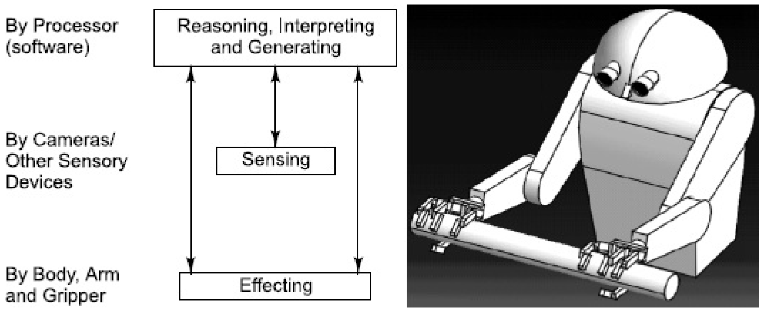
Robotic hands having multiple articulated fingers provide versatility in grasping different objects if a large number of sensory feedbacks and intelligence are provided. The process of prehension controlled by feedback loops with position and force of the fingers can be shown in Fig. 5.40.

Figure 5.40 The Eye–memory integration for human prehension



The robotic prehension is shown in Fig. 5.41.

Figure 5.41 Simulation of robotic prehension

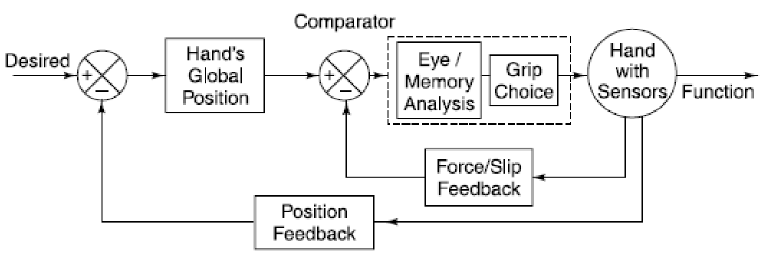


There are three steps on prehension:

1. Sensing
2. Reasoning, Interpreting and Generating
3. Effecting

The scheme of eye–memory analysis with hand's position in feedback loops is shown in Fig. 5.42.

Figure 5.42 Feedback loop in prehension



5.7.1.2. Sensors for Robotic Gripper

For the robot griper, a number of sensors have been employed:

- Contact sensor. Contact sensors such as limit switches have been used to indicate that the contact has been made between the object and the gripper.
- Position and displacement sensors. In order to get information about the position of each joint, potentiometers and encoders have been used as sensors.
- Force sensors. Electrical strain gauges type force sensors of (i) octagonal ring type and (ii) the cantilever type have been employed on the fingertips.
- Slip sensor. Two different types namely (i) rotational drum type slip sensor and (ii) spherical ball type slip sensor have been used.
- Motor speed sensor. Motor speed has been sensed by noncontact type sensor like optical encoder to indicate the position and thereby motor speed has been controlled through software manipulation.

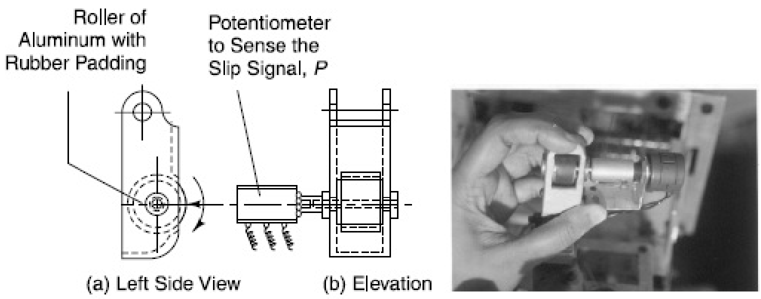
5.7.1.3. Slip Sensor

Slip may occur during different postures under grip of the robot hand. To resist the slippage, the fingers are to exert grip force depending on the weight of the object and the coefficient of friction between the part surface and the finger surface in the static condition. Due to change in the orientation of the hand, the object orientation also changes.

In order to sense the slip, three different types of slip sensors have been employed:

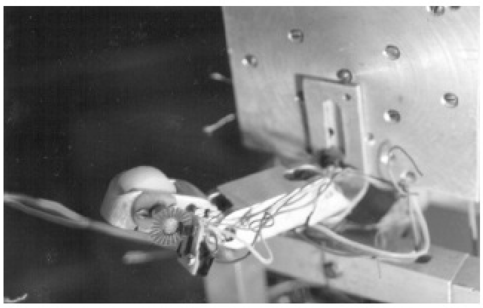
- Rotating drum type with potentiometer, shown in Fig. 5.43(a, b)
- Rotating drum type with optical encoder, shown in Fig. 5.43(c)
- Spherical ball type with optical encoder, shown in Fig. 5.44

Figure 5.43 Rotating drum type slip sensors with potentiometer (a, b) and optical encoder (c)



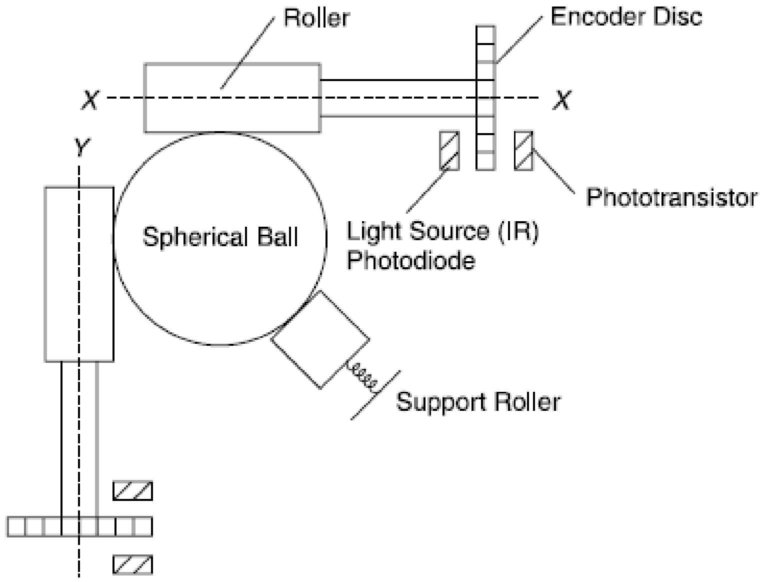
(a) Left Side View

(b) Elevation



(c)

Figure 5.44 Spherical ball type slip sensor



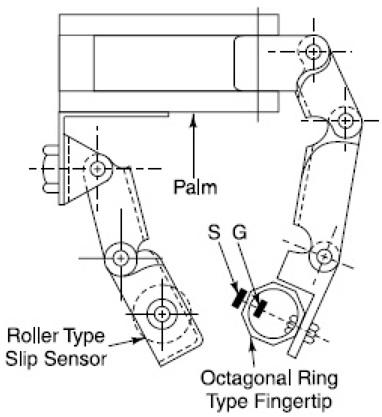
For this kind of system, the slip sensation in both the direction (*X* and *Y* as mentioned in the Fig. 5.44) are possible. The springs supported roller shown in the figure is used to keep the spherical ball in contact with the two rollers.

5.7.1.4. Force Sensor

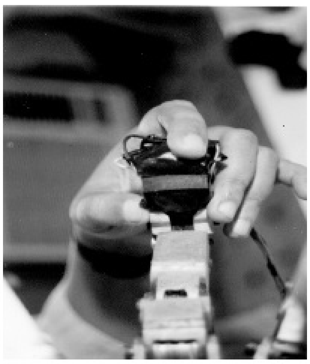
Force sensors are required to determine the gripping force to hold the object avoiding excessive pressure on the object and at the same time preventing slippage of the object. The different types of force sensors have been employed. They are:

- Octagonal ring type force sensor as shown, in Fig. 5.45
- Cantilever type force sensor, as shown in Fig. 5.46

Figure 5.45 Octagonal ring type force sensor

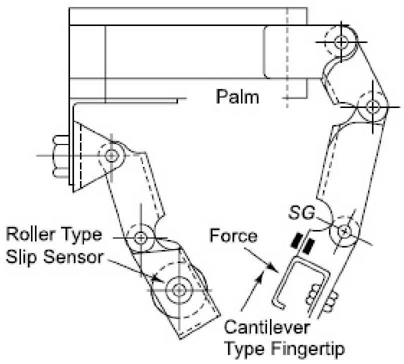


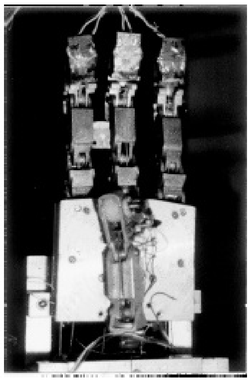
(a) Octagonal Ring



(b) Photograph of the Ring

Figure 5.46 Cantilever type force sensor



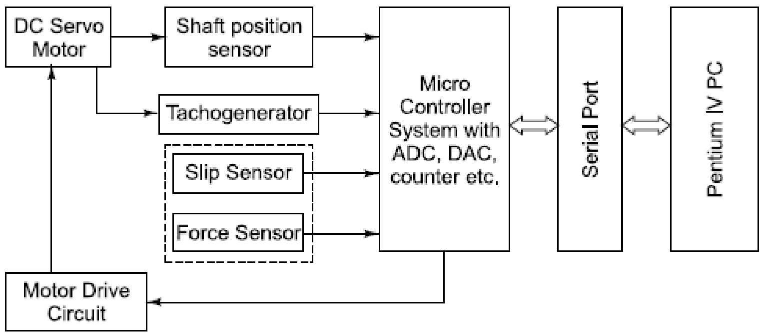


Electrical strain gauges have been used to sense the strain due to force using Wheatstone bridge circuit with proper instrumentation and amplifier circuitry.

5.7.2. Development of Control System for Tendon Actuated Dexterous and Precision Grip

The control scheme emphasizing PC-based control architecture, sensory interfaces and control system software is shown in Fig. 5.47. The microcontroller (89C52) receives signals from different sensors fitted into the gripper and actuator motor (DC servomotor) and then sends these signals to PC through RS 232 port. The control software is located in the PC which on getting the feedback from different sensors determines the required motor shaft position for successful grasping and sends approximate signal to the motor drive circuit through RS 232 port and microcontroller system.

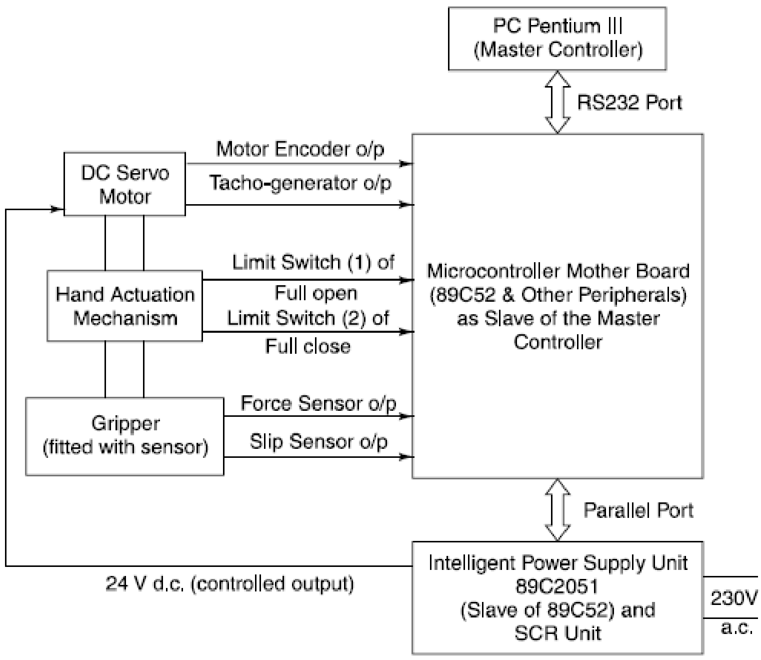
Figure 5.47 Finger motion control



The control architecture shown in Fig. 5.48 is a three-level hierarchy and consists of

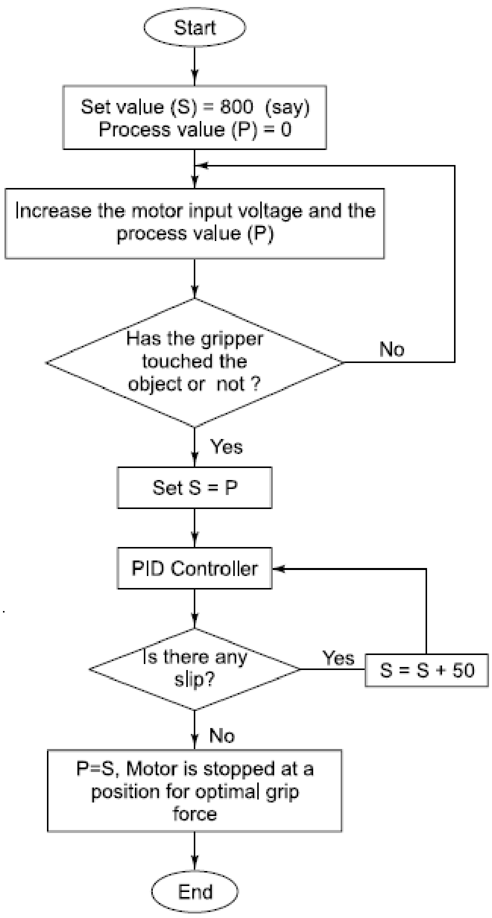
- a PC,
- a microcontroller (89C52 chip), and
- an intelligent power supply unit (89C2051 chip).

Figure 5.48 Block diagram of the control architecture



The flow chart of control algorithm is shown in Fig. 5.49.

Figure 5.49 Flow chart for control algorithm



The PC acts as the master controller. The software for the control algorithm is written in visual C++. The software calculates the error, es = (process value – set value). The computer communicates with the microcontroller that acts as a slave. The microcontroller helps the master in supplying the required data from the different sensors and limit switches. Evaluating the controller mode equation through software, the master downloads the firing angle information and motor direction information to the Microcontroller, which in turn sends these signals to the intelligent power supply unit through its parallel port. The intelligent power supply unit consists of SCR bridge circuit, driver circuit, a reversible relay unit, and microcontroller 89C2051 which detects the zero of a.c. signal by the aid of zero-crossing detector circuit and sends the firing pulse signal to the gate of SCR through driver circuit and pulse transformer. Thus a controlled voltage in appropriate direction is fed to the input of the motor so that it can achieve a position with a controlled speed.

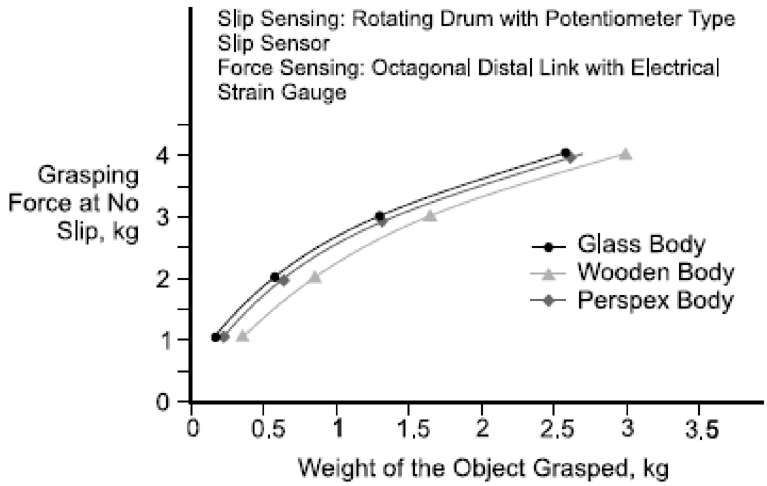
A d.c. servomotor actuates the flexion of the thumb and three fingers. The characteristic of a servomotor has a high torque at speeds near zero, and slope of the torque–speed curve is negative in operating range. The motor has rated output of 94 W at rated speed (max) of 4000 rpm. The motor has a gearbox of reduction ratio of 41.6:1 with epicyclic train. In order to control the grasp position of the gripper, the motor position is controlled by controlled input voltage. The motor should achieve the desired set position smoothly without any overshoot and undershoot.

The slip sensor continuously monitors whether the object is slipped or not. If it slips, the control software increases the motor input supply voltage and its rotational speed so that the fingers can further bend and wrap round the object to exert greater magnitude of force to prevent object slippage. The process value is the motor encoder output, and the set value is chosen by a software program. The set value is dynamic. The software program makes the set value equal to the process value.

5.7.3. Some Experiments with Slip and Force Sensors for Tendon Actuated Gripper

Three different types of sensors such as (a) force sensor (b) slip sensor and (c) motor shaft position sensors besides some limit switches have been developed and tested. The force sensors have been mounted on the distal tips of the index, middle and ring fingers. The slip sensor has been mounted on the tip of the thumb. The grasping force characteristic for no-slip conditions under different weights of the object of different materials namely glass, perspex, aluminum, and wooden objects have been noted as the grasping force depends not only the weights, but also on the surface quality. Using the rotating drum type slip sensor on the thumb, the response data for the gripping force for different materials have been plotted in Fig. 5.50.

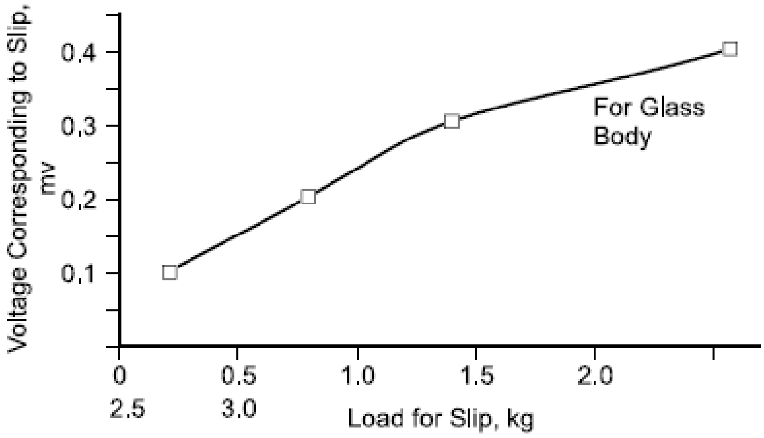
Figure 5.50 Effect of grasping force on the object to be grasped



The graph indicates that at lower load region, there is a sharp rise of force, whereas at the higher load region the curve tends to have relatively flat characteristics. This is because of the fact that at higher normal load, more asperity contacts occur on a large area between object and the gripper surface. The frictional force increases for gripping at higher load thereby allowing large weight of the object to be gripped without slip. Since the surface finish of glass object is better than machined wooden surface, the gripping force holding glass has been found to be of higher magnitude than that for holding wooden object of the same weights.

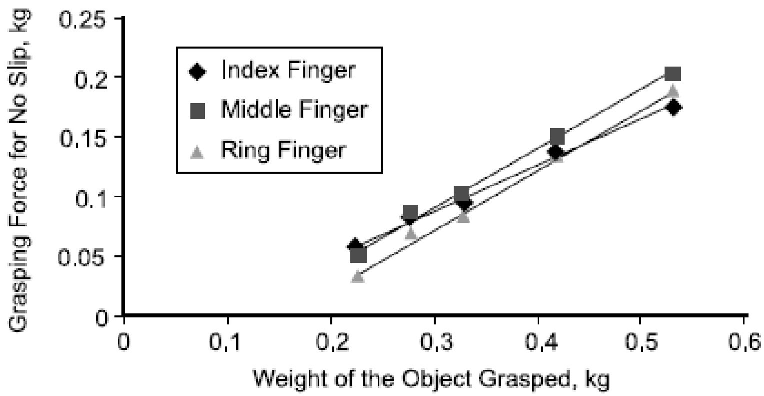
It has also been observed that glass objects of equal weights are gripped with the same amount of normal grip force irrespective of whether the slip sensor is of drum type or of spherical ball type. The response data for glass object have been shown in Fig. 5.51 with spherical ball type slip sensor. In both the cases, octagonal ring type force sensors have been employed.

Figure 5.51 Grasping force vs. slip load plot with optical encoder type slip sensor



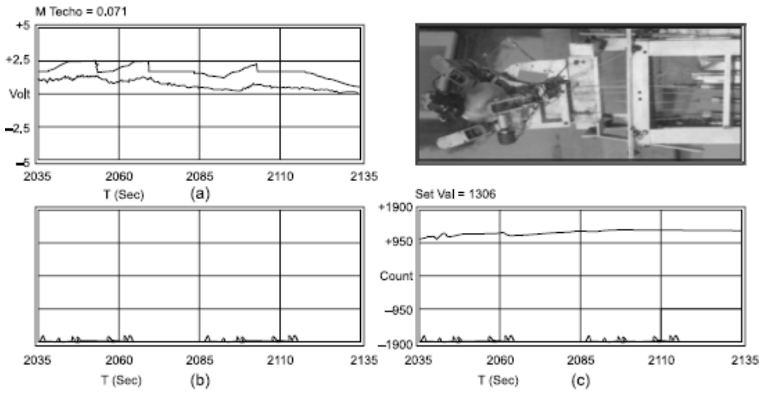
Similar experiments have been repeated with the cantilever type force sensors instead of octagonal ring types, and it has been found that cantilever type force sensors indicate higher sensitivity than octagonal type. Experimental data have been taken to observe the response of index, middle and ring finger during successful grasping of wooden object without slip, and the data are plotted in the Fig. 5.52. It has been observed that the component of the force through middle finger is always of greater magnitude compared to the other fingers as the thumb is in opposition to the middle finger.

Figure 5.52 Effect of grasping force on the weight of different objects



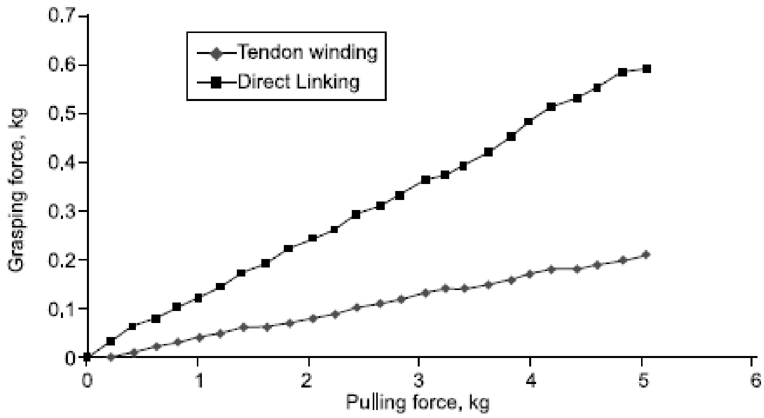
Online control characteristics during grasping of different objects have been observed. FromFig. 5.53, it has been observed that the slippage of the object has occurred before 2092.6 sec and ensures that both set and process value have been increased before 2092.6 sec to grasp the object with higher magnitude of force. The slippage has stopped after 2092.6 sec, and then both the process and set value have remained constant at 1306-count value and corresponding force is 0.373 kg.

Figure 5.53 Online control characteristics during grasp of aluminum object



Comparison of actuation forces between tendons actuated hand and direct-linked hand has been shown in the Fig. 5.54. It is evident that the direct-linked hand has been effective in exerting greater actuation force for gripping objects than that for gripping by tendon-actuated hand.

Figure 5.54 Comparative study of tendon actuated and direct-linked hand



5.7.4. Development of Control System for Direct-Linked Robot Hand

The control system for grasp planning of a direct-linked dexterous manipulator is described further. The gripper consists of distal, proximal and metacarpophalangeal joints except the thumb. The main axle is coupled to stepper motor (unlike d.c. servo motor in case of tendon actuated gripper). A single motor used for individual finger provides bidirectional rotation for opening and closing of the fingers. A number of sensors with feedback including vision system has been employed.

Vision Sensor (CCD Camera) has been used for grasp preplanning and analysis.

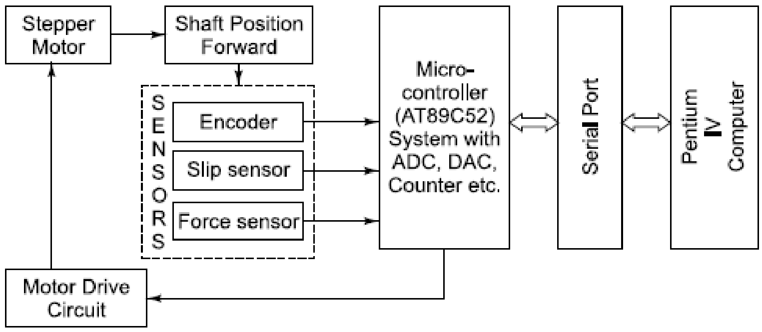
The vision-based control scheme has been employed to generate the amount of preshape. The vision assistance helps the fingers to preshape to a particular distance so that fingertips are a few units apart from the object.

Slip sensors and force sensors have also been used.

On touching the fingers, the slip sensor works and at no slip condition the force is detected by the force sensor.

The computer-based control scheme for sensor-integrated gripper is illustrated in Fig. 5.55.

Figure 5.55 The overall computer-based control scheme

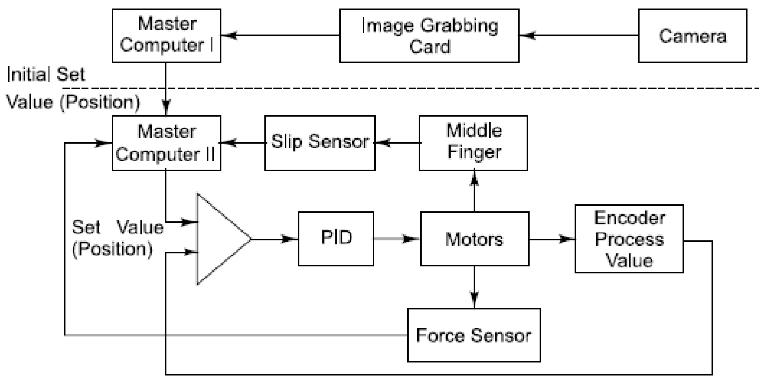


The control software is located in the computer which on getting the feedback signals from the sensors determines the required motor shaft position for successful grasping and sends corresponding signal to the motor drive circuit through RS-232 port and microcontroller system. The position, slip and force sensors have been employed.

The object shape information can be obtained by vision system. As soon as the fingers touch the object, the force sensors read the data at no-slip condition and generate a database for the grasping force for different materials.

The vision-based preshaping, slip and force feedbacks are shown in Fig. 5.56.

Figure 5.56 The vision-based preshaping and slip feedback for generation of database of force for stable grasp (at no slip condition)

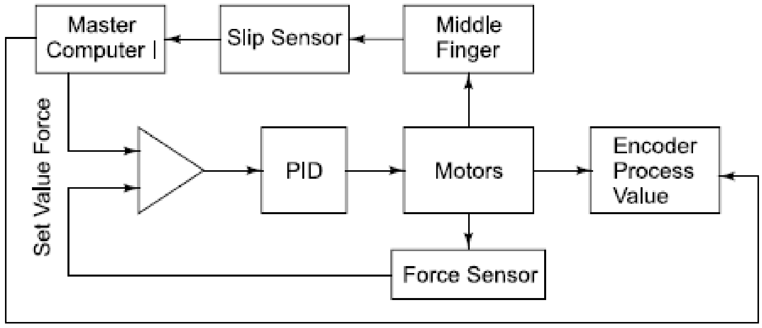


Cuboidal, Cylindrical and Spherical glass, aluminum and plastic objects have been grasped by the hand.

The schematic diagram of the force feedback with PID control is shown inFig. 5.57. The controller combines the proportional, integral and derivative modes by using the expression given by,

$$\text{PID\_OUT} = k_p e_s + k_p k_I \int_0^0 es\,dt + k_p k_D des/dt + P_I(0)$$

Figure 5.57 The force feedback for stable grasp



where, k_p is the proportional gain constant, k_I is the integral gain constant, k_D is the derivative gain constant, e_s is the error given by the difference between the set value and the process value and $P_I(0)$ = the controller output when the observation starts. The gain values can be set from the software.

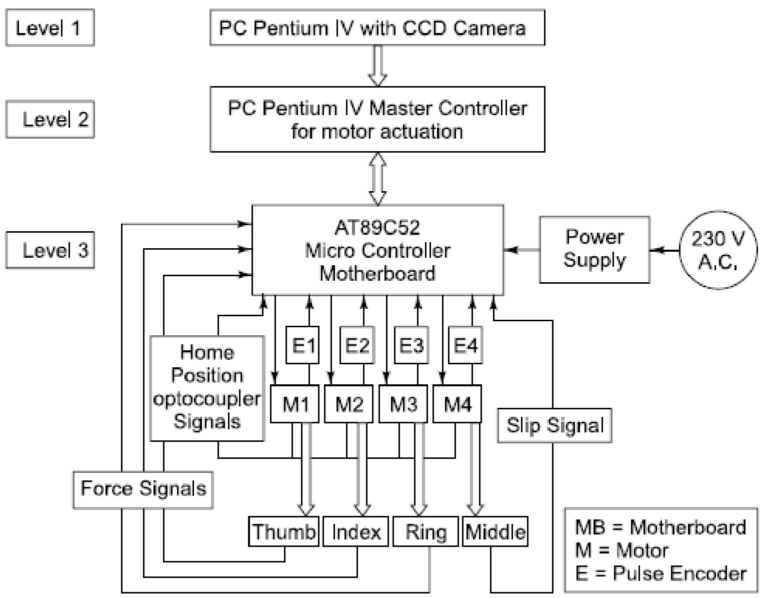
The control architecture for direct-linked robotic hand is a three level hierarchy, which consists of the following blocks:

a. the Master Computer-I with vision feedback

b. the Master Computer-II, and

c. a microcontroller Mother Board with microcontroller chip (AT89C52) as shown in Fig. 5.58.

Figure 5.58 Control hierarchy



The flow chart depicting the vision-assisted preshape and slip feedback is shown in Fig. 5.59(a), and the force feedback flow chart is shown in Fig. 5.59(b).

Figure 5.59(a) Flow chart for the vision and slip feedback

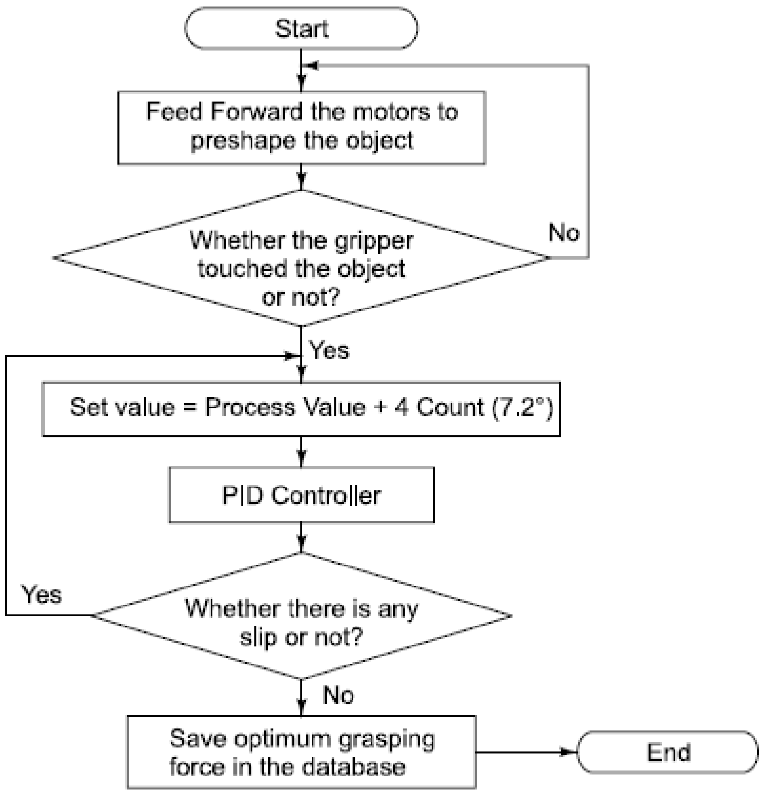
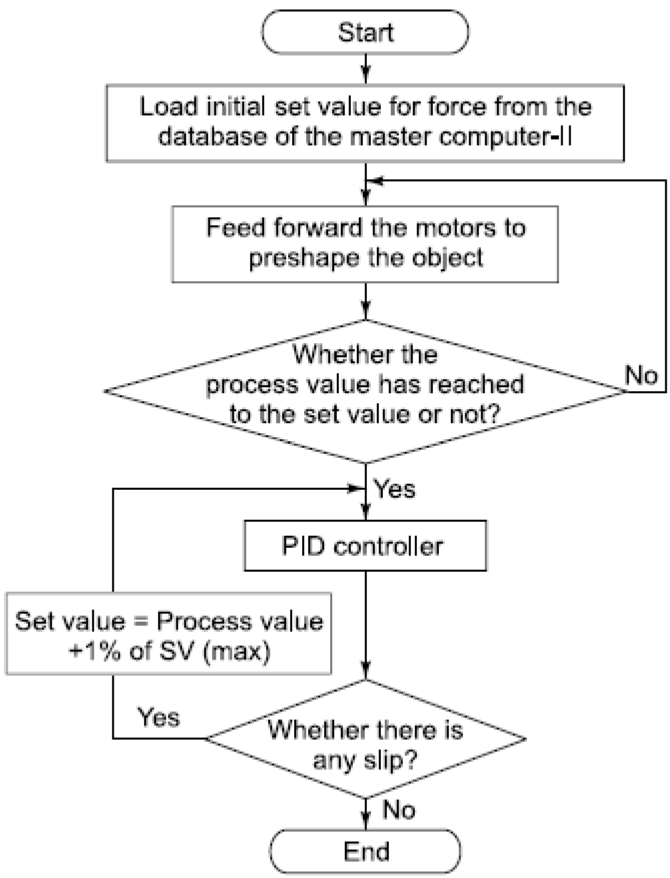


Figure 5.59(b) Flow chart showing the force feedback



Figures 5.60 and 5.61 show the photographic setups for the hand grasping objects of different shapes.

Figure 5.60 The hand grasping a cylindrical object

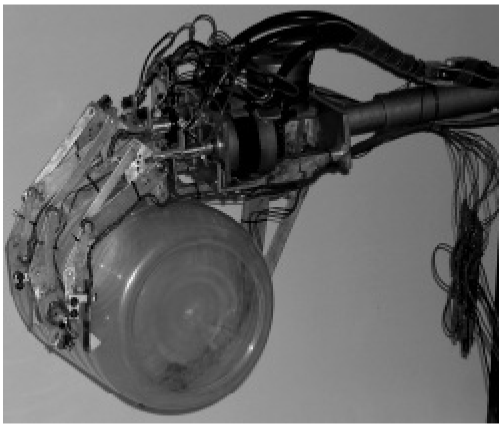
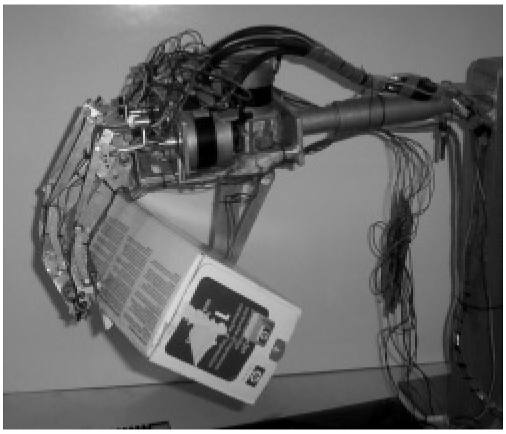


Figure 5.61 The hand grasping a cuboidal object



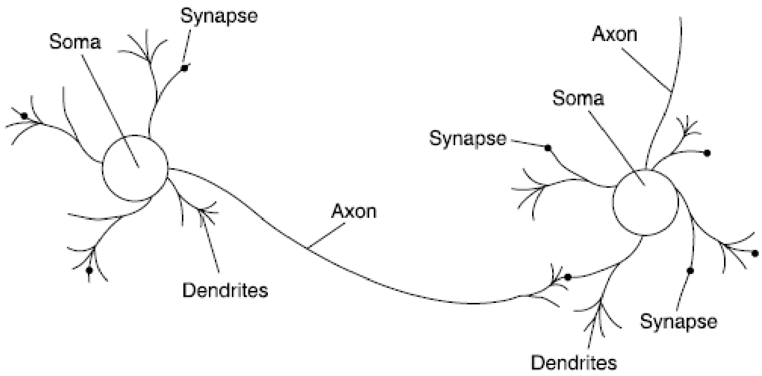
5.8. SOFT COMPUTING

The real-world problems are often too complex or ill-defined to model mathematically, or the mathematical modeling becomes highly nonlinear. Moreover, some sort of imprecision of input data and uncertainty of knowledge are always inherent to these problems. For these reasons, it may not be always possible to solve such problems using the traditional hard-computing methods. In such cases, modeling based on soft computing tools like neural networks, fuzzy logic, genetic algorithm and particle swarm optimization may be used which, unlike hard computing methods, are tolerant of factors like imprecision, uncertainty, partial truth, etc. and can quickly generate approximate solutions to such problems even in the presence of the above factors. Moreover they offer other advantageous features like learning and adaptation. The fundamentals and scope of application of the four aforementioned soft computing tools will be briefly discussed here.

5.8.1.1. Neural Network

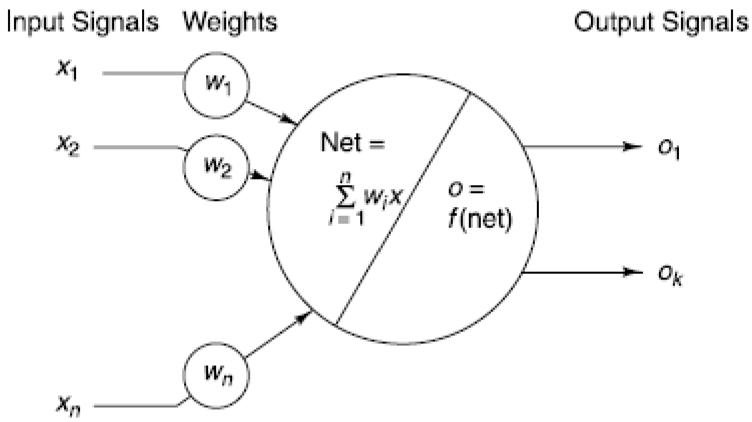
An artificial neural network is a model of reasoning based on the human brain. The brain contains a neural network consisting of a large number of interconnected information-processing units called neurons. A schematic diagram of a neural network showing only two neurons and their connections is shown in Fig. 5.62. A neuron consists of a cell body called the soma, a number of fibres called dendrites and a single long fibre called the axon. The dendrites receive the electrical signals from the axons of other neurons. The axon transmits the electrical signal from one neuron to other neurons via the dendrites. The connection between the axon and the dendrite is called a synapse. At the synapses, the electrical signals are modulated by different amounts. The synapses release chemical substances that cause changes in the electrical potential of the soma. When the potential reaches its threshold, an electrical pulse called the action potential is sent down through the axon.

Figure 5.62 A schematic diagram of biological neural network



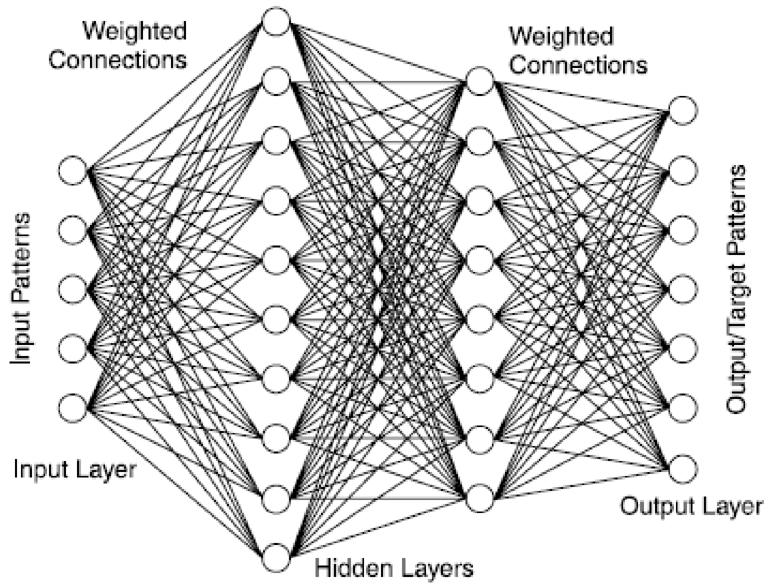
An artificial neural network is designed to act fundamentally like a biological neural network. Figure 5.63 shows a schematic diagram of an artificial neuron receiving input signals (x_i, i = 1 to n) from a number of neurons and emitting output signals (o_i, i = 1 to k) to be transmitted to a number of neurons. In an artificial neural network, the signals are in the form of numerical values rather than electrical signals. The action of the synapses is simulated by multiplying each input value by a suitable weight w. A biological neuron fires an output signal only when the total strength of the input signals exceeds a certain threshold. This phenomenon is modelled in an artificial neuron by calculating the weighted sum of the inputs to represent the total strength of the input signals, and applying a suitable activation (threshold) function to the sum to determine its output.

Figure 5.63 Schematic diagram of an artificial neuron



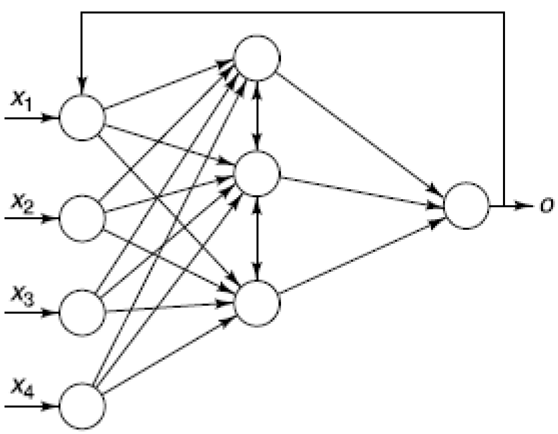
Neural networks can be classified based on their topology (architecture) and the method of training. The most common neural network architectures are (1) feedforward neural networks, (2) feedback neural networks, and (3) self-organizing neural networks. Feedforward neural networks are the most popular and widely used ones. There are two types of neural networks in this category: multilayer perceptron (MLP) neural network and radial basis function (RBF) neural networks. Figure 5.64 shows a feedforward architecture of a typical neural network consisting of a number of layers. Each layer contains a number of neurons, depicted by circles in the figure. Each layer has full interconnection to the next layer but no connection within a layer. The first layer of the network is known as the input layer whose neurons take on the values corresponding to different variables representing the input pattern. The second layer is known as a hidden layer because its outputs are used internally and not used as the final output of the network. In MLP neural network, there may be more than one hidden layer (two hidden layers are shown in the figure). The final layer of network is known as the output layer. The values of the neurons of the output layer constitute the response of the neural network to an input pattern presented at the input layer. In MLP neural networks, the number of hidden layers in a network is an important design parameter. Further the number of neurons in a hidden layer also has to be chosen judiciously. The general rule is to design a neural network model that uses fewer parameters (weights and biases).

Figure 5.64 Feedforward architecture of a typical neural network



In recurrent or feedback neural networks, the outputs of some neurons are also fed back to some neurons in layers before them. Thus, signals can flow in both forward and backward directions, as shown in Fig. 5.65. A recurrent network is said to have a dynamic memory. The output of such networks at a given instant reflects the current input as well as the previous inputs and outputs. An example of the recurrent networks is the Hopfield network. Hopfield networks are typically used for classification problems with binary input pattern vectors. Another type of neural network is the self-organizing neural network that consists of neurons arranged in the form of a low-dimensional grid. Each input is connected to all the output neurons. This type of network is useful in classifying high-dimensional data by constructing its own topology.

Figure 5.65 A typical recurrent neural network



Neural networks need to be trained so that they produce proper response to a given input vector. The training process is an iterative process that adjusts the parameters (weights and biases) of the network until the network is able to produce the desired output from a set of inputs. The process of training the network can be broadly classified into supervised and unsupervised learning. A number of training algorithms based on the supervised learning are available of which the most common is the backpropagation algorithm. The backpropagation algorithm supplies the neural network with a sequence of input patterns and desired output (target) patterns, which together constitute the training exemplars. As an input pattern is presented to the neural network, the output response is calculated on a forward pass through the network. In Fig. 5.64, the output of each neuron j in the hidden layer is computed according to the model of Fig. 5.63. Given the input signal x_i at the input neurons, the output o_j is given by

$$o_j = f(\Sigma w_{ji} x_i)$$

(5.12)

where w_{ji} is the weight associated with the jth neuron of the hidden layer and the ith neuron of the input layer. The function f is called the activation function. Some commonly used activation functions are:

$$\text{Log sigmoid: } o = f(t) = \frac{1}{1+e^{-ct}}\text{, where } c \text{ is a constant}$$

(5.13)

$$\text{Tan hyperbolic: } o = f(t) = \tanh\,(ct/2)\text{, where } c \text{ is a constant}$$

(5.14)

$$\text{Identity: } f(t) = t$$

(5.15)

The output of each neuron in the output layer is computed in a similar manner. The final output is compared to the desired output, and error terms are calculated for each output neuron. A function of the errors of the neurons in the output layer is then propagated backwards through the network to each layer, and weights of each of the interconnected neurons are adjusted in such a way that the error between the desired output and the actual output is reduced. Any optimization method can be used to find out the weights that minimize the error. Early backpropagation algorithms were based on the steepest-descent algorithm, according to which the maximum decrease in the function is in a direction opposite to the direction of the gradient of the function. Another commonly used backpropagation algorithm is based on the Levenberg–Marquardt method, which is a combination of the steepest-descent method and the quasi-Newton method. At initial iterations, Cauchy's method is followed and the algorithm gradually moves towards the quasi-Newton method.

In back propagation algorithms using the steepest-descent or Levenberg-Marquardt method, the weights may correspond to a local minimum only. This problem can be solved by adding a momentum term to the training rule, which forces the weights to keep moving in the same direction in the error surface without becoming trapped in a local minimum. Momentum simply adds a fraction of the previous weight update to the current weights. This increases the size of the step taken towards minimum. It is, therefore, necessary to reduce the learning rate (a factor that determines how much change in the weights is carried out at each step) when using a lot of momentum. If a high learning rate is used, the algorithm may oscillate around the minimum and may become unstable. Too small a learning rate will take a long time to converge. The optimum value of learning rate is often found by trial and error. The process of training the neural network using supervised learning is applicable to problems where representative exemplars of both input pattern and output (target) patterns are known. In many problems where the target patterns are unknown, unsupervised learning is used. In the unsupervised learning process, the network is provided with a dataset containing input patterns but not with desired output patterns. The unsupervised learning algorithm then performs clustering of the data into similar groups based on the measured attributes or features of the given input patterns serving as inputs to the algorithms. After the training of the neural network by a suitable method, the trained network needs to be tested with unseen data (not included in the training dataset).

Neural networks have been used successfully for a number of applications in different fields like, for example in manufacturing for modeling and control of manufacturing processes, prediction of product and process quality parameters, monitoring of manufacturing processes and so on, in process planning for automating various decision making tasks like feature recognition from CAD database, selection of manufacturing operations, selection of optimum values of manufacturing process conditions, etc., in production scheduling, in robotics for developing algorithms for controllers, for robot path planning and navigation, in computer vision for developing algorithms for pattern recognition and classification and so on.

5.8.1.2. Fuzzy Logic

Fuzzy logic is a branch of logic especially designed for approximate reasoning using imprecise propositions based on fuzzy set theory, in a way similar to classical reasoning using precise propositions based on the classical set theory. Therefore in order to understand this notion, it is useful to first understand how the classical reasoning works using the classical set theory. The classical set theory is governed by the logic that uses only one of the two possible truth values: true or false. Thus an element either belongs to a set or does not belong to it. In other words, an element has a degree (grade) of membership of either 100% (1) or 0% (0) within the set. To understand this concept, let us consider an example. Suppose that perfectly clean water is defined as water in which there is no suspended material or dirt. Then according to the classical set theory, a statement that the water is perfectly clean is considered to be true only if there is no dirt present in the water. However, even if the water is fairly clean i.e. there is only a small amount of dirt in the water, no matter however small it may be, the above statement will be termed as false. Thus the classical set theory does not allow for representing the degrees of imprecision, indicated by words or phrases such as fairly, very, quite possibly and so on. On the other hand, in case of fuzzy set theory, it is possible to have multivalued logic in addition to the two possible truth values, true and false such as for example, very true, not very true, more or less true, not very false, very false, etc. In other words, a proposition may be partly true or partly false to a certain degree. Thus an element belongs to a fuzzy set with a certain degree of membership that may vary from 100% (1) to 0% (0). If the above example is again considered, according to fuzzy set theory, perfectly clean water will have a degree of membership of 100% (1) in the fuzzy set defining clean water. However, water with a small amount of dirt is still fairly clean, perhaps 90% clean and so it will have a degree of membership of 90% (0.9) in the clean water set. Dirty water would also be part of the clean water set, but may have a degree of membership of 0% (0). If, on the other hand, a fuzzy set called dirty water is also defined, then the perfectly clean water will have a degree of membership of 0% in the dirty water set, while dirty water will have a 100% degree of membership in that set. Water with a 90% degree of membership in clean water set may have a different degree of membership, say 15% in the dirty water set. Thus the water under consideration may have two different degrees of membership: one in the clean water set and the other in the dirty water set.

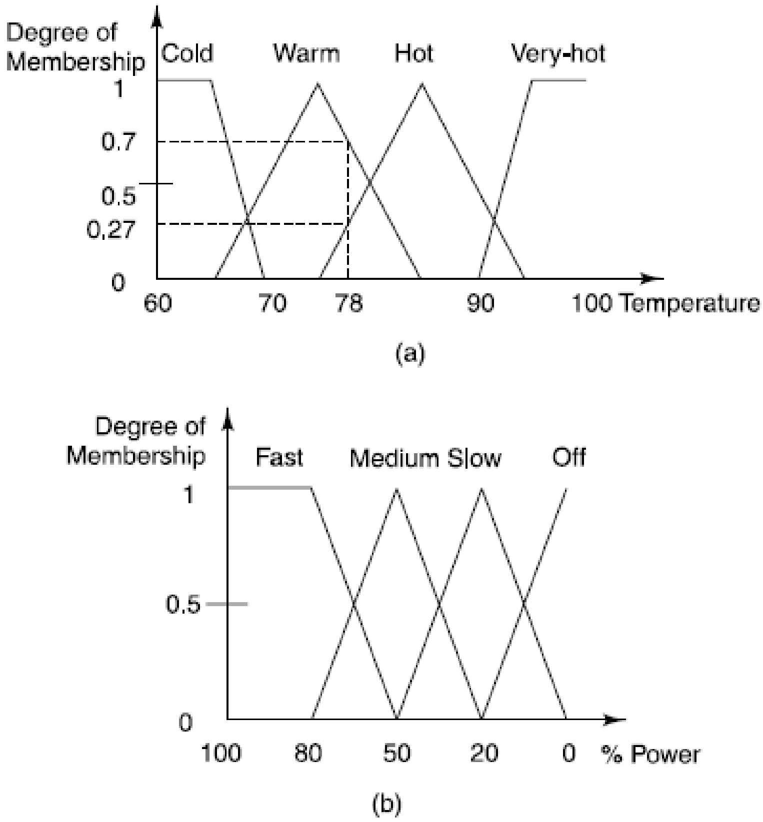
Since the concept of fuzzy set theory was first introduced by L.A. Zadeh around 1965, it has been used for a number of different applications such as for developing fuzzy logic controllers, fuzzy clustering algorithms, fuzzy mathematical programming and so on. Out of all such applications, the fuzzy logic controller is probably the most commonly used one and will be discussed here. The fuzzy logic controller was first developed by E. Mamdani around 1975. The Mamdani style fuzzy inference process is performed in the following three steps: fuzzification, rule evaluation and defuzzification.

Fuzzification It is the process of expressing the crisp input and output values in the fuzzy form by determining their degrees of membership in the appropriate fuzzy sets. In General, a fuzzy set $A(x)$ is represented by a pair of values, the first one being the element x and the second one being its membership value denoted by $\mu_A(x)$ as given further.

$$A(x) = \{(x,\ \mu_A(x)),\ x \in X\}$$

Let us consider the variable temperature. In order to express the crisp value of the temperature variable in the fuzzy form, the desired range of temperature is first of all divided into a number of linguistic fuzzy sets such as Very-Hot, Hot, Warm and Cold. For example, let us suppose that the desired temperature range is between 60° and 100°F. Then the temperature ranges assigned to the above fuzzy sets may be as shown in Fig. 5.66(a).

Figure 5.66 Fuzzification



The corresponding membership functions can be expressed as follows:

Very-Hot	:	{(90,0), (95,1), (100,1)}
Hot	:	{(75,0), (85,1), (95,0)}
Warm	:	{(65,0), (75,1), (85,0)}
Cold	:	{(60,1), (65,1), (70,1)}

A crisp value of the temperature variable is then fuzzified against the appropriate fuzzy sets by assigning the appropriate degrees of membership in the corresponding fuzzy sets. For example, a temperature of 78°F may be assigned a membership value of 0.27 in the fuzzy set Hot and 0.7 in the fuzzy set Warm.

Rule Evaluation At the core of the fuzzy logic controller are the fuzzy rule base consisting of fuzzy rules and the fuzzy inference engine to evaluate the fuzzy rules. A fuzzy rule is defined as a conditional statement and can take one of the following forms

Rule 1: IF *x* is *A*1

THEN *z* is *C*1.

Rule 2: IF *x* is *A*2

AND *y* is *B*2

THEN *z* is *C*2.

Rule 3: IF x is $A3$

AND y is $B3$

THEN z is $C3$.

where the IF part is called the antecedent of the rule and the THEN part is called the consequent of the rule, x, y and z are linguistic variables defined in the universes of discourse X, Y and Z respectively, and A, B and C are linguistic values determined by fuzzy sets on the universes of discourse X, Y and Z respectively.

The next step following the fuzzification is to take the fuzzified inputs and apply them to the antecedents of the fuzzy rules. If a fuzzy rule has multiple antecedents, in order to evaluate the disjunction of the rule antecedents, the OR fuzzy operation is used, the result of which is the maximum of the two values. In order to evaluate the conjunction of the rule antecedents, the AND fuzzy operation is used, the result of which is the minimum of the two values.

Defuzzification The final step in the fuzzy inference process is defuzzification, which is the conversion of a fuzzy output value to an equivalent crisp number for actual use. For example, suppose that the output power setting for an air-conditioning system is fuzzified into the different fuzzy sets Off, Low, Medium and High as shown in Fig. 5.66(b) and suppose that the result of the rule evaluation is 25% membership in Low and 75% membership in Medium. Defuzzification converts these fuzzy output values into a single crisp number, which can be sent to the air-conditioning system. There are a number of methods of defuzzification out of which a commonly used method is the Centre-of-Gravity method. In this method, the membership degree of the output variable is multiplied by the corresponding maximum singleton value of the output fuzzy set in question and the resulting values for each set are added together and divided by the summation of output membership degrees to obtain the equivalent crisp value of the output. Mathematically, it is given by the following formula

$$\text{Output} = \frac{\sum_{x=a}^{b} \mu_A(x).x}{\sum_{x=a}^{b} \mu_A(x)}$$

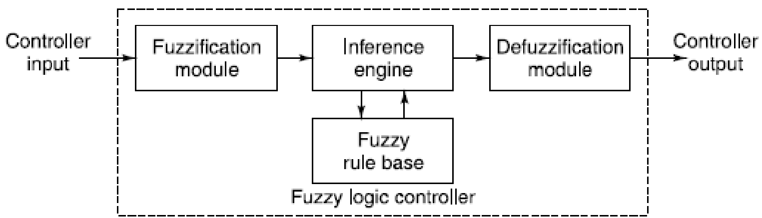
(5.16)

For example, suppose that the degrees of membership of the output power setting of an air-conditioning system are 0.4 for Low and 0.6 for Medium and the maximum singleton value for Low is 30% and for Medium is 50% of full power. The crisp output value for the air-conditioning system would be

$$\text{Output} = \frac{0.4 \times 30\% + 0.6 \times 50\%}{0.4 + 0.6} = 42\%$$

The overall fuzzy logic controller that combines the 'fuzzification-rule base-inference engine-defuzzification' module is shown in Fig. 5.67.

Figure 5.67 General structure of a fuzzy logic controller



Fuzzy logic has been successfully used in applications involving modeling and control of various manufacturing processes, in process planning for automating selection of manufacturing operations, selection of manufacturing process conditions, manufacturing set-up planning and other decision making tasks, applications in robotics for control of robot manipulator, autonomous mobile robot navigation, in data clustering which is a powerful method of data mining to extract useful information from datasets, and others.

5.8.1.3. Genetic Algorithms

Genetic algorithms (GAs) are a class of population-based search and optimization algorithms first introduced by J. Holland in 1965 after getting inspired by the mechanics of natural genetics and natural selection. They are unlike many conventional search algorithms in the sense that they simultaneously consider many points in the search space. They usually work with strings of typically binary numbers or bits (0/1) representing the problem variables. A binary string can be compared to a biological chromosome and each bit of the string compared to a gene value. Let us consider the problem of optimization of a machining operation with the cutting speed and the feed as the decision variables. Suppose that the variable cutting speed is in the range 0–310 m/min; it can be represented with a 5-bit binary number with 00000 representing zero speed and 11111 representing the speed of 310 m/min. The length of a binary string is decided based on a desired accuracy in the values of the variables. A finer resolution may be obtained by choosing a higher bit number. Similarly let us consider that a cutting feed in the range of 0 to 0.15 mm/rev is represented by a 4-bit number with 1111 representing the feed of 0.15 mm/rev in the usual way. A typical combination of cutting feed and speed can be represented by putting these two binary numbers together to make it a 9-bit string. For example, 100001010 is a string representing a cutting speed of 160 m/min and feed of 0.1 mm/rev. The search for the optima is started by taking a number of random strings forming the initial population. Choosing an appropriate population size is crucial. The population of strings is successively evolved by three operators: reproduction, crossover and mutation.

In reproduction, the good strings in the population are probabilistically assigned a large number of copies. The performance of the strings often called the fitness is used to decide how good the string is. The fitness is evaluated with the help of some function representing the constraints of the problem. For infeasible solution, a very low value of fitness may be assigned. There are a number of ways to do reproduction. One popular and easy-to-implement method is the tournament selection. In this, each string plays duel tournaments with two other randomly chosen strings. In a tournament, the fitnesses of the two strings is compared and the winning string is retained. It is clear that the best string will always remain and the worst string will be eliminated from the population. The fate of the other strings depends on chance. Finally, it is expected that as a result of this operation, the number of good strings will be more than the bad ones in the population.

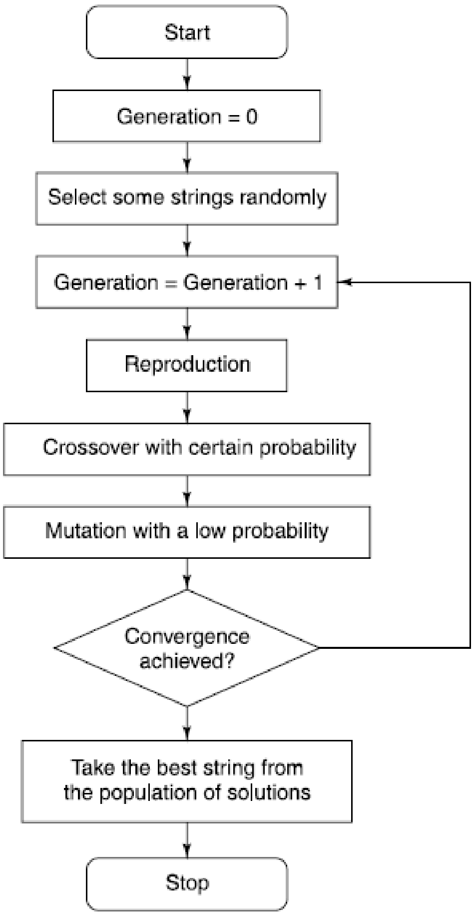
The second operation is crossover. In this, two new strings are created by exchanging the bits between two strings. For example, consider two strings chosen at random from a population, called parent strings. A single-point crossover operation is performed by randomly choosing a crossing site along the string and by exchanging all the bits on the right-hand side of the crossing site as shown:

$$\begin{array}{c|c} 010001 & 111 \\ 100110 & 011 \end{array} \Rightarrow \begin{array}{c|c} 010001 & 011 \\ 100110 & 111 \end{array}$$

The new strings formed are called the children strings. The crossover operation is performed with certain crossover probability, usually lying between 0.75 and 0.95.

In the last operation mutation, one or more bit of a string may be changed randomly. However, the probability of mutation is kept quite low, between around 0.01 and 0.05. The mutation operation serves the crucial role of preventing the algorithm from becoming stuck in the local optimum. After the application of three GA operators in succession, a new generation is formed and an iteration is said to be complete. The iterations are carried out until the average fitness in successive generations more or less becomes constant. The entire methodology is illustrated by a flowchart in Fig. 5.68.

Figure 5.68 A flow chart illustrating the methodology of genetic algorithm



Genetic algorithms have been successfully used for solving various optimization problems in different areas like in product design, in manufacturing for process parameter optimization, process planning, machine component grouping in cellular manufacturing systems, in robotics for optimum path generation for navigation and obstacle avoidance, in production scheduling, routing, and others.

5.8.1.4. Particle Swarm Optimization

The particle swarm optimization (PSO) is a population-based algorithm for the search of approximate solutions to optimization problems, developed by Eberhert and Kennedy in 1995 after getting inspired by the social behavior of a flock of birds while searching for food, where social sharing of information takes place and individuals can profit from the discoveries and previous experience of all other companions. Each individual (called a *particle*) in the population (called a *swarm*) is assumed to fly over the search space looking for promising regions of the landscape. For example, for a function minimization problem, the promising regions could be those having lower function values compared to the others visited previously. Each particle moves with an adaptable velocity within the search space and retains a memory of the best position ever attained by it, as well as the best position ever attained by any individual particle of the swarm. Each particle then adjusts its own flying according to its own flying experience (i.e. its own best position) as well as the flying experience of its other companion particles (i.e. best position of the swarm).

The PSO algorithm is initialized by a population of potential solutions, which is collectively called a swarm. Each potential solution, also called a particle, is assigned a randomized position and velocity. The performance of each particle is measured according to a predefined fitness function. The particles are then flown through the problem space. Suppose that the search space is D-dimensional. Let the ith particle of the swarm be represented by a D-dimensional vector

$$X_{id} = (x_{i1}, x_{i2}, \dots, x_{iD})^T$$

where d = 1, 2,…, D and i = 1, 2,…, N, and N is the size of the swarm.

Let the velocity (position change) of the particle be represented by another D-dimensional vector

$$V_{id} = (v_{i1}, v_{i2}, \dots, v_{iD})^T$$

Each particle as it moves through the problem space keeps track of the best solution or fitness achieved by it so far along with its location. This value is called *p*best. Suppose that the best previously visited position of the i^{th} particle is denoted as

$$P_{id} = (p_{i1}, p_{i2}, \dots, p_{iD})^T.$$

Another best value that is tracked is the overall best value, and its location, obtained so far by any particle in the swarm. This value is termed *g*best. Suppose the position of the g^{th} particle is the overall best previously visited position of the swarm and is denoted by

$$P_{gd} = (p_{g1}, p_{g2}, \dots, p_{gD})^T.$$

At each time step, the velocity and position of each particle is modified so that it moves towards the *p*best and *g*best according to the following equations

$$\begin{aligned} v_{id}^{t+1} &= \chi\{wv_{id}^{t} + c_1 Rand()^t \times (p_{id}^{t} - x_{id}^{t}) + c_2 Rand()^n \times (P_{gd}^{t} - x_{id}^{t})\} \\ x_{id}^{t+1} &= x_{id} + v_{id}^{t+1} \end{aligned}$$

(5.17)

where χ is called constriction factor, w is an inertia weight, c_1 and c_2 are positive constants called cognitive and social parameters respectively, *rand*( ) and *Rand*( ) are random values in the range [0,1] and t determines the iteration number.

The first equation is used to calculate i^{th} particle's new velocity by taking into consideration three terms: the particle's previous velocity, the distance between the particle's previous best and current positive and finally, the distance between the swarm's best (the position of the best particle in the swarm) and the i^{th} particle's current position. Then following the second equation, the i^{th} particle flies towards a new position. The role of constriction factor χ is used to control and constrict the magnitude of the velocity in constrained optimization probles (in unconstrained optimization problems, it is usually set equal to 1.0). The role of the inertia weight w is considered very important in PSO convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle towards the *p*best and *g*best positions. The cognitive parameters c_1 represent the tendency of individuals to duplicate past behaviors that have proven to be successful, whereas the social parameters c_2 represent the tendency to follow the success of others.

PSO algorithm has been used successfully for solving optimization problems in design, control, process planning, production scheduling, in robotics for path planning and obstacle avoidance and so on.

5.9. EXERCISES

5.1 What is AI? How is it connected with Robotics?

5.2 What are the functions of the sensors?

5.3 What are the different types of sensors? Classify them.

5.4 Distinguish between tactile and non-tactile sensors. Give examples of each type.

5.5 Discuss the following in the light of robotics sensors: response, accuracy, sensitivity and linearity.

5.6 What is meant by:

a. range sensor, and

b. proximity sensor

5.7 How do you sense the positional accuracy of a robot? What is the suitable type of sensors to measure the position.

5.8 Write short notes on LVDT and potentiometer.

5.9 With the aid of a sketch, describe briefly a strain-gauge type wrist force/ torque sensor.

5.10 What is a slip sensor? Suggest some transducer to measure multi directional slip.

5.11 Write short notes on inductive proximity switch, acoustic sensor and artificial skin.

5.12 What is pattern recognition? Briefly describe a sensing device to generate the contour picture of a workpiece.

5.13 What is robot vision? What are the types of vision sensor used to take the image of an object?

5.14 Distinguish between CID and CCD cameras. How does a Vidicon camera work? Distinguish between line scan and area scan vision sensors.

5.15 What are the functions of a vision processor? What are the steps necessary in the image processing?

5.16 What is thresholding? Why is it necessary?

5.17 Distinguish between gray and binary pictures. Describe their relative advantages in real time image processing.

5.18 What is template matching? Describe briefly the method.

5.19 What do you mean by the following terms:

a. sensing

b. segmentation

c. feature extraction

d. recognition

e. interpretation and scene understanding

5.20 What do you mean by edge detection? Describe a quantitative technique to find out the edges of an object.

5.21 How do you recognize an object? What are the possible features that may be extracted to identify an object?

5.22 Illustrate an application example where computer vision may be employed.

5.23 Give an outline of the control scheme for a sensors integrated robot hand.

5.24 What are the different soft computing techniques? What are the advantages of their applications in the field of automation.

5.25 How is an artificial neural network modelled after the biological neural network?

5.26 Explain the principles of supervised and unsupervised learning of artificial neural network.

5.27 Explain the principle of a fuzzy logic controller, describing briefly its different components.

5.28 Explain the role of different operations namely reproduction, crossover and mutation in a genetic algorithm.

5.29 Explain briefly the steps in a particle swarm optimization algorithm.

5.10. BIBLIOGRAPHY

Agin, J.G., "Vision Systems", *Handbook of Industrial Robots,* Ch. 14, Ed. S.Y. Nof, 1985.

Bejczy, A.K., "Issues in Advanced Automation for Manipulator Control", *Proceedings of the Joint Automatic Control Conference,* 1976.

Bejczy, A.K., "Smart Sensors for Smart Hands", *AIAA/NASA Conference on 'Smart' Sensors,* Hampton, VA, Nov. 14–16, 1978.

Bejczy, A.K., *et al.,* "Evaluation of Proximity Sensor Aided Grasp Control for Shuttle Arms", *Proceedings of 15th Annual Conference on Manual Control,* Wright State Univ., Dayton, OH, March, 20–22, 1979.

Brik, J., *et al.,* "Image Feature Extraction Using Diameter Limited Gradient Direction Histograms", IEEE Trans. Vol. PAM-1, No. 2, April, 1979.

Deb, S.R., *et al.,* "A Slip Sensor for Robot Grippers with Feedback", *13th All India Machine Tool Design and Research Conference*, Calcutta, Nov., 1988.

Deb, S.R., "Computer Vision Makes Robots More Smart", *Feedback* (A Journal of the R.C.C., Calcutta), 8, No. 9–10, Sept/Oct., 1985.

Ejiri, M., *et al.,* "A Process for Detecting Defects in Complicated Patterns", *Computer Graphics & Image Processing,* 1973.

Fu, K.S., R.C. Gonzalez, and C.S.G. Lee, "Robotics: Control, Sensing, Vision and Intelligence", McGraw-Hill International Editions, 1987.

Gonzalez, R.C. and P. Wintz, *"Digital Image Processing",* Addison-Wesley, Reading, MA, 1977.

Harmon, L.D., "Automated Tactile Sensing", *Robotics Research,* 1, No. 2, MIT Press, Cambridge, MA, 1982.

Harmon, L.D., *Touch-sensing Technology: A Review Technical Report,* Society of Manufacturing Engineers, MI, 1980.

Holman, J., *Experimental Methods for Engineers,* 3rd ed., McGraw-Hill, NY, 1978.

Holland, S.W. *et al., "CONSIGHT—1: A Vision Controlled Robot System forTransferring Parts from Belt Conveyors", Computer Vision and Sensor-based Robots,* Plenum Press, NY, 1979.

Jarvis, J.F., "Experiments in the Automation of Visual Inspection", *Proc. of the Joint Automatic Control Conference,* PA, 1, 1978.

Kelley, R.B., *et al.,* "A Robot System Which Acquires Cylindrical Workpieces from Bins", *IEEE Trans. SMC,* 12, 1982.

Konig, M., Contactless *recognition and position determination of objects by incoherent optical correlation,* Kransskopt–Verlag Buabh, Moinz, 1977.

Kuo, B.C., *Automatic Control* Systems, 4th ed., Prentice-Hall, Englewood Cliffs, N.J., 1982.

Lord Corporation, "Lord Tactile Sensors", *Marketing /Technical Brochure,* Carry, NC. Lang, H., *Programmable Workplece Recognition Device,* Montage and Hand-habugstechnik 3, 1976.

Mese, M., *et al.,* "An Automatic Position Recognition Technique for LSI Assembly", *Proc. 5th International Joint Conference on Artificial Intelligence,* 1977.

Mundy, J.L., "Automatic Visual Inspection", *Proc. Conference Decision and Control,* 1977.

Nitzan, D., *et al., Machine Intelligence Research Applied to Industrial Automation*—9th Report, 1979, 10th Report, 1980, SRI Projects, SRI International, CA.

Nevins, J.L., and D.E. Whitney, "Computer Controlled Assembly", *Scientific American,* February, 1978.

Pavlidis, T., "A Review of Algorithms for Shape Analysis", CG*IP,* 7, 1978.

Perkins, W, "Model-Based Vision System for Scenes Containing Multiple Parts", *Proceedings of the 5th International Joint Conference on Artificial Intelligence,* Cambridge, 1977.

Rosen, C.A., *et al.. Exploratory Research in Advanced Automation,* Second and Third Report, Stanford Research Institute, N.S.F. Grant, 1974.

Rosenfield, A. and Y. Nakagawa, "A Note on Polygonal and Elliptical Approximation of Mechanical Parts", *Pattern Recognition,* II, 1979.

Rosen, C.A. and D. Nitzan, "Use of Sensors in Programmable Automation", *Computer,* IEEE, December, 1977.

Seltzer, D.S., "Tactile Sensors Feedback for Different Robot Tasks", Robots 6 *Conference, Proceedings,* March 2–4, 1982, Detroit, Mich. Robotics International of SME, Dearborn, MI, 1982.

Thomson, A.M., "Camera Geometry for Robot Vision', *Robotic Age,* 3, 1981.

Thompson, W.B., and R.P. Krugar, "A Technical and Economic Assessment of Computer Vision for Industrial Inspections and Robotic Assembly", *Proc. IEEE, 69,* No. 12, 1982.

Thompson, A.M., "Introduction to Robot Vision", *Robotics Age,* Summer, 1979.

Wang, S.S.W. and P.M. Will, "Sensors for Computer Controlled Mechanical Assembly", *The Industrial Robot,* March, 1978.

Wilz, J.H., "Chain Code", *Robotics Age,* 3, April, 1981.