3-1-1987

Sensors: A Key to Successful Robot-Based Assembly

Shaheen Ahmad
Purdue University

Ahmad, Shaheen, "Sensors: A Key to Successful Robot-Based Assembly" (1987). *Department of Electrical and Computer Engineering Technical Reports.* Paper 554.
https://docs.lib.purdue.edu/ecetr/554

Sensors: A Key to Successful Robot-Based Assembly

Shaheen Ahmad

TR-EE 87-4
March 1987

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

SENSORS: A KEY TO SUCCESSFUL ROBOT-BASED ASSEMBLY

Shaheen Ahmad
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907
USA

INTRODUCTION

Computer controlled robots offer a number of significant advantages in manufacturing and assembly tasks. These include consistent product reliability and the ability to work in harsh environments. The programmable nature of robotic automation allows the possibility of applying them to a number of tasks. In particular, significant savings can be expected in batch production, if robots can be applied to produce numbers of products successfully without plant re-tooling. Unfortunately, despite considerable progress made in robot programming [Lozano-Perez 83] [Paul 81] [Ahmad 84] [Gruver *et al.* 84] [Bonner & Shin 82] and in sensing [Gonzalez & Safabakhsh 82] [Fu 82] [Hall *et al.* 82], [Goto *et al.* 80], [Hirzinger & Dietrich 86], [Harmon 84], kinematics and control strategies [Whitney 85] [Luh 83] [Lee 82], a number of problems still remain unsolved before en-mass applications take place. In fact, in current applications, the specialized tooling for manufacturing a particular product may make up as much as 80% of the production line cost. In such a production line the robot is often used only as a programmable parts transfer device.

Improving robots ability to sense and adapt to different products or environments so as to handle a larger variety of products without retooling is essential. It is just as important to be able to program them easily and quickly, without requiring the user to have a detailed understanding of complex robot programming languages and control schemes such as RCCL [Hayward & Paul 84], VAL-II [Shimano *et al.*, 84], AML [Taylor *et al.*, 83], SRIL-90 [Ahmad 84] or AL [Mujtaba & Goldman 79]. Currently there are a number of Computer Aided Design (CAD) packages available which simplify the robot programming problem. Such packages allow the automation system designer to simulate the assembly workcell which may consist of various machines and robots. The designer can then pick the motion sequences the robot has to execute in order to achieve the desired assembly task. This is done by viewing the motions on a graphical screen from different viewing angles to check for collisions and to ensure the relative positioning is correct, much the same way as it is done in on-line teach playback methods (see Figure 1). Off-line robot programming on CAD stations does not always lead to successful results due to two reasons:

(i) The robot mechanism is inherently inaccurate due to incorrect kinematic models programmed in their control system [Wu 83] [Hayati 83] [Ahmad 87] [Whitney *et al.* 84].

(ii) The assembly workcell model represented in the controller is not accurate. As a result parts and tools are not exactly located and their exact position may vary. This causes a predefined kinematic motion sequence program to fail, as it cannot deal with positional uncertainities.

Sensors to detect real-time errors in the part and tool positions are obviously required with tailored sensor-based motion strategies to ensure assembly accomplishment. In this chapter we deal with how sensors are used to successfully ensure assembly task accomplishment. We illustrate the use of various sensors by going through an actual assembly of an oil pump. Additionally we illustrate a number of motion strategies which have been developed to deal with assembly errors. Initially, we discuss a number of sensors found in typical robotic assembly systems in Section 1. In Section 2 we discuss how and when sensors are to be used during an assembly operation. Issues relating to sensing and robust assembly systems are discussed very briefly in Section 3. Section 4 details a sensor-based robot assembly to illustrate practical applications.

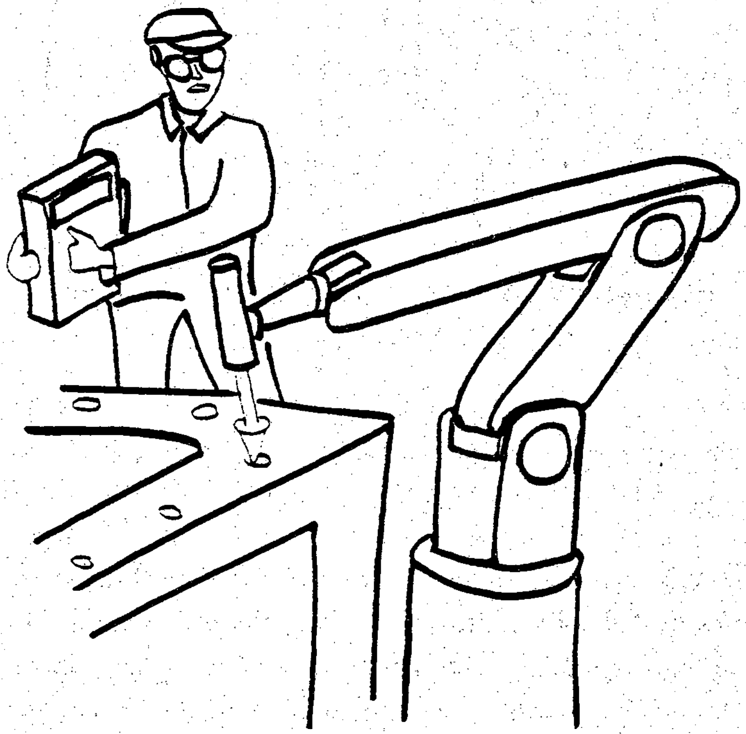


Figure 1: On-line Teach Playback Methods

Section 1: Types Of Sensors Utilized In Robotic Applications

There are fundamentally two types of sensors. There are those that rely on mechanical contact via the robot structure to the sensors or directly with the sensor and its environment, such as force and touch sensors. Other sensors do not require contact with the environment such non-contacting sensors would include vision, optical and ultra-sonic ranging sensors. Thermal sensors may also be of non-contacting type. The mode of use of these sensors is also different. For example, force or tactile data would not reveal useful information about an object until a contact is made. Similarly ranging or visual feature identification cannot be carried out if the view is obscured or is out of the ranging distance. These issues are further discussed in the Section 2. A brief description of some commonly found sensors and the physics of their operations is now presented.

Contacting Sensors

Contacting sensors may include a force sensing structure, a mechanical limit switch, a linear potentiometer, a linear voltage differential transformer (LVDT), a tactile sensor, etc. We however limit our discussion to a force sensing structure, an optical touch sensor array and a linear potentiometer. Figure 2 illustrates a force sensing structure which was developed by DFVLR of West Germany [Hirzinger & Dietrich 86].

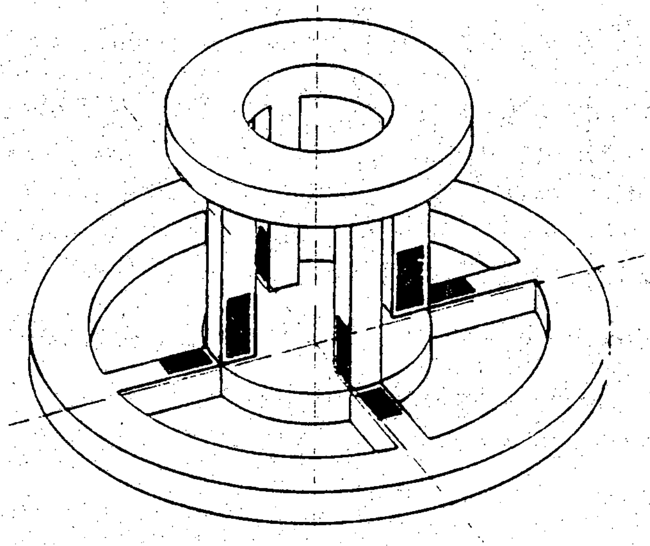


Figure 2: DFLVR Force Torque Sensor

When a force is applied at a point on the sensing structure, the elemental members of this structure will deform much like a spring, except the structure is designed to behave like a six dimensional spring. This deformation is small but can be measured by optical or electromechanical techniques, such as with strain gauges. The sensor may employ n-strain gauge elements with R^{nx1} output vector $\underset{\sim}{\epsilon} = (\epsilon_1,...,\epsilon_n)^t$. A force sensor which is deformed by a vector $\underset{\sim}{x} \in R^{6x1}$, would output a force $\underset{\sim}{f} \in R^{6x1}$. If [K] is R^{6x6} matrix representing the structural stiffness in the elastic region of operation, then:

$$\underset{\sim}{f} = [K]\ \underset{\sim}{x}$$

If [C] is a R^{6xn} matrix which transforms n-strain gauge sensor readings to deformation $\underset{\sim}{x}$, then:

$$\underset{\sim}{f} = [K]\ [C]\ \underset{\sim}{\epsilon}$$

assuming a linear relationship exists between the strain gauge deformation and the applied force. Most commercially available force sensing structures will process the signals from the strain gauge element reading to a force signal in a user defined coordinate frame. Force sensors employing capacitive measurement techniques have also been developed [Sinden & Boie 86].

A linear potentiometer may be used to measure displacement or a force (if combined with a compliant structure). A possible use is shown in Figure 3.

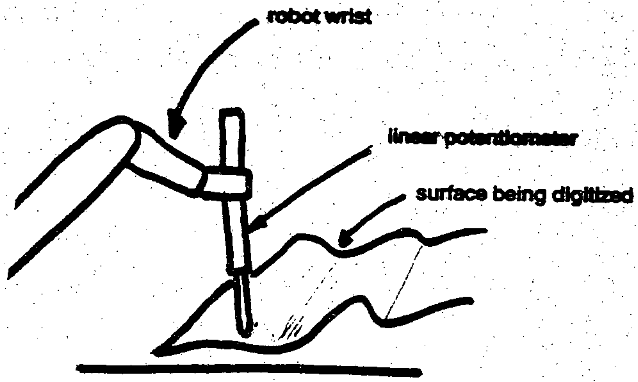


Figure 3

A use of a linear potentiometer to obtain surface shape. The method is simple and lengthy, but it works.

The linear displacement $\underset{\sim}{x}$ between the contact point and the pre-programmed path of the robot end effector is used to obtain the shape of the surface.

An optical touch sensor is shown in the Figure 4, it is reported by Bege in [Bege 84].

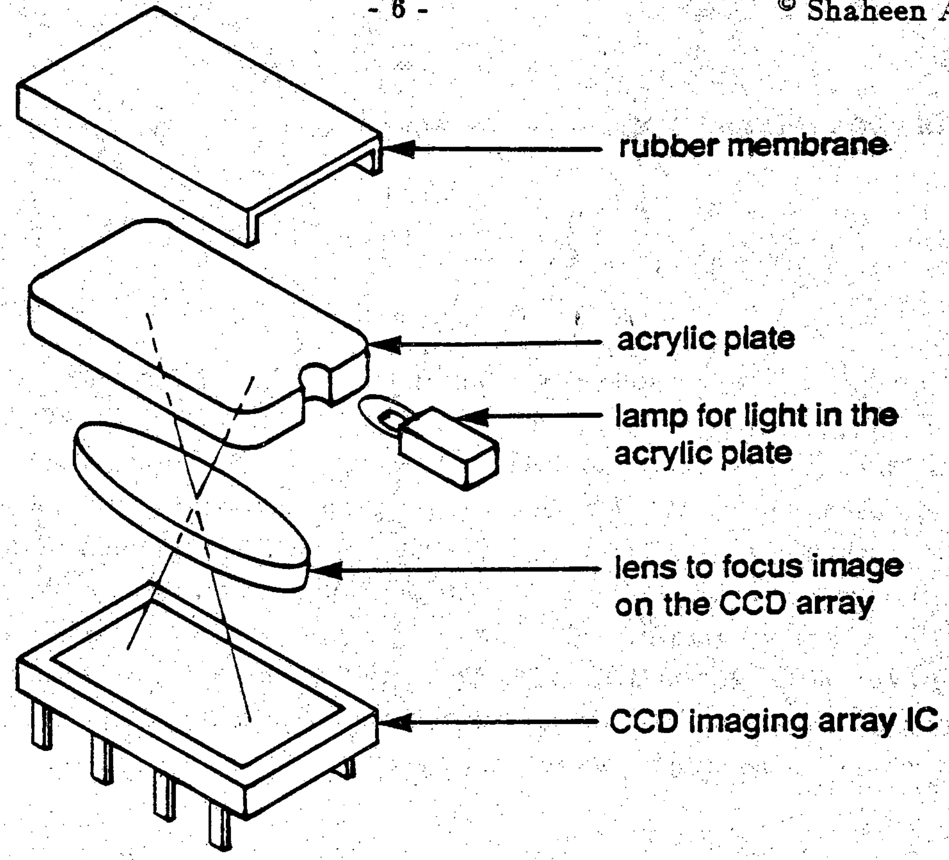


Figure 4: An Optical Tactile Sensor

A number of other tactile array sensors have been constructed based on various technologies [Raibert & Tanner 82] [Hillis 82]. The underlying physics on which this device functions are as follows: when there are no objects placed on the elastic membrane total internal refraction is experienced by the light waves in the acrylic plate, hence the image formed by the lens on the CCD array is black. Once an object is placed on the elastic membrane, the light in the acrylic plate is diffracted by the higher refractive index of the touching membrane. The diffracted rays travel through the acrylic plate and are focused by a lens onto the CCD array. The advantage of this sensing scheme is that the actual contact between the sensor surface and the environment is minimal. Also, existing image processing techniques may now be utilized to analyze the image formed on the CCD array.

Noncontacting Sensors

Noncontacting sensors include optical, acoustic ranging, vision and magnetic sensors. Vision sensing is explained in greater detail in other chapters of this text. In this section we briefly explain acoustic ranging.

An ultrasonic ranging system relies on the fact that sound at a constant temperature and pressure will travel at a constant speed. Also, if an ultrasonic wavefront strikes an obstacle, depending on the acoustic impedance of the object's surface, part of the wave energy will be retransmitted in the form of an echo. The time between transmission of the ultrasonic pulse and the time for the echo to return to a receiver which is usually located close to the transmitter, can be used to calculate the distance to the object (see Figure 5).

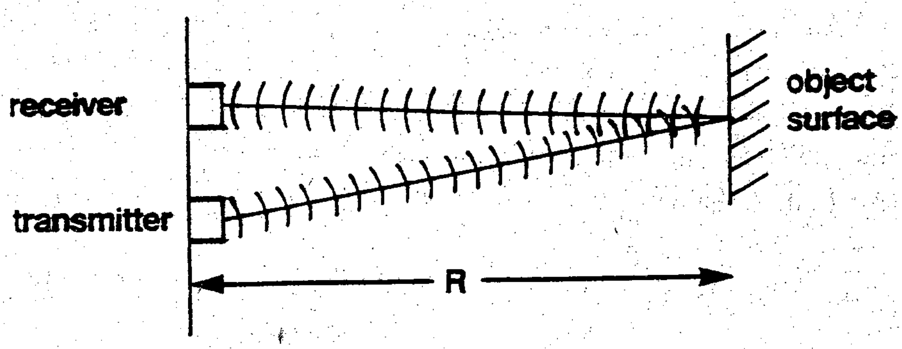


Figure 5
An ultrasonic ranging system

If V_s is the velocity of sound, and τ is the period of time between pulse transmission and reception, then range R is given as:

$$R \cong \frac{1}{2} V_s \tau$$

A practical ultrasonic ranging system would have to employ various pulsing frequencies to deal with noise and temperature or pressure variations. For more detailed description on an ultrasonic system refer to [Polaroid 82]. Methods of determining surface information specifically from ultrasonic data is discussed in Brown [Brown 85] [Brown 86].

Section 2: How And When To Use Sensors In Robot Assembly Programs

As mentioned in the introduction, kinematically programmed robot programs can fail if parts become slightly displaced from their preprogrammed paths, or if other uncertainties arise.

Consider an assembly, such as the oil pump shown in Figure 11, which consists of two gears enclosed in the oil pump casing. This task consists of manipulating the gears from their initial position and inserting them into their respective bearing housings. In order for this assembly to be successful, it is necessary that the trajectory of the gear shaft follow a predefined path relative to the bearing housing. This is not necessarily a unique sequence. If there is a deviation from the set of possible paths the assembly sequence may fail. It is therefore necessary to utilize sensing to indirectly and directly verify the relative motion sequence. Indirect operations would include obtaining the correct position and orientation of the gear shaft and the oil pump casing prior to grasping the parts, obtaining the position of the pump casing before the gear insertion. Direct measurements would include measuring the relative orientation of the gear shaft with respect to the bearing housing (located on the pump casing) during the insertion process. In these instances the sensing operation is being used to verify and calibrate manipulation operations.

However, only in certain instances will the sensor give useful information. An example of this is that vision will only provide useful information if the view is not obscured and the measurable features are in range. Similarly, the force sensors can only give useful information if the robot tools are in contact with the environment. Likewise, the tactile sensor will provide useable data if the part is in contact with the touch sensor. The sensing instance has to be in a correct geometric context in order for the sensing operation to be useful. This implies the viewpoint has to be appropriately selected, as well as the sensing instance has to be as close as possible in time to the related motion sequence.

Which Sensor to Use

In a predefined robot assembly system the questions are relatively easy to answer. A robot motion sequence can be broken down into transfer operations, contact operations such as grasping, insertions and other mating operations.

The type of sensor to utilize for an assembly sequence verification can be approximately determined from the type of manipulation operation. An example of this is during the termination phase of an insertion process, the force sensor will monitor a sharp change in f_z[*] as the peg makes contact with the bottom of the hole. It would therefore be appropriate to utilize a force sensor to monitor the termination phase of the insertion. However, visual information may also be used to verify the insertion if the peg is visible. The position of the joints may also be used to verify the insertion.

However, the rate at which visual information must be processed may not be economically feasible. Similarly, the backlash and compliance present in the robot drive train may not allow accurate measurement of the peg's position if the joint sensors are mounted on the motors instead of on the actual robot joints.

---

[*] f_z is the force peg experiences along the z direction, the direction along which the axis of the peg and the axis of the hole coincides at the end of the insertion process.

Utilizing Multiple Sensors for Updating Object Position[*] Information

A number of sensors may be used to obtain information about an object's position. Methods of combining data from different sensors is currently under investigation [Henderson & Shilcrat 84] [Durrant-Whyte 86] [Stevenson 86]. Figure 6 below illustrates this concept:

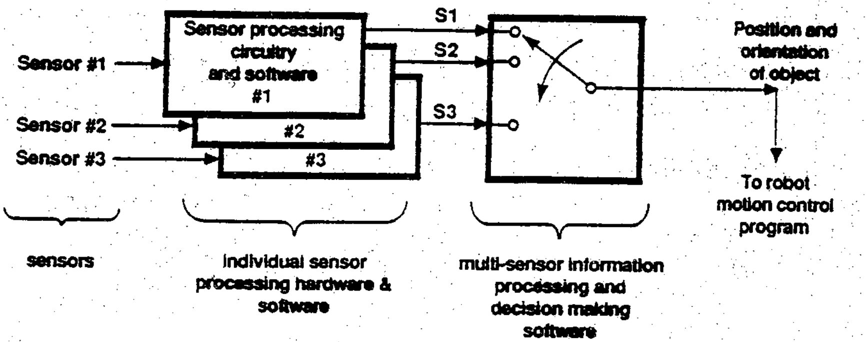


Figure 6: Method of Multiple Sensor Utilization.

In such a system a number of questions must be answered by the mechanism which combines the sensory data. These include:

(i) the correctness of the geometric context:
  (a) view point, is the data originating from the sensor viewing the desired object surface, edge or other useable features.
  (b) sensing instance, will the data from this sensing process be useful for immediate manipulation sequence?
(ii) statistics of the sensors being used. How error prone is the extracted data from a particular sensor?

A number of researchers approach these problem from a statistical [Durrant-Whyte 86] or a numerical standpoint [Stevenson 86]. A hierarchial conceptual servo has also been discussed as a method of interfacing sensors to a manipulation sequence [Kent & Albus 84], such a system must address the above fundamental questions directly or indirectly in its sensor-fusion process. Grimson [Grimson 86] developed an algorithm to recognize and locate an object utilizing minimal set of sensory data. He also considers the determination of the optimal viewpoint, these viewpoints will require the minimum number of sensed points to uniquely identify the objects position and orientation.

---

[*] Position implies position and orientation.

Section 3: Robust Sensor-Based Assembly Systems

The objective of research in the area of robust sensor-based robot assembly systems is to develop a formal theory for programming and planning of sensor-based robot assembly systems. These systems must be able to guarantee robustness even though assembly parts position, sensing actions, and robot operation may have uncertainty associated with them. Additionally, events which disturb the preplanned program can be expected. These include dropping of a part during a transfer, and unexpected collision with the environment or with another manipulator. Other disturbances would include machine failures. A system which is capable of dealing with all aspects of assembly planning with uncertainty and is able to guarantee real-time operations, architecturally has three components (see Figure 7).

(i) Off-line planning and code generation mechanism,

(ii) Intelligent machines which execute the code that is generated,

(iii) On-line system monitor which ensures the assembly system performs the scheduled tasks despite errors that may occur.

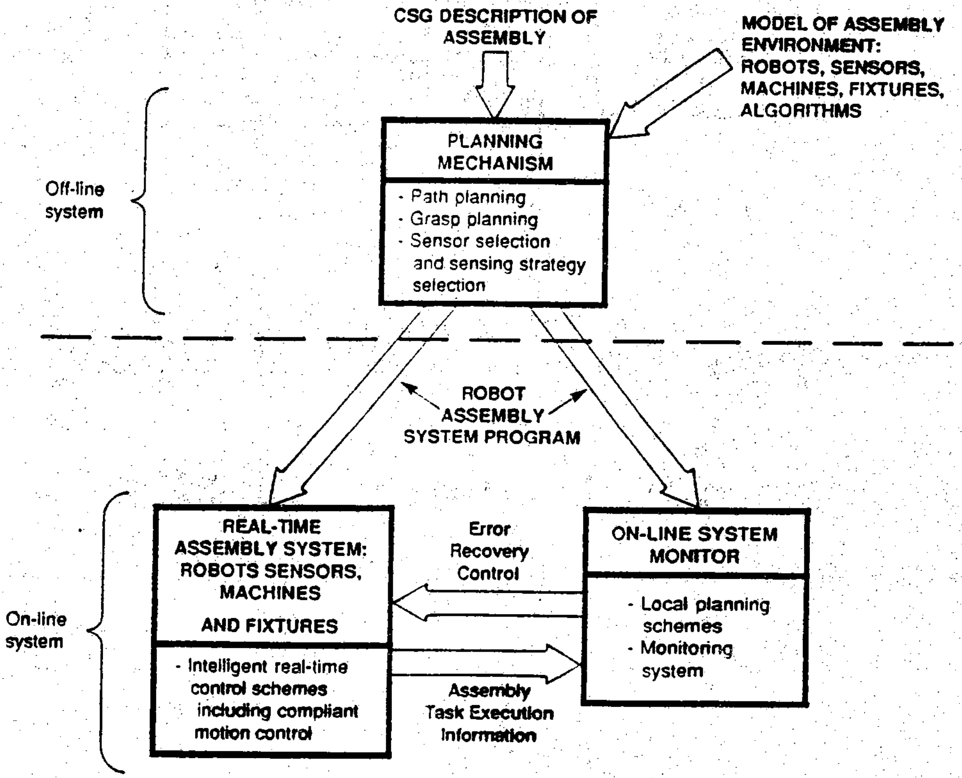


Figure 7: Assembly Planning, Execution, Monitoring System

The *planning mechanism* deals with the path planning of the manipulator and that of the machines in the assembly environment, given the solid geometric description of the object it is to assemble. It also decides the sensing operations which are essential to add robustness to the assembly operations. Motion strategies which deal with uncertainty are also selected by the planner. The set of grasping positions for the assembly components must also be calculated by the grasp planner.

The *assembly system,* which consists of various sensors, robots and other machines, must be able to respond and synchronize to the primitive actions specified by the planner.

The *on-line system monitor* receives information on the assembly from the planning mechanism, and the execution information from the assembly system components. From the two sets of information, it can decide whether an assembly system error has occurred. If so, it can take over and provide error-recovery.

There has been considerable research effort in the area of robot path planning, grasping, error recovery and assembly planning. We only briefly discuss previous effort in the area assembly planning systems.

Past Research In Assembly Planning:

Lozano-Pérez [Lozano-Pérez 76], Taylor [Taylor 76], Brooks [Brooks 82], have considered assembly planning systems. Lozano-Pérez [Lozano-Pérez 76] considers an assembly planning process to be a three stage effort. In the first stage, a general plan interms of class of operations to be performed is developed. For example, object A is to be placed on object B. In the second phase, the grasp planning and robot path planning is carried out. In the third stage sensing is incorporated to deal with uncertainity.

Taylor [Taylor 76] considered each operation to possess a certain set of preconditions and to achieve a certain set of postconditions. The task of the planner is then to satisfy these conditions subjects to the constraints imposed by the geometry of the task.

Brooks [Brooks 82] developed a symbolic error analysis package. This package is able to propagate errors through a motion sequence and is used to check generated plans. These plans are checked interms of whether they satisfy the required error constraints. If these constraints are not satisfied plans can be altered by introducing new operations so as to guarantee satisfaction.

Section 4: A Practical Implementation of A Sensor-Based Robot Assembly System

In order to gain an understanding of the task involved in the design of a practical sensor-based robot assembly program utilizing currently available industrial equipment, we document an experiment performed in our robot laboratory class. This experiment is performed by first year graduate students and seniors in the School of Electrical Engineering at Purdue University. This experiment is a part of a laboratory class entitled "Real-time Robot Control Laboratory". The sensor-based robot assembly experiment is used as a base to educate research students in the area of automated sensor-based robot assembly systems. This experiment involves the assembly of an oil pump, the assembly sequence is manually generated and programmed. It is interesting to note that the oil pump assembly is not difficult to perform manually. It is, however, not possible to perform this assembly reliably (everytime) with kinematic programming techniques that are quite commonly used in pick and place industrial operations. The reasons for this have been stated earlier and are reiterated here, they are:

(i) parts have manufacturing tolerance [Requicha & Chan 86],

(ii) parts are not exactly located due to inaccuracy in the assembly set-up or inaccurate calibration of the assembly environments, and

(iii) robots are not accurate in absolute programmed mode [Ahmad 87] [Whitney *et al.* 84], they introduce errors into the assembly operations.

In order to make assembly systems robust, sensing must be used before and after an assembly action to verify its completion. At this point, a general theory to plan assembly operations which guarantees robustness does not exist. The patterns of information which are received at each stage of assembly must be manually generated, verified and programmed, even though these information patterns are derived from geometric properties of the assembly components.

In order to cope with various assembly errors, a number of practical calibration techniques and sensor-based motion strategies have been developed [Inoue 74], [Wills & Grossman 74], [Goto *et al.* 80] [Mason & Salisbury 85] [Brady *et al.* 82]. Some of these practical strategies are demonstrated in the below oil pump assembly experiment. This experiment is described in three subsections; the first subsection, Section 4.1, describes the equipment and experimental setup. The second subsection, Section 4.2, details the sensing and sensor based motion strategies employed to make the assembly robust. The third subsection, Section 4.3, pictorially depicts the assembly sequence.

Section 4.1: Equipment and Assembly Description

The main purpose of this experiment was to examine the practical use of different sensors in a complex assembly task and to expose various problems involved in a real-time robot assembly. The task is to assemble an oil pump independent of exact location and orientation of parts by using a sensor guided manipulator. The apparatus which was available for the implementation was an IBM RS-1 robot and GE Optomation vision system.

Description of Experimental Apparatus:

Robot and Imbedded Sensors: The RS-1 robot is a six degree-of-freedom Cartesian robot. This robot has three linear joints, three rotary jointed wrist and a gripper, see Figure 8.

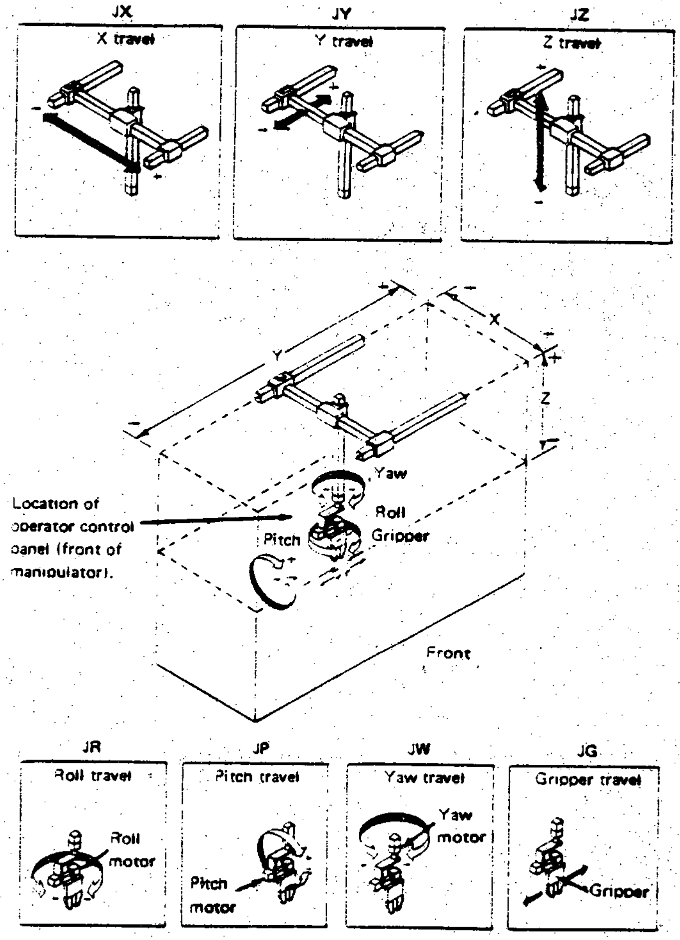


Figure 8. Robot Joints [IBM 83]

The robot has a six strain gauge force sensor built into the gripper which detects forces at the tips, sides, and perpendicular to the gripper surfaces. It is also equipped with a part detector which consists of an infra-red light emitting diode (LED) and a phototransistor located in the gripper. Once the part is in the grasp position between the two fingers of the gripper, the infra-red beam is broken, indicating the presence of a part. These sensors are shown in Figure 9. The RS-1 robot is programmed via AML (A Manufacturing Language) language to control the manipulator [Taylor *et al.* 83].

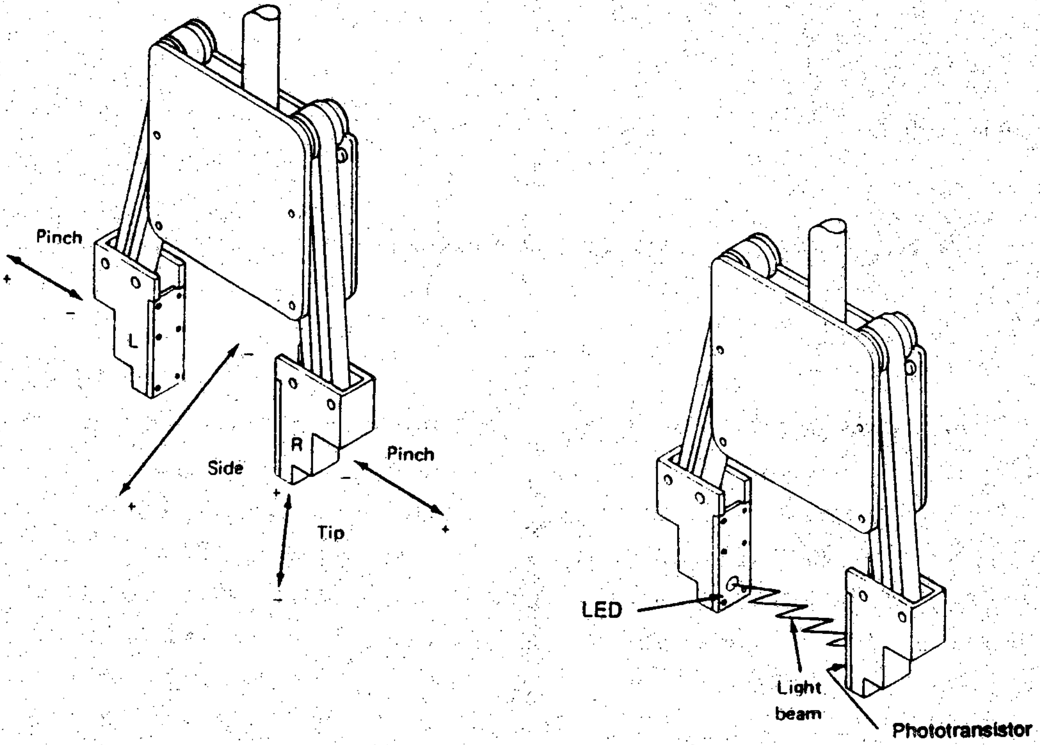


Figure 9. Strain Gauges and LED Part Detector [IBM 83]

Binary Image Processing System:

A General Electric Optomation binary image processor is also used in this assembly. The binary image processor can only provide information about the silhouette of an object. A binary image is defined as a binary function which takes a value of zero or one over the entire image plane. The image of an object in the field of view is converted to digital video data. This is thresholded to a binary representation before feature based recognition is carried out. Up to four solid state CCD cameras can be simultaneously processed by the Optomation system. The CCD cameras which are used have a 256 x 256 matrix of photosensitive pixels. The camera used in this experiment is mounted on the robot and it moves with the arm in a horizontal plane to monitor the assembly workspace.

The Optomation system has a relatively fast processing time, but it is incapable of distinguishing overlaps, shadows and images with nearly the same reflectivity as the background. It is necessary that images which are presented to the Optomation have a high object-to-background contrast. It can only provide planar information about an object. VPL (Visual Programming Language) language was used for setting up the image processing macro's which the system executes.

The Optomation system is able to store up to 100 objects in its database by using a built-in function, namely "QTY.ITEMS". The Optomation system is able to compute image features such as the area and perimeter of an item in the field of view by using built-in functions such as AREA(i) and PERIMETER(i), where "i" is the object index. The X and Y coordinates of the center of mass of an image in the camera coordinates can also be determined by built-in functions such as "CENTROID.X(i)", "CENTROID.Y(i)".

Integration of the Automation Devices:

The vision system is set up to communicate with the RS-1 controller through a parallel port. This allows the RS-1 controller to obtain data processed by the vision system about the objects in the workspace, and then to generate the manipulator movement accordingly. The Optomation is also connected to VAX 11/780 UNIX machine through a serial port; this connection is only used to back up software. The overall hardware configuration is summarized in Figure 10.

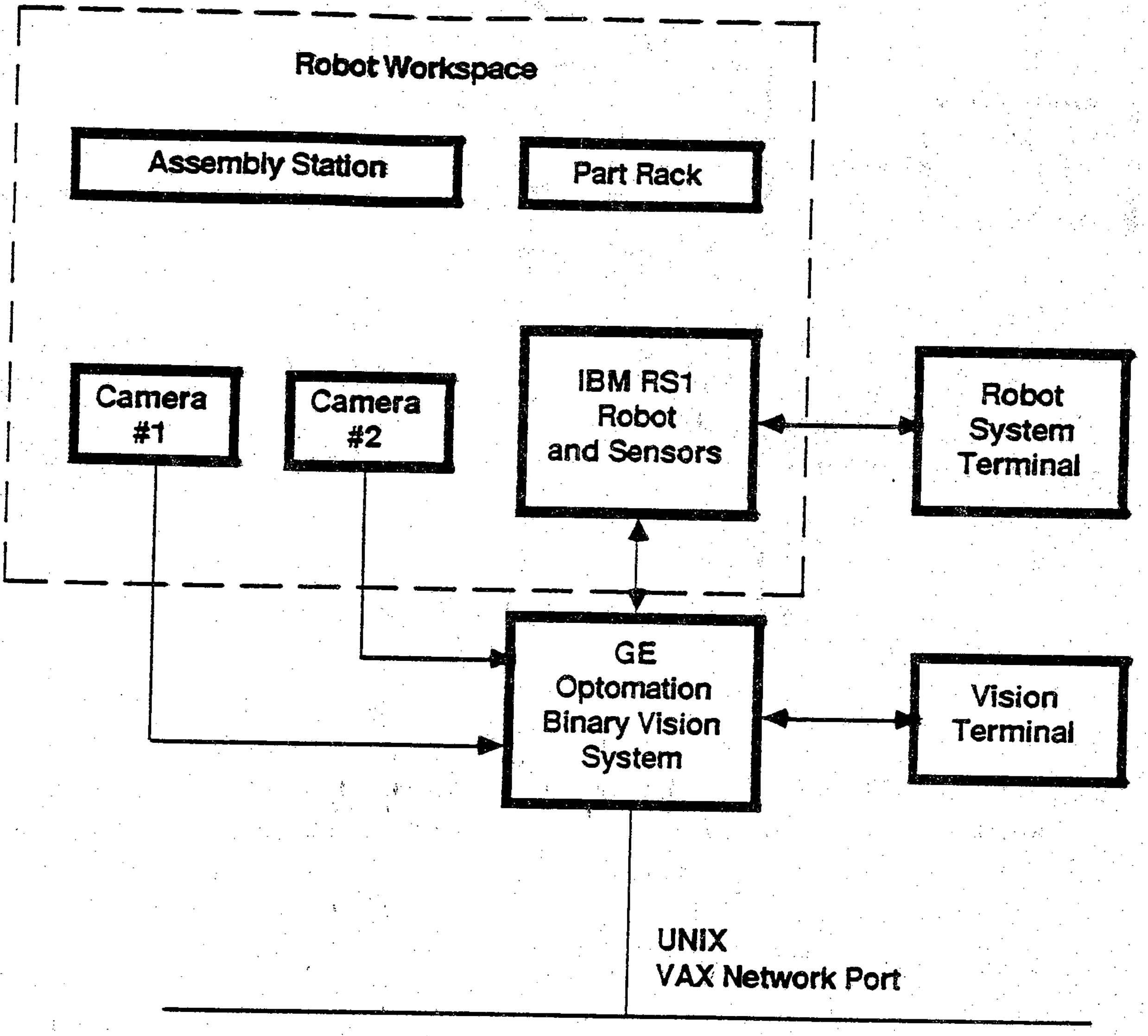


Figure 10. Work Station Configuration

Assembly Components:

Figure 11 shows the eight components of the oil pump assembly. The pump casing has two bearing housings which have to be mated with the gear shafts. The pump casing lid is assembled such that the gear shafts are inserted into the bearings located on the lid. Four bolts are used to fix the lid to the casing; these bolts and bolt holes are symmetrically located around the oil pump casing.

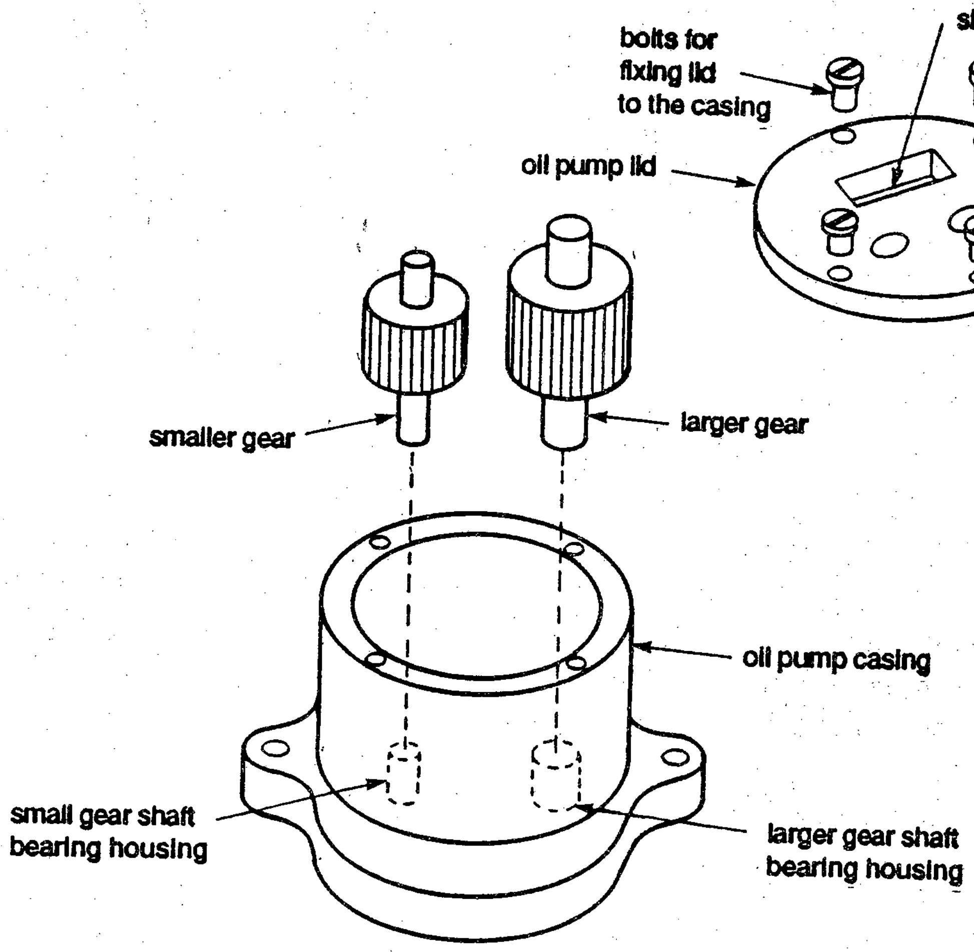
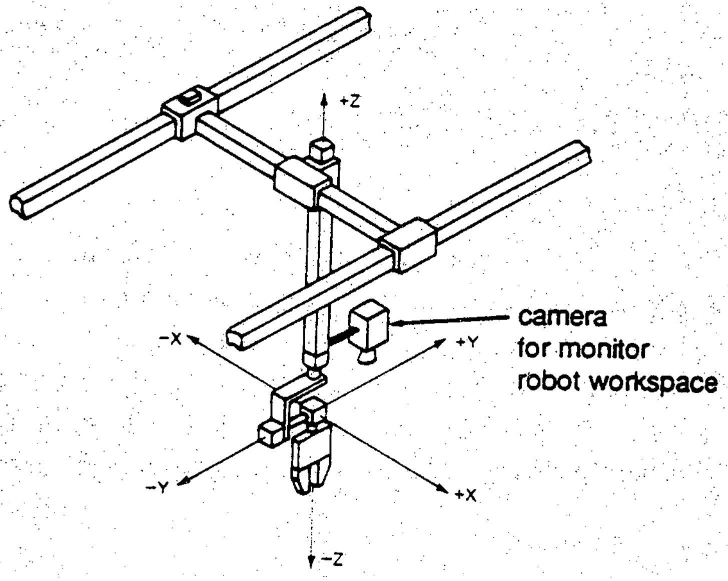


Figure 11. Oil Pump Components

Layout of the Assembly Workstation

The pump casing is securely located by a fixture, the pump casings lid, the gears and bolts are randomly located on a light box. Since the image processor is binary, the lighting is an important factor in accurately locating the part features. Back lighting created by the light box provides clear images and eliminates possible shadows. This lighting strategy improves the accuracy of visual sensing and subsequently enhances the manipulator's precision.

Coordinate Frames : The height (Z-coordinate) of the table, hence, the Z-axis location of all the assembly components, are known. To locate a feature point such as the centroids of the parts or holes, the processor finds the X- and Y-coordinates of the feature points in the camera's coordinate frame. These positions are sent to the robot controller where they are converted into the world coordinates. The world coordinate frame, which is also called box frame, is parallel with the robot wrist coordinates when the wrist roll, pitch and yaw rotational angles have values of zero as shown in Figure 12.



The box frame
The joints JX, JY, JZ, JP and JW have the value 0.0.
The joint JR is −45.0.

Figure 12. World Frame [IBM 83] and Camera Location

The camera is mounted on the robot (see Figure 12), its coordinate frame is always at a fixed offset in the X and Y direction from the robot's X and Y cartesian joints. Therefore, if the position of the X and Y joints of the robot is known in the world frame, the relative position of the camera in the world frame can be determined by one physical measurement. Once this relationship is found, the conversion of coordinates from the camera frame to the box frame is easily carried out by a scaling and a transformation matrix.

Assembly Sequence:

The assembly sequence can be outlined in three general stages. First, the gears are located and then placed into the gear housing. The lid is then mated with the two gear shafts already in their respective bearing housings. Finally, the four bolts are located and placed into the four holes around the pump casing. This sequence is summarized in Figure 13.

Sequence	Step 1	Step 2	Step 3
Stage I	Determine gear locations, grasp	Gross transfer motion	Insert gears into pump housing
Stage II	Determine lid location and orientation, grasp	Gross transfer motion	Place lid on pump housing
Stage III	Determine bolt location, grasp	Gross transfer motion	Insert bolts into pump housing

Figure 13. Assembly Sequence

Section 4.2 : Practical Sensor-Based Motion Strategies

In this section we discuss a number of practical sensor-based motion strategies which were employed in the oil pump assembly, they are:

Move ( ) until condition ( )

This motion strategy relies on an event to occur in order to terminate a motion. The first parenthesis will usually contain a position goal which may be predefined, the second may contain a sensor or other conditional events. This strategy has often been called a guarded move [Will & Grossman 75] [Brady *et al.* 82]. An example use of this strategy is during placement when there is uncertainty about the height of the platform.

If an object's position is not exactly known and it must be grasped, this conditional movement can be used to accurately center the gripper. Assume that the maximum width of the object is smaller than the maximum opening of the gripper. Assume also, that an LED beam sensor is available on the gripper and the height of the object and the table on which the object is placed on is known. A conditional move in X-direction, with the gripper wide open until the LED sensor beam is broken, allows the determination of a face of the box that bounds the object. A second conditional move in the Y-direction allows the determination of the second face of the bounding box. Knowing the faces of the bounding box, and if the orientation of the part can be determined from this information, an accurate centered grasp can be found.

Servo ( ) while condition ( )

This is a servo process. The first parenthesis will contain an expression which will be be updated to represent a position or a force goal, while the second will represent a logical or a sensor directed condition. Examples of this strategy include force, position servoing, compliant motions, visual servoing, and other sensor-directed servo processes. Consider that an object is to be visually tracked, the gripper frame is to be aligned with the object frame which is obtained from the vision processor. This can be implemented by the above sensor-based motion construct.

Insertions and Spiral Searching

A number of strategies are required to correctly perform an insertion.

Approach from a Fixed Direction. This ensures that the direction in which the gear shaft (the peg) is to be moved is always the same [Goto *et al.* 80] [Inoue 74].

Compliant Motion. In order to complete the insertion, the peg is required to be placed over the hole, planar XY motions are required to locate it, and a force along the Z direction is required to maintain contact with the xy plane. Once the peg is located over the hole, forces in the XY directions must be minimized to prevent jamming of the peg. In this motion sequence, the force servoing is specified in the orthogonal directions to those in which position servoing is desired [Mason 81].

Spiral Searching. During an insertion, if the hole is not located at the approach phase, a spiral search is initiated to locate it. This search may be decomposed into incremental movements in the X- and Y-directions. Each of these increments must be smaller than the diameter of the hole.

Conditions for Detecting the Hole and the Termination of the Insertion Phase. If the hole is detected, there will be a momentary change in the force along the Z-direction. This condition must be monitored and used to initiate the insertion phase. At the completion of the insertion phase, the force f_z will again return to the prespecified value and the position of the peg should be at the bottom of the hole. Once this occurs, the insertion phase is complete [Goto *et al.* 80].

Visual Search Strategy:

Search : This strategy is used to locate the objects in the workspace. This routine is an example of "Move () until condition ()" movement which is executed while the vision system is constantly trying to locate the objects. Once a desired object is found in the field of vision, the motion is terminated.

The visual search algorithm can be translated to VPL language as follows:

```
ITEMS = 0
DO WHILE ITEMS = 0
   ...
  ITEMS = QTY.ITEMS
END WHILE
```

When a part is detected, the variable QTY.ITEMS becomes non-zero at that stage the while loop is exited. A message is then sent to the robot indicating that the part is found. Upon receiving that message, the search motion is terminated.

Part Recognition: A challenging problem in robot vision tasks is to identify each component with the minimum number of features in a short period of time. Fortunately in this assembly with adequate lighting simple binary features can be used to distinguish between objects. Features of each component such as area, perimeters, and number of holes are used to identify the presence of an object in the workspace. For example, screws have smaller area and circumference compared to gears, the lid has several holes which distinguish it from all others.

Recognition algorithm is implemented in vision software by adding conditional statements inside the while loop in the search routine discussed above; for example, the following modification may be used to distinguish the bolts from the gears based on the area of the parts:

```
ITEMS = 0
DO WHILE ITEMS = 0
   ...
  TAKE.PIX(1)
  IF QTY.ITEMS<>0 THEN
    FOR I=1 TO QTY.ITEMS DO
      IF AREA(I) > (SCREW_AREA_MIN) AND AREA(I) < (SCREW_AREA_MAX) THEN ITEMS=QTY.ITEMS
    NEXT I
  END IF
END WHILE
```

The first if statement checks the presence of parts, the second if statement uses the feature of the part, namely the area, to identify the screw. The "for loop" is used to index through all the items in the field of view.

Calculating the Orientation of An Oil Pump Lid from a 2D-Binary Image:

In order to grasp a geometrically known object correctly, the position and the orientation of the object has to be precisely determined. Symmetric 2D objects such as the gears and bolts can be grasped once the binary image centroid is computed. The oil pump lid is circular and has four bolt holes symmetrically located about its circumference. Furthermore, it has two holes to locate the gearshaft and a slot is also located on the lid (See Figure 14).

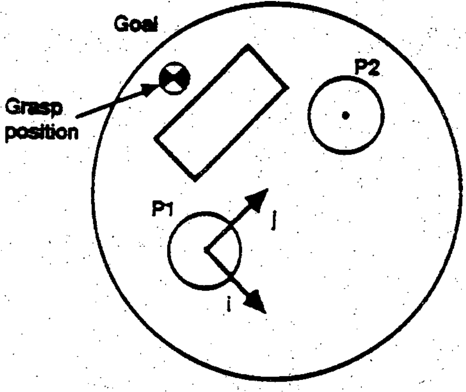


Figure 14: Oil Pump Lid Features

In order to determine the exact orientation of the lid, the following procedure is used:

(i) The position of the two holes to locate gear shafts are determined as $\underset{\sim}{P}_1$ and $\underset{\sim}{P}_2$.

(ii) A coordinate frame is located at P_1, such that the unit vector $\hat{\underset{\sim}{j}}$ is given as:

$$\hat{\underset{\sim}{j}} = \frac{\underset{\sim}{P}_2 - \underset{\sim}{P}_1}{|\underset{\sim}{P}_2 - \underset{\sim}{P}_1|}$$

The unit vector $\hat{\underset{\sim}{i}}$ is determined as:

$$\hat{\underset{\sim}{i}} = \hat{\underset{\sim}{j}} \times \hat{\underset{\sim}{k}}$$

$$\text{where } \hat{\underset{\sim}{k}} = (0,0,1)^t$$

(iii) If the position of the slot in the established coordinate is in first quadrant, then the grasp position GOAL in this frame is given as:

$$\text{GOAL} = (x_g, y_g)^t$$

Otherwise,

$$\text{GOAL} = (-x_g, y_g)^t$$

This is shown in Figure 15.

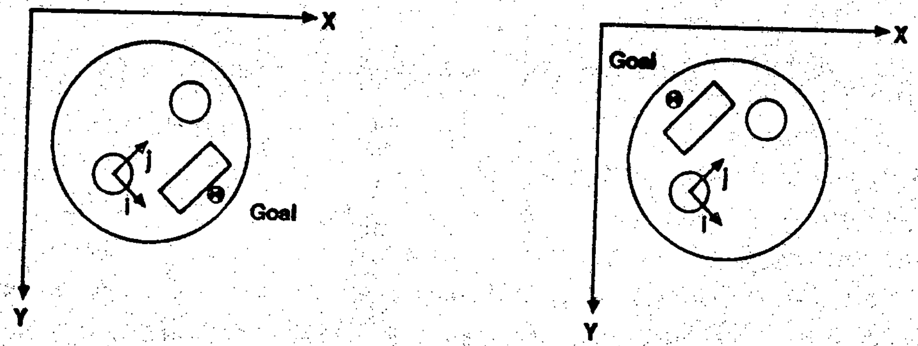


Figure 15
Two Orientations of the Oil Pump Lid

This information is needed to position and orientate the gripper accurately prior to picking up the lid.

Section 4.3: The Assembly Program

First Stage: Gear Assembly

Figure 16 illustrates the sequence of motions the robot must execute in order to assemble the gear box. The robot initiates the assembly by sending a message to the Optomation to locate the gears. The approximate position of the light table on which the gears are placed on is known. A visual search is performed to exactly locate a gear, the vision system constantly takes pictures and, based on the visual search strategies discussed in the above, it stops when a gear is found. The robot is then signaled to grasp the gear, which it approaches to a known height with a guarded move. The LED and force sensors are used to trigger this conditional move. When the LED, or the force sensor is tripped, the robot will grasp the gear shaft and commence the departure from the grasp position.

The robot then moves the gear above a solid surface of a known height and checks the height of the gear in order to identify which gear has been grasped. Because of the depth of the pump housing, the short gear cannot be inserted if the long one is already in place; therefore, the short gear has to be assembled first. If the robot initially picks the longer gear, it will put it in a known position and pick up the short one. The robot then approaches the pump housing. Before this transfer movement is executed, it will rotate the part in order to minimize the part slippage during gross motion.

To insert the short gear, the robot orients the gripper to a certain pitch angle and lowers the gear while the force sensors are monitored. Once the gear hits the bottom of the pump housing, the movement is stopped, and the gear is inserted into the bearing housing. The second gear is inserted vertically, and if the two gears do not mesh, the robot twists the longer gear until they mesh. The robot verifies the gears are in place by lowering the gripper and monitoring the height at which the force sensor triggers or the LED sensor beam is broken.

Second Stage: Lid Assembly

Figure 17 depicts the motion sequence for the lid assembly. The sequence is initiated by sending a message to Optomation indicating that the lid is to be located. Once the lid is located, the robot tries to grasp it. If it is unsuccessful a spiral search is executed to locate the grasp position accurately.

The robot then approaches the pump housing, and aligns the lid's coordinate frame with that of the pump housing (this coordinate frame is determined by visual measurements). It tilts the gripper to a specified pitch angle and lowers the lid until contact is made utilizing a force guarded move. The robot drags the lid along the Y-coordinate of the pump, the robot then opens the gripper slightly and pulls along X-axis of gripper until desired force is thresholded to ensure both gear shafts are located in their bearing housings. The gripper is then raised, and lowered onto the lid, to ensure the lid is mated with the pump housing and the gear shafts are inserted into the lids bearing housings.

Third Stage: Securing the Oil Pump Lid with Bolts

Figure 18 illustrates the sequence of motions involved in securing the oil pump lid with the four bolts. This sequence is initiated by the robot sending a message to the Optomation that the third stage in the assembly process is started. Optomation then locates the bolts in its visual area. The robot approaches the bolts with a guarded move during which the LED and force sensors are monitored. The bolts are placed symmetrically on the pump lid, which is located on top of the pump casing. The pump coordinate frame was previously determined in the second assembly stage. Once the bolts are located in the bolt holes, the robot picks up an electric screwdriver and tightens each of the bolts.

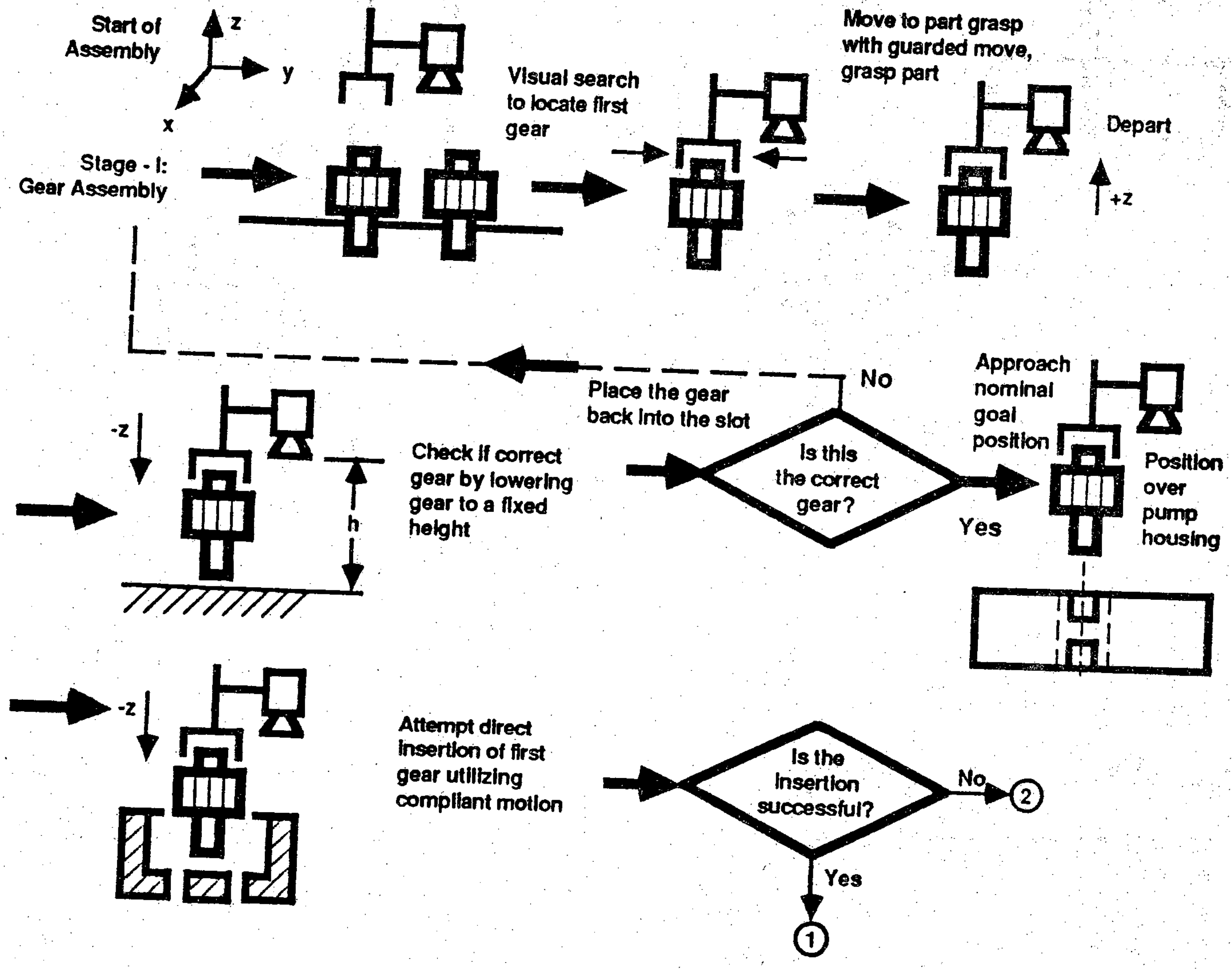


Figure 16(a): Gear Assembly

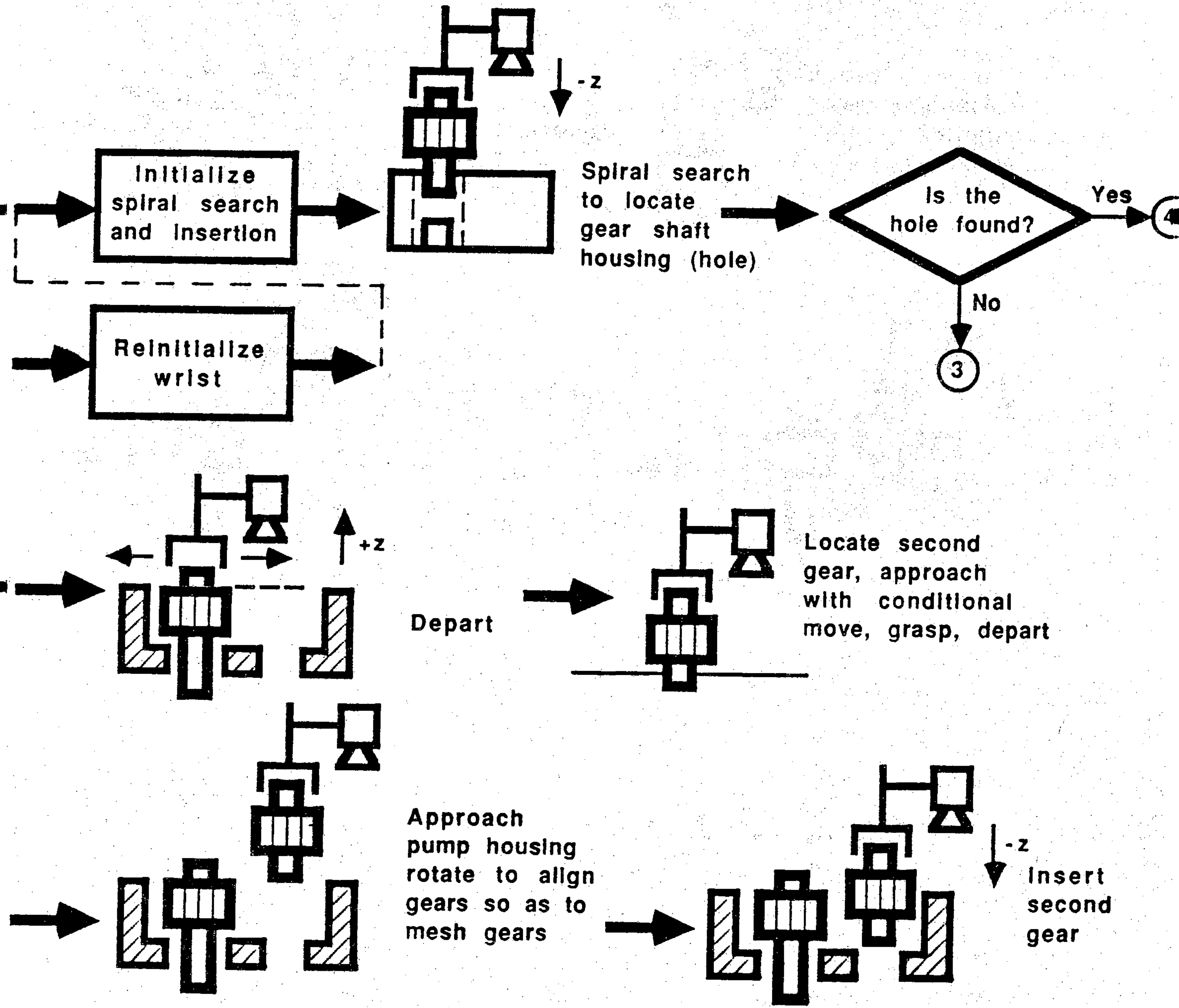


Figure 16(b): Gear Assembly

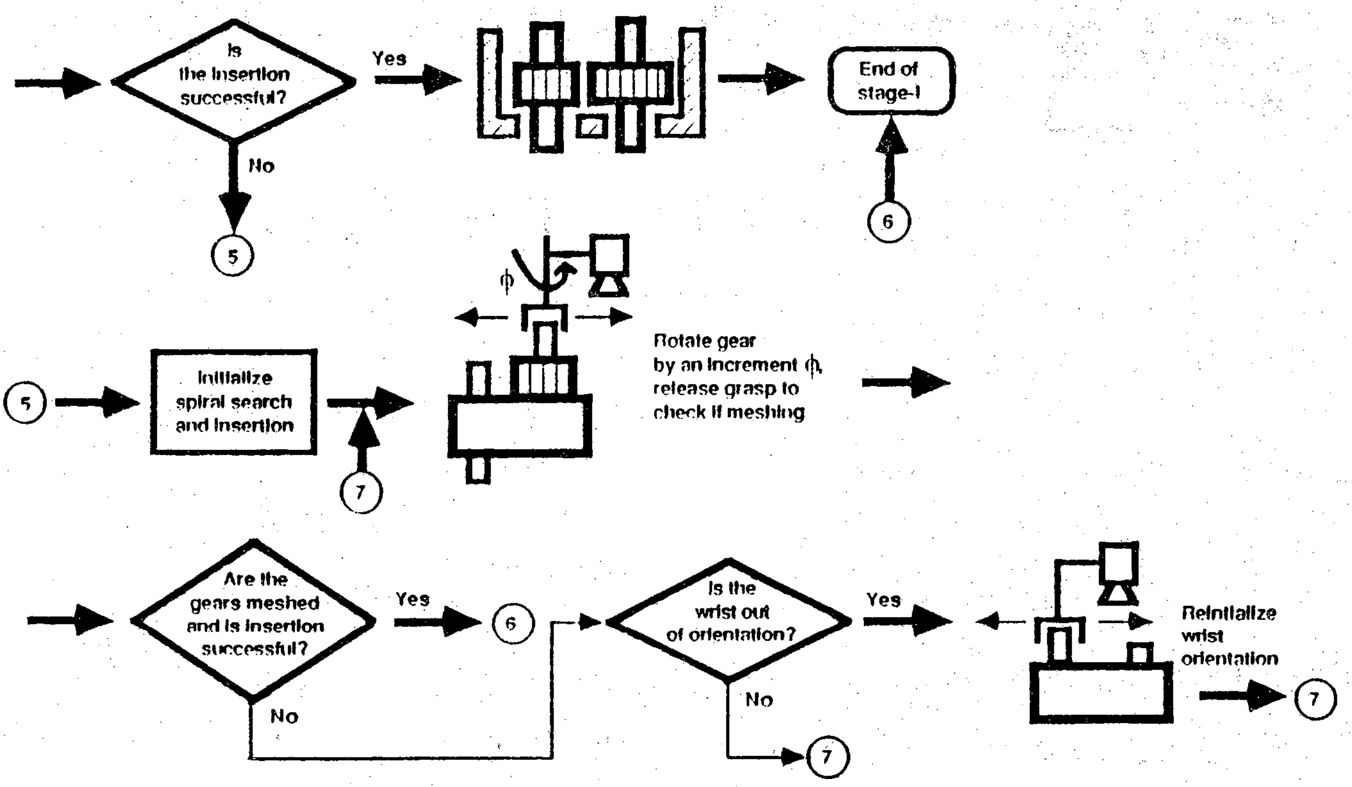


Figure 16(c): Gear Assembly

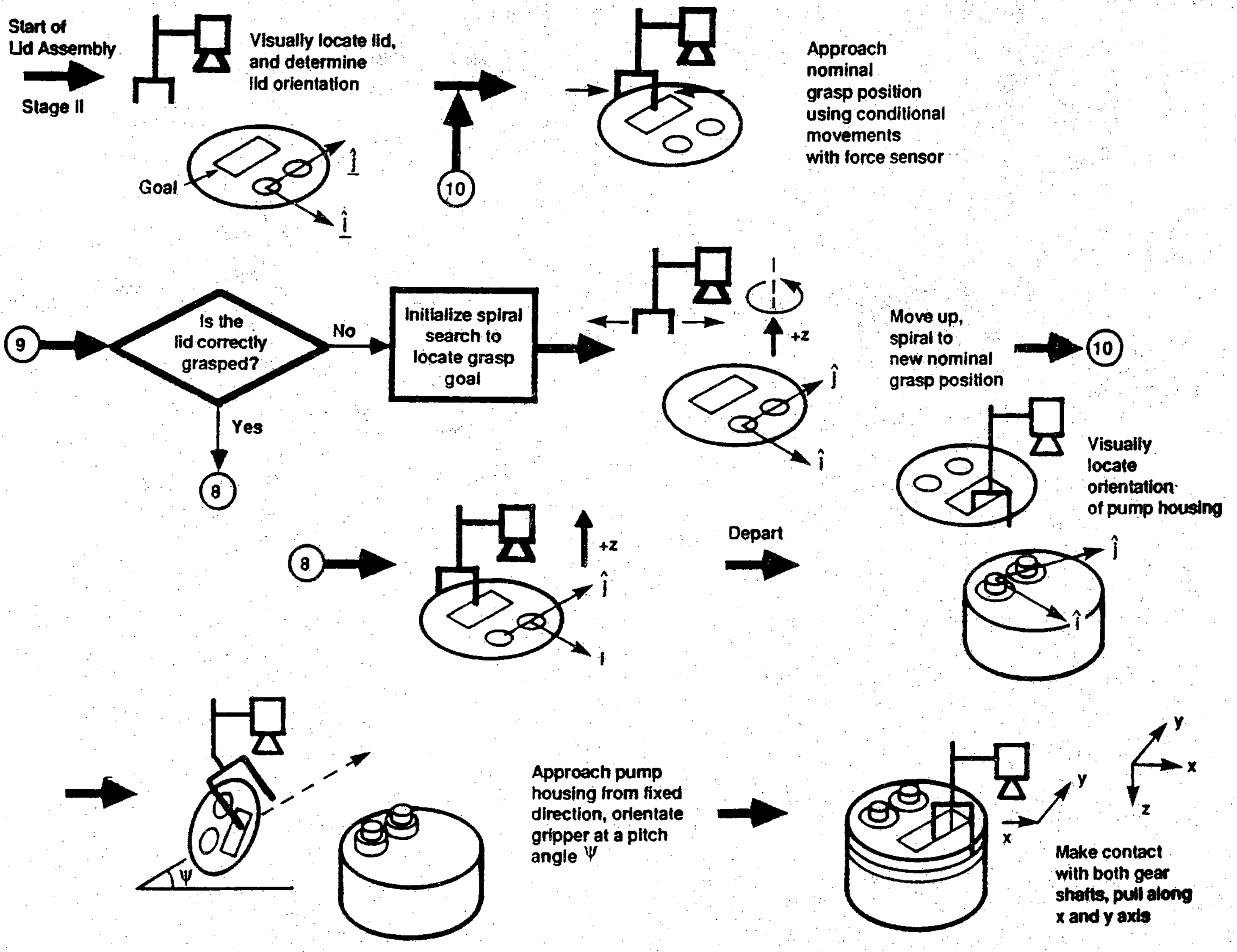


Figure 17: Lid Assembly

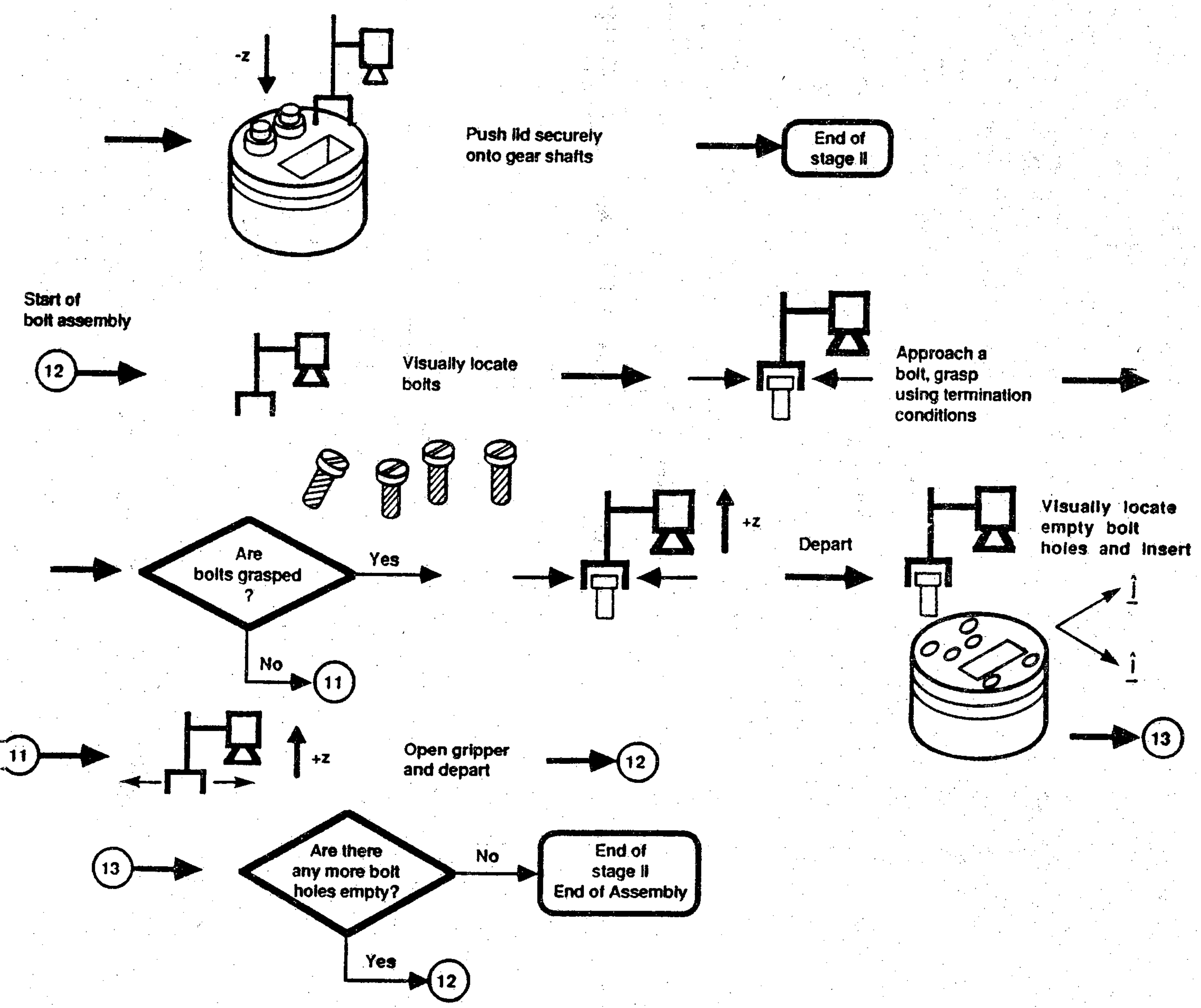


Figure 18: Bolt Assembly

Summary

In this chapter we briefly discussed the major problems encountered in robot based assembly systems. It is evident from current applications that ease of programming and the ability of the robot systems to deal with uncertainty is essential for robust and flexible assembly systems. Sensors are an essential component in detecting uncertainty. However, information which is generated from the sensors can only be utilized when the geometric context is correct. Multiple sensors may be used to update an object's position. Research in this area is currently in progress. Sensors can also be used to directly coordinate the motion of the robot to deal with many positional uncertainties. Some examples have been illustrated in this chapter.

We illustrated the use of simple commercially available sensors in the assembly of a diesel engine oil pump. This experiment shows that parts do not have to be precisely presented for a reliable assembly operation, if sensing and sensor-based motion strategies are utilized.

Acknowledgements

I would like to thank Greg Shiu, Zoreh Erfan, Peter Bovenzi, Cathy Tanner, Jill Comer and Rick Guptill for their comments on the initial text.

I would also like to thank Mary Schultz for her excellent typing.

References

[Ahmad 84]
Ahmad, S., "Robot Level Programming Languages and the SRIL-90 Language", Proceedings of the IEEE COMPSAC Conference, Chicago, IL, November 1984.

[Ahmad 87]
Ahmad, Shaheen, "Analysis of Robot Drive Train Errors, Their Static Effects and Their Compensation," TR-EE-87-4, Purdue University, February 87.

[Bege 84]
Bege, S., "An Optical Tactile Array Sensor," SPIE Vol. 521 Intelligent Robots and Computer Vision (1984); p. 271.

[Bonner & Shin 82]
Bonner, S., Shin, K. G., "A Comparative Study of Robot Languages", IEEE Computer, Vol. 15, No. 12, pp. 82-96, December 1982.

[Brady *et al.* 82]
Brady *et al.*, "Robot Motion: Planning and Control, MIT Press, 1982.

[Brooks 82]
Brooks, R.A., "Symbolic Error Analysis and Robot Planning," International Journal of Robotics Research, 1, 4, 1982.

[Brown 85]
Brown, M.K,. "Locating Object Surfaces with an Ultrasonic Range Sensor," Proceedings of the 1985 IEEE Robotics and Automation Conference, St. Louis, MO, March 1985.

[Brown 86]
Brown, M.K., "On Ultrasonic Detection of Surface Features," Proceedings of the 1985 IEEE Robotics and Automation Conference, San Francisco, April 1986.

[Durrant-Whyte 86]
Durrant-Whyte, H. F., "Consistent Integration and Propagation of Disparate Sensors Observations," Grasp Laboratory Report, University of Pennsylvania, April 1986.

[Fu 82]
Fu, K.S., "Pattern Recognition for Automatic Visual Inspection", IEEE Computer, Vol. 15, No. 12, pp. 34-40, December 1982.

[GE] "PN-2304 Optomation II VPL Reference Manual", General Electric Co. NY.

[Gonzalez & Safabakhsh 82]
Gonzalez, R.C., Safabakhsh, R., "Computer Vision Techniques for Industrial Applications and Robot Control", IEEE Computer, Vol. 15, No. 12, pp. 17-32, December 1982.

[Goto *et al.* 80]

Goto, I., Takaeyasu, K., Inoyama, T., "Control Algorithm for Precision Insert Operation Robots," IEEE Trans. Systems Man and Cybernetics, SMC-10, 1, 1980.

[Grimson 86]

Grimson, W.E.L., "Disambiguating Sensory Interpretations Using Minimal Sets of Sensory Data," Proceedings of the 1986 IEEE Conference on Robotics and Automation, San Francisco, March 1986.

[Gruver *et al.* 84]

Gruver, W.A., Soroka, B.I., Craig, J.J., Turner, T.L., "Industrial Robot Programming Languages: A comparative Evaluation," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-14, No. 4, July 1984.

[Hall *et al.* 82]

Hall, E., Tio, J.B.K., McPherson, C.A., Sadjadi, F.A., "Measuring Curved Surfaces for Robot Vision", IEEE Computer, Vol. 15, No. 12, pp. 42-54, December 1982.

[Harmon 84]

L.D. Harmon, "Automated Touch Sensing," International Journal of Robotics Research, Vol. 1, No. 2, 1982.

[Hayati 83]

Hayati, S., "Robot Arm Geometric Link Parameter Estimation", Proceedings of the 22nd IEEE Conference on Decision and Control, December 1983.

[Hayward & Paul 84]

Hayward, V. Paul, R.P., "Introduction to RCCL: A Robot Control "C" Library," Proceedings of the IEEE Robotics and Automation Conference, March 1984, Atlanta, Georgia.

[Henderson & Shilcrat 84]

Henderson, T., Shilcrat, E., "Logical Sensor Systems," Journal of Robotic Systems, 1(2), 169-193, 1984.

[Hillis 82]

Hillis, W.D. "A High-Resolution Image Touch Sensor," International Journal of Robotics Research Vol. 1, No. 2, (1982).

[Hirzinger & Dietrich 86]

Hirzinger, G., Dietrich, "Multisensory Robots and Sensor Based Path Generation," Proceedings of the 1985 IEEE Robotics and Automation Conference, San Francisco, April 1986.

[IBM 83]

"A Manufacturing Language Concepts and User's Guide", International Business Machines, IBM 1983.

[Inoue 74]

Inoue, H., "Force Feedback in Precise Assembly Laboratory, Massachusetts Institute of Technology, AIM-308, Aug. 1974.

[Kent & Albus 84]

Kent, E. W., Albus, J.S., "Servoed World Models as Interfaces Between Robot Control Systems and Sensory Data," Robotica, Vol. 2., pp. 17-25, 1984.

[Lee 82]

Lee, C.S.G., "Robot Arm Kinematics, Dynamics and Control", IEEE Computer, Vol. 15, No. 12, pp. 63-80, December 1982.

[Lozano-Perez 76]

Lozano-Pérez, T., "The Design of a Mechanical Assembly Systems," Artificial Intelligence Laboratory, Massachussettes Institute of Technology, AI-TR-397, 1976.

[Lozano-Perez 83]

Lozano-Pérez, T., "Robot Programming," Proceedings of the IEEE, Vol. 71, No. 7, July 1983.

[Luh 83]

Luh, J.Y.S., "An Anatomy of Industrial Robots and Their Controls," IEEE Transaction on Automatic Control, Vol. AC-28, pp. 133-153, Feb. 1983.

[Mason & Salisbury 85]

Mason, M. T., Salisbury, J. K. Jr., "Robot Hands and the Mechanics of Manipulation," MIT Press, 1985.

[Mason 81]

Mason, M. T., "Compliance and Force Control for Computer Controlled Manipulators," IEEE Trans. Systems Man and Cybernetics, SMC-11, 6, 1981.

[Mujtaba & Goldman 79]

Mujtaba, S.M., Goldman, R., "The AL Users Manual", STAN-CS-79-718, Stanford University, January 1979.

[Paul 81]

Paul, R. P., "Robot Manipulators: Mathematics, Programming and Control", MIT Press, 1982.

[Polaroid 82]

Polaroid Corporation, "Ultrasonic Range Finders," Polaroid Corporation, 1982.

[Raibert & Tanner 82]

Raibert, M.H., Tanner, J.E. "Design and Implementation of a VLSI Tactile Sensing Computer," International Journal of Robotics, Research 1,3 (1982), 33-44.

[Requicha & Chan 86]

Requicha, A., Chan, S. C. "Representation of Geometric Features, Tolerances, and Attributed in Solid Modelers Based on Constructive Geometry," IEEE Trans. Robotics and Automation, RA-2, 3, 1986.

[Ruokangas *et al.* 86]

Ruokangas, C.C., Black, M.S., Martin, J. F., Schoenwald, J. S., "Integration of Multiple Sensors to Provide Flexible Control Strategies," Proceedings of the IEEE Robotics and Automation Conference, March 1986, San Francisco, CA.

[Shimano *et al.* 84]

Shimano, B. E., Clifford, C. G., Spalding III, C. H., "VAL-II: A New Robot Control System for Automatic Manufacturing", Proceedings of the IEEE Robotics and Automation Conference, Atlanta, Georgia, March 1985.

[Sinden & Boie 86]

Sinden, F.W., Boie, R.A., "A Planar Capacitive Force Sensor with Six Degrees of Freedom," Proceedings of the 1985 IEEE Robotics and Automation Conference, San Francisco, April 1986.

[Stevenson 86]

Stevenson, C., "Offline Programming for Model Based Sensor Controlled Robots," Ph.D. Thesis 1986, School of Electrical Engineering, Purdue University, West Lafayette, Indiana.

[Taylor 76]

Taylor, R.H., "The Synthesis of Manipulator Control Programs from Task-Level Specifications," Artificial Intelligence Laboratory, Stanford University, AIM-282, July 1976.

[Taylor *et al.* 83]

Taylor, R. H., Summers, P.D., Meyer, J.M., "AML: A Manufacturing Language", The International Journal of Robotics Research, Vol. 1, No. 3, 1982.

[Whitney *et al.* 84]

Whitney, D. E., Lozinski, C. A., Rourf, J. M., "Industrial Robot Calibration Method and Results," Proceedings of ASME Computers in Engineering Conference, Las Vegas, 92-100, August 1984.

[Whitney 85]

Whitney, D.E., "Historical Perspective and State of the Art in Robot Force Control," Proceedings of the 1985 IEEE Conference on Robotics and Automation, St. Louis, March 1985.

[Will & Grossman 75]

Will, P.M., Grossman, D.D., "An Experimental System for Computer Controlled Mechanical Assembly," IEEE Transactions on Computers, C-24, 9(1975), 879-888.

[Wu 83]

Wu, C.H., "The Kinematic Error Model for the Design of Robot Manipulators", Proceedings of the 1983 American Controls Conference, San Francisco, CA, June 1983.