STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM - 220

STAN-CS-396

THE USE OF SENSORY FEEDBACK IN A PROGRAMMABLE ASSEMBLY SYSTEM

BY

Robert Bolles and Richard Paul

SUPPORTED BY

ADVANCED RESEARCH PROJECTS AGENCY
ARPA ORDER NO. 2494
PROJECT CODE NO. 3D30

COMPUTER SCIENCE DEPARTMENT
Scnool of Humanities and Sciences
STANFORD UNIVERSITY

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM-220

OCTOBER 1973

COMPUTER SCIENCE DEPARTMENT
REPORT NO. CS-396

THE USE OF SENSORY FEEDBACK IN A PROGRAMMABLE ASSEMBLY SYSTEM

by

Robert Bolles and Richard Paul

ABSTRACT: This article describes an experimental, automated assembly system which uses sensory feedback to control an electro-mechanical arm and TV camera. Visual, tactile, and force feedback are used to improve positional information, guide manipulations, and perform inspections. The system has two phases: a 'planning' phase in which the computer is programmed to assemble some object, and a 'working' phase in which the computer controls the arm and TV camera in actually performing the assembly. The working phase is designed to be run on a mini-computer.

The syst cm has been used to assemble a water pump, consisting of a base, gasket, top, and six screws. This example is used to explain how the sensory data is incorporated into the control system. A movie showing the pump assembly is available from the Stanford Artificial Intelligence Laboratory.

This research was supported in part by the Advanced Research Projects Agency of the Office of Defense under Contract No. DAHC 15-73-C-0435,

TABLE OF CONTENTS

INTRODUCTION

At the Stanford Artificial Intelligence Project we are developing a programmable automation system. The system consists of an electro-mechanical arm and a TV camera interfaced to a computer. It is a part of a larger research program on perception, manipulation and problem solving [McCarthy]. The automation system has two phases; a planning phase during which an operator teaches the computer the task it is to perform, and a working phase in which the result of the planning phase (ie. the plan) is used by the computer to control the arm and TV camera in performing the actual assembly. While performing the assembly, the computer interacts with the task through visual, tactile, and force feedback. The planning is both manual and symbolic. That is, the operator can manually move the arm to &fine positions (ie. programming by doing), but the force limits and vision are defined in programming languages. The planning is done once per task. The resulting plan can be used repeatedly and is designed to be run on a mini-computer. Currently the visual processing is handled by an independent, but cooperating task running on a large computer [Feldman].

Mechanical arms have been used in industry for spot welding, pick-and-place operations, etc., but with little or no sensory feedback. They have also been used in conjunction with TV cameras by research organizations for manipulating idealized block structures. The system described here represents one of the first successful attempts to incorporate sensory feedback into a system which is designed to deal with realistic assembly tasks [Dewar].

This article describes the automated assembly of a model "T" Ford racing water pump as a demonstration of the system and its concept. The emphasis is on explaining how the various types of sensory feedback are accomplished. The assembly of the pump consists of locating the pump base, mounting the top with a gasket, bolting the top down with six screws, and testing to see that the rotor turns freely.

WORK STATION AND TASK DESCRIPTIONS

The work station consists of an electromechanical arm, a TV camera, and a work space containing tools, dispensers, and parts (see picture I), The arm is shown in picture 2. It has an absolute positioning accuracy of approximately one tenth of an inch. Its working area and speed are similar to those of a human [Scheinman]. The control of the arm will be discussed in the next section. The camera's pan, tilt, focus, and lens turret are computer controlled (see picture 3). The components of the work space are shown in picture 4 and labelled in the related diagram (see picture 5).

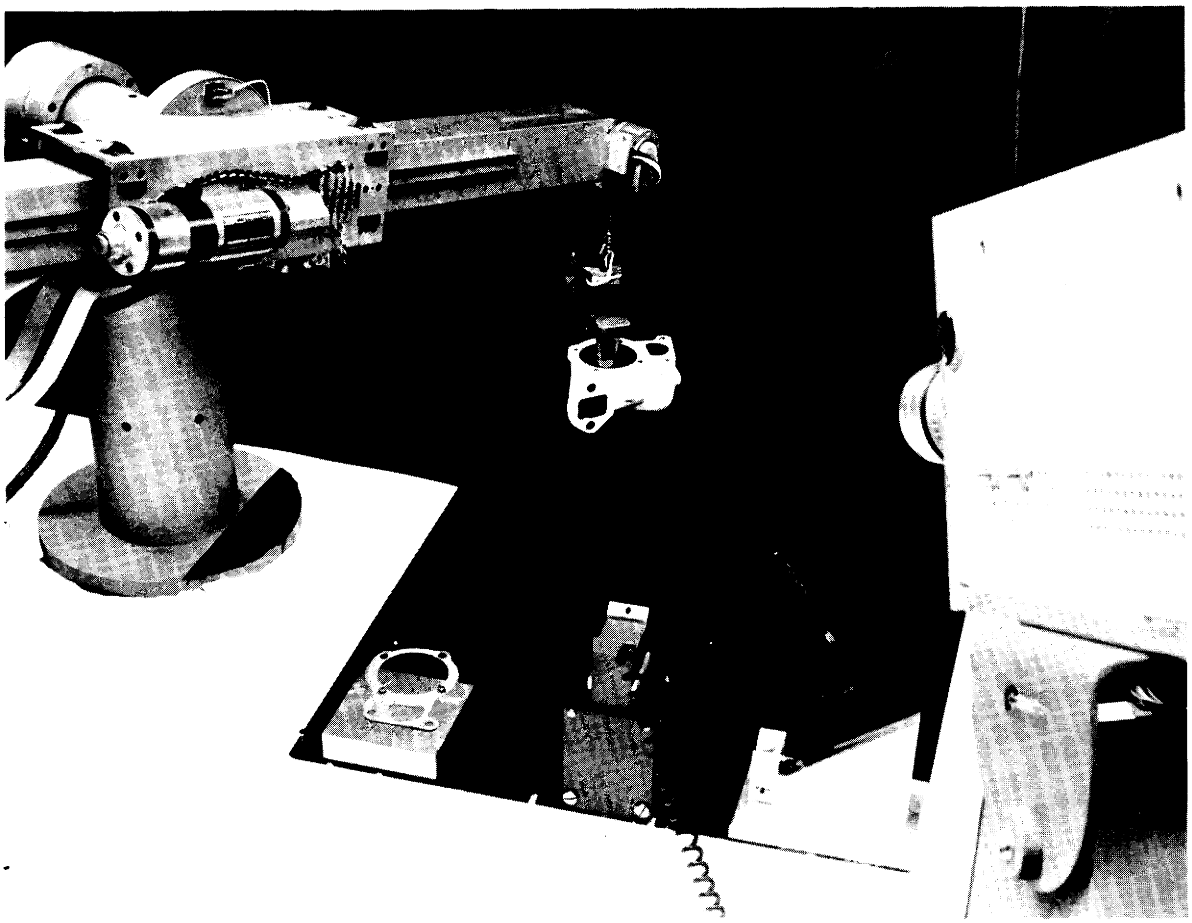
The assembly steps are listed below. They represent an ordered list of tasks that were accomplished and will be referenced in the following discussion as various sensing techniques are discussed.

	Visually locate the pump base
II.	Determine the final grasping position by touch
III.	Place the pump base in its standard position
IV.	Insert the two aligning pins
V.	Put on the gasket
VI.	Visually check the position of the gasket
VII.	Put on the top
VIII.	Screw in the first two screws
IX.	T&e out the aligning pins
X.	Screw in the last four screws
XI.	Check the force required to turn the rot or

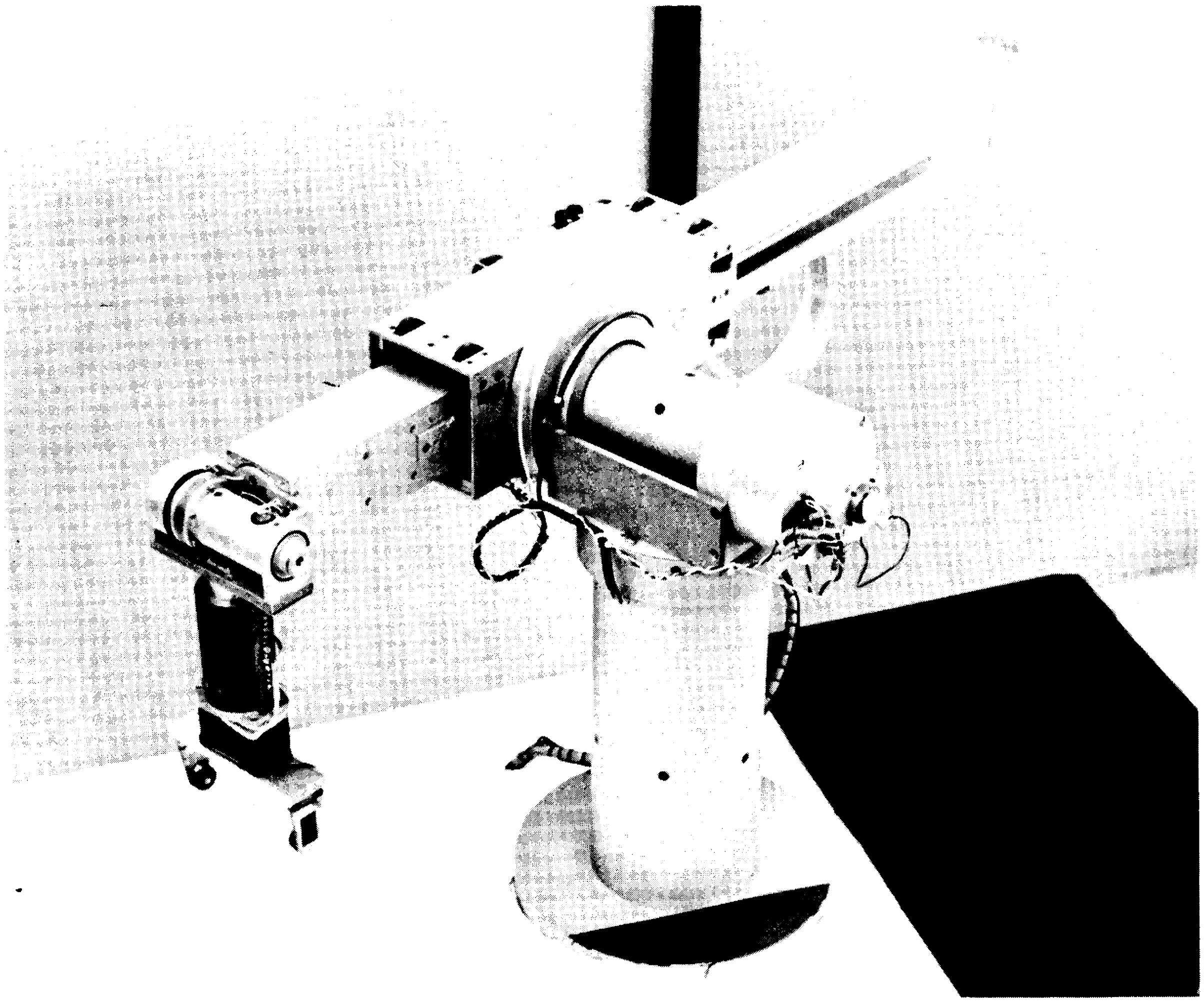
The table below lists the approximate times associated with the different parts of the assembly task.

PLANNING PHASE	60 sec. (execution on PDP- 10)
WORKING PHASE	5 min. 32 sec. (total)
MANIPULATION	4 min.
LENS CHANGING & REFOCUSING	1 min. 30 sec. (a part of which can be overlapped with manipulation)
VISUAL PROCESSING	2 sec. (execution on POP- 10) (for both location and gasket checking)

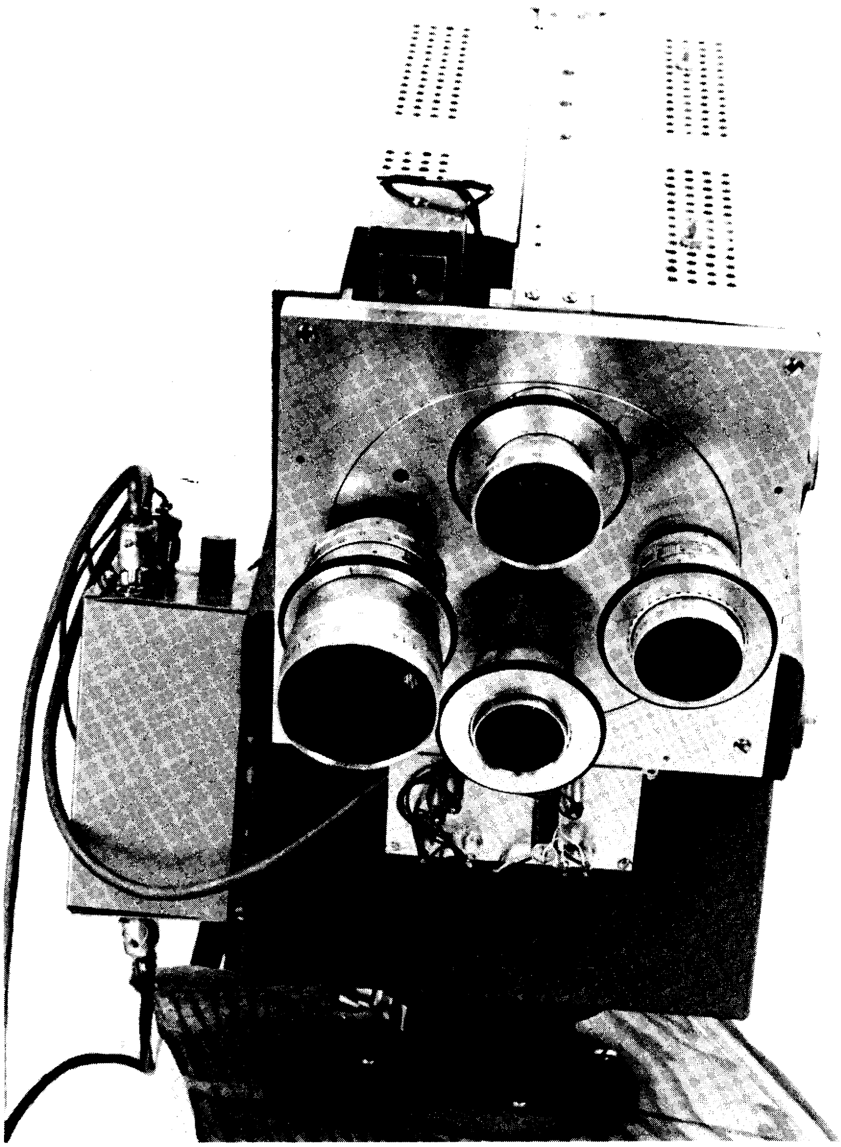
Since this task (assembling the water pump) was the first of its kind, the tools, dispensers, and programming system were developed as needed. This development extended over four or five months. However, if the tools and dispensers already existed for a task of equal complexity, programming the manipulations would only take a couple of days. In addition, minor changes to such an assembly program could be made quite easily. For example, adding another screw to the water pump assembly would only require a few minutes work.



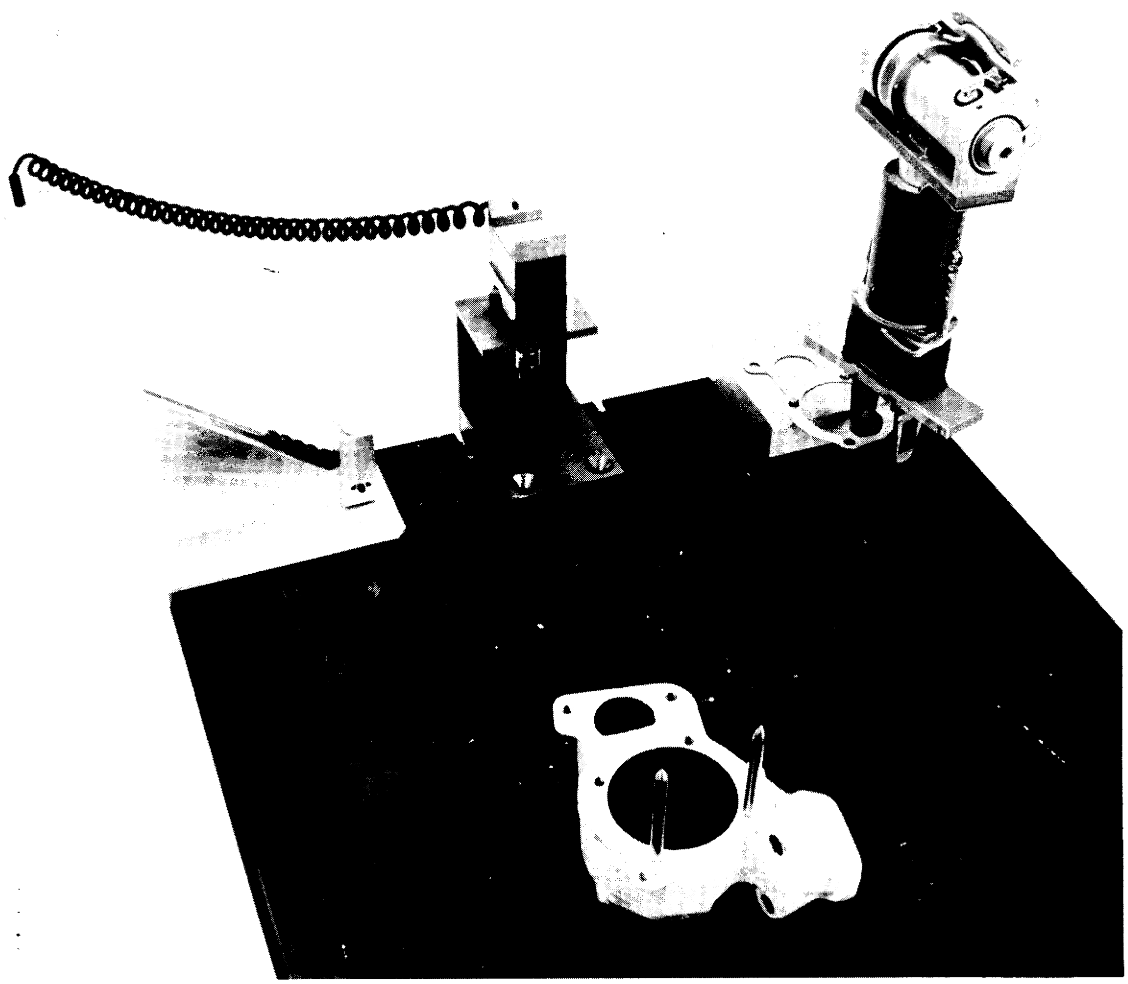
Picture 1.
General View of the Work Station



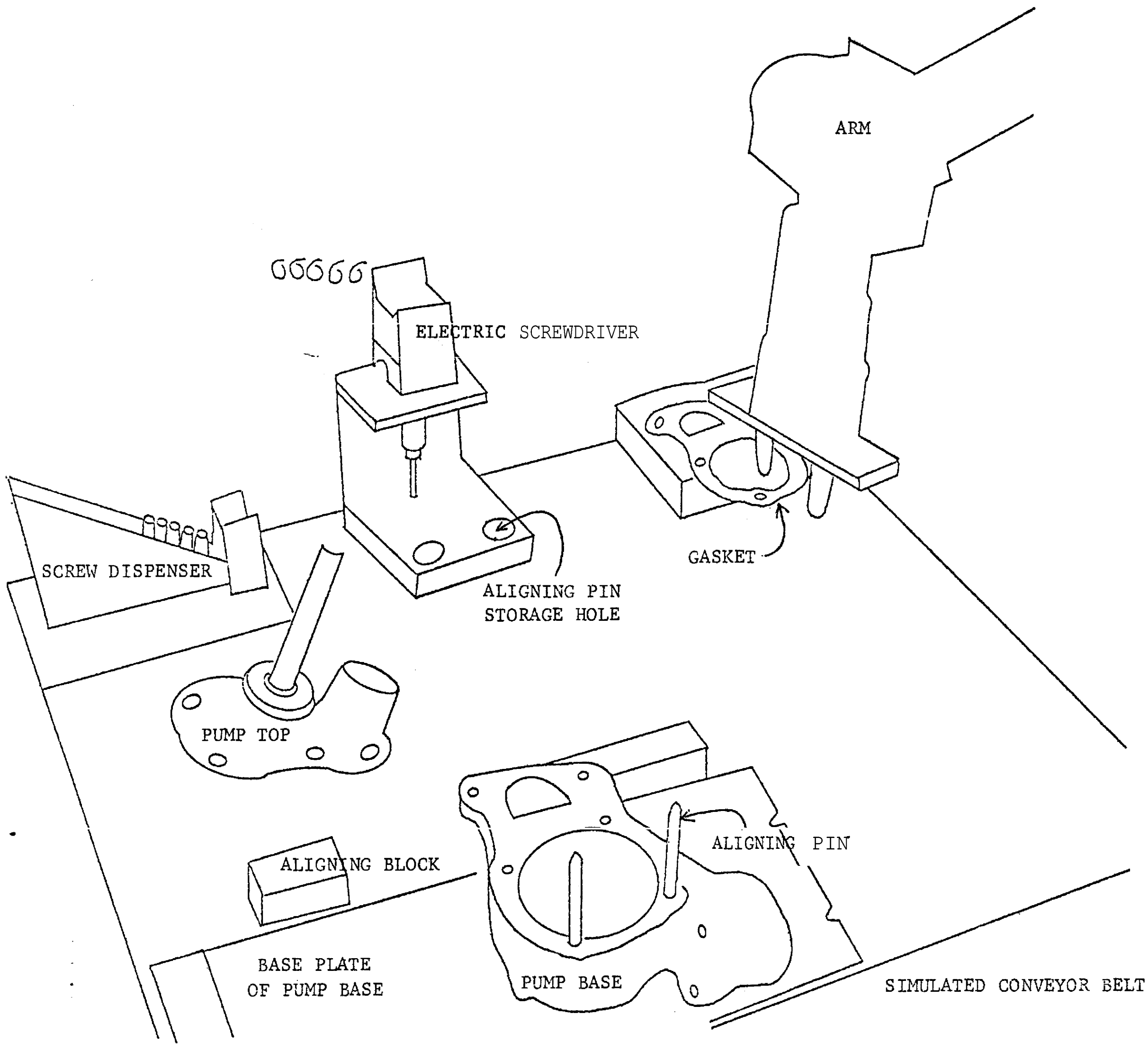
Picture 2.
Scheinman Arm



Picture 3.
Cohu Camera



Picture 4.
The Work Space
(see picture 5 for labels)



Picture 5.
A Labelled Diagram of the Work Space

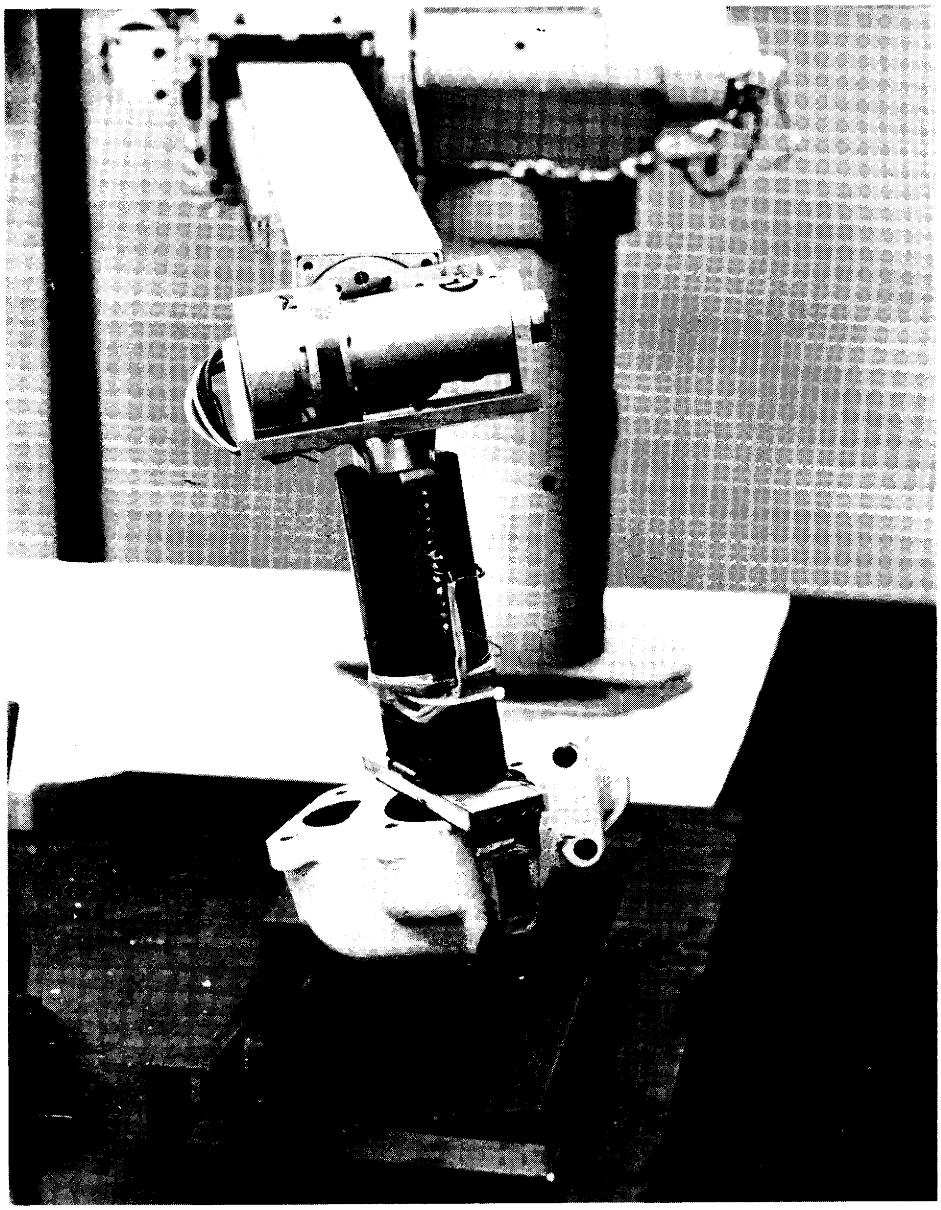
CONTROL OF THE ARM

The arm has six joints (five rotary and one sliding) and it is possible to place the hand at any position and at any orientation. Each joint is powered by an electric motor which is under computer control. The joint positions are measured by potentiometers and are read into the computer via A/D converters. Similarly, the joint velocities are read into the machine via A/D converters from tachometer generators. A real-time program (the servo loop) directly controls the joints' forces and indirectly controls joint velocities and positions. Every sixtieth of a second the servo reads the position and velocity information and determines the joint output torques from the difference between the observed and planned values. There is a built-in safety feature which shuts off all drives and applies all brakes if the computer fails to respond every sixtieth of a second. A more detailed description of the servo loop can be found in reference [Paul].

There is a set of equations based upon the kinematic structure of the arm which relates the force, position, and velocity of the hand to the combination of forces, positions and velocities of the six joints. These equations, involving sines, cosines, roots, etc. are solvable on the computer even though they contain some degenerate sub-cases. The solution routine is currently part of the planning section and is used to compute the forces required to compensate for the weight of the arm and any load it may be carrying. These compensating forces are always applied when the arm is in motion. Thus, if all the brakes are turned off the arm will not fall; it will remain stationary, but will be free to be moved manually in any direction.

If we want the hand to exert a force in some direction, the solution routine can be used to compute the required joint forces. When these forces are added to the normal compensating forces the arm will exert the specified force.

Normally, when we have the arm exert a force, we want the hand to be free to move in the direction of the force. Sometimes it is important to provide some additional freedom so that the arm can comply with external constraints. For example, if we want the arm to slide an object across an essentially horizontal surface, we want to allow the arm to move up and down so that it can conform with the surface as it moves across it. This freedom is achieved by servoing all the joints except one joint which provides for a vertical motion. This one unservoed joint is called a 'free' joint. Free joints can also provide the freedom to spin about some axis. In the pump assembly, for example, after the pump base has been located and picked up, it has to be placed in a standard position. The standard position is defined by a rectangular corner formed by a pair of aligning blocks. The first step in this alignment involves positioning a straight edge of the pump base along a surface of one of the blocks. This is accomplished by pushing the base into the block and simultaneously freeing the joint which allows the base to spin so that it can align itself with the surface (see picture 6).



Pict ure 6.
Art-n Pushing the Pump Base against the Aligning Hocks

To summarize, a motion of the arm consists of a trajectory, some compensating forces, and possibly a force to exert and some joints to free, In addition the termination of the motion has to be specified. It can be defined as a position to be reached, a force limit to be reached, an activation of a touch sensor, etc. Thus, the arm can be told to screw in a screw until a certain torque is reached, or it can be told to insert a shaft until a certain force limit is reached (indicating that the shaft has been seated). The next section will explain in detail how the arm is programmed to perform this type of feedback.

FORCE FEEDBACK (WITH AN EXAMPLE ARM PROGRAM)

The positioning of the pump base relative to the arm is not accurate enough to allow the arm to insert a pin in a number 10 screw hole reliably. Therefore, to increase the reliability, a spiral search is used to try all nearby locations if the initial insertion attempt has failed. Picture 7 shows the arm inserting a pin in a hole. The first insertion attempt fails because the pin lands on the top of the base (see frame B in picture 7). The second attempt succeeds.

Three things can happen when the arm is trying to insert a pin: (1) the pin can go in the hole, (2) the pin can miss the hole and land on the top of the base beside the hole, or (3) the pin can miss the hole and also miss the top of the base. To test for these three possibilities the insertion is broken into two parts:

A. Try to insert the pin part way ... if it fails to go in part way, it must have landed on top of the base beside the hole (case 2), so continue around in the spiral and try another spot. If it went in part way, go to step B.

B. Try to seat the pin in the hole (i.e. move down a short distance and expect to meet some resistance as the pin seats in the hole) ... if no resistance is felt, the pin must have missed the hole and the top of the base (case 3), so continue around in the spiral. If resistance is felt, the pin is properly seated (case 1).

What follows is a hand language program to carry out this algorithm. It is included along with a detailed explanation of the various instructions in order to show the current level of programming required by the system.

The position of the hand to pick up the pin is referred to as P. This position is defined by moving the hand to where the pin is located and typing "HERE P." The program reads the current position of the hand and stores it in P. Similarly the hand (holding the pin) is moved to the position for insertion and "HERE T" is typed. Manually moving the arm to define positions and orientations is the easiest way of programming some assembly operations. It is a form of "programming by doing" or "learning by doing."

```
        MOVE P                  ;GO TO THE PIN
        CLOSE 0.1
        MOVE T                  ;GO TO THE HOLE
        SEARCH .07
L1:     MOVE T                  ;GO TO THE HOLE
        STOP [0 0 -50]
        CHANGE [0 0 -1] 0.6     ;TRY TO GO DOWN WITHOUT MEETING RESISTANCE
        SKIPE 23
        AOJ L1
```

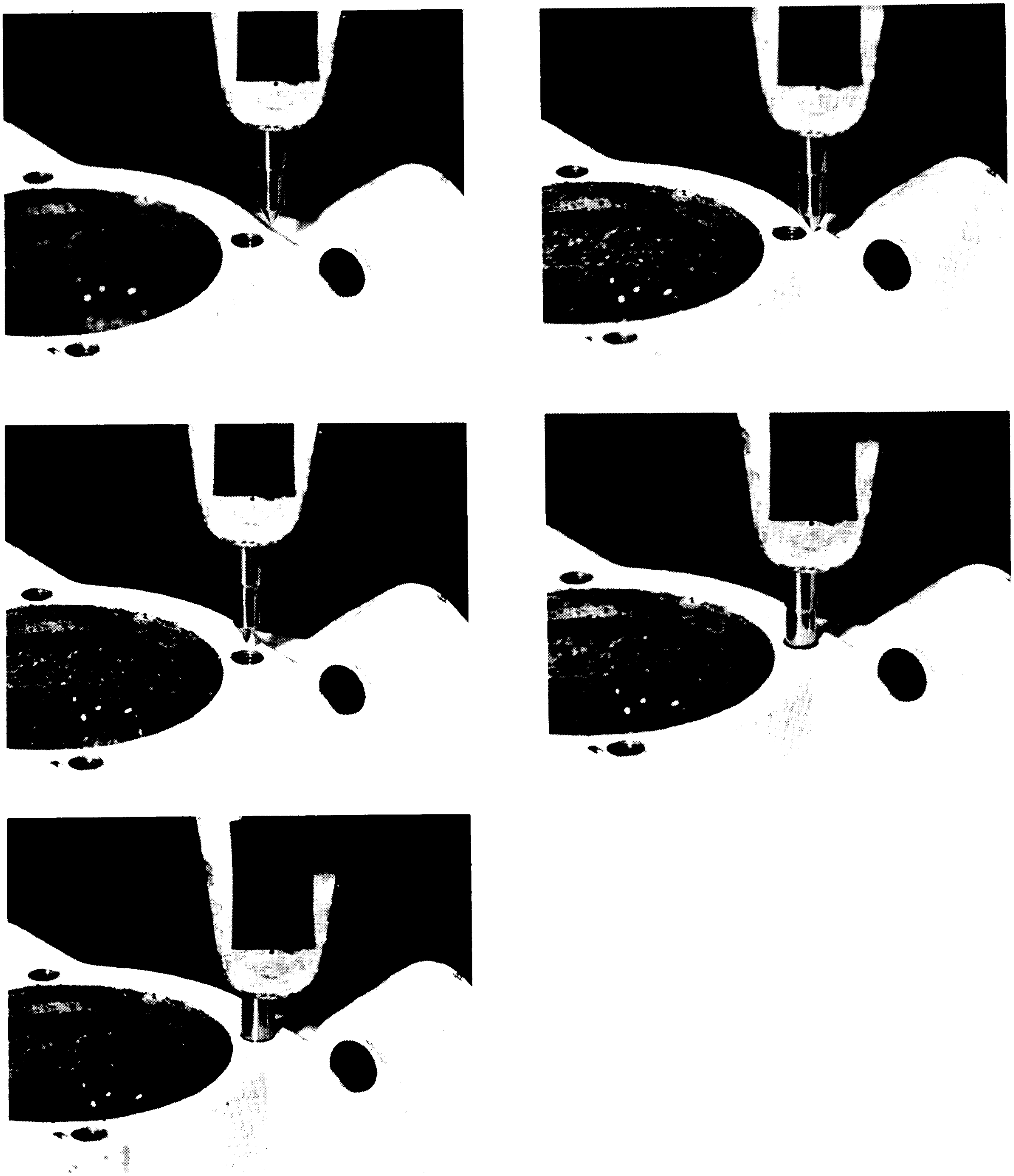


Figure 7,
Inserting a Pin in a Screw Hole

A. Pin Poised near Hole
B. Pin Sitting on the Base, beside the Hole
C. Pin Poised over the Hole
D. Pin Partially Inserted in the Hole
E. Pin Seated in the Hole

```
STOP [0 0 -50]
CHANGE [0 0 -1] 0.6          ;SHOULD MEET SOME RESISTANCE
SKIPN 23
AOJ L1
SAVE H
OPEN .5
CLOSE 0.1                    ;AND CHECK THAT IT IS STILL THERE
OPEN 1
```

The first instruction generates a trajectory from the current location of the hand to the position "P". The hand is then in position to grasp the pin. The next instruction, "CLOSE 0.1", causes the fingers to close until they grasp something. Every time the hand grasps anything, the minimum thickness must be specified, and forms an implicit inspection check. If the grasp is made and the check indicates that the opening is less than the minimum specified, the arm will stop operation and indicate the error.

With the pin now in hand the arm moves to the insertion point at "T". The "SEARCH .07" instruction sets up counters to conduct a spiral search of .07 inch steps. We now enter the insertion loop at label L1, a move is made to "T" and the hand is directed to move down 0.6 inches by the CHANGE instruction. The numbers within square bracket "[0 0 -1]" indicate the direction and the scalar "0.6", the distance to move. The previous instruction "STOP [0 0 -50]" will cause the arm to stop if the force in the downwards direction exceeds 1.50 ozs. during the "CHANGE". Now the relationship between the position "T" and the hole is such that if the pin is inserted in the hole it will meet no resistance during the 0.6 inch motion. If the pin is beside the hole and lands on the top of the pump, the force will quickly reach 50 ozs. and the hand will stop. If the hand fails to stop on the force limit, indicating that the pin is either in the hole or has missed the hole and the top of the base, an "ERROR" state is generated. In this particular case, the error is error 23. The instruction following the "CHANGE," "SKIPE 23" will cause the next instruction to be skipped if the error occurred, indicating in this case that all is well.

If the pin has landed on the top of the pump, missing the hole, the force limit is reached and the arm stops without generating an error state. When the SKIPE 23 instruction is executed no skip occurs and the AOJ L1 instruction is executed. AOJ is a mnemonic for "add and jump." The adding that occurs is the addition of the search step to the current position. The jump is to the label, L1, and the spiral search continues. The arm will stay in this loop, searching around "T" in 0.07 inch steps and trying to insert the pin in the hole until the pin moves down without meeting resistance.

After the pin has successfully been inserted part way, the stopping force is set to 60oz. and the hand is driven down 0.6 inches. If the pin is in the hole, the hand will stop before going 0.6 inches and no error will occur. The error test is a "SKIPN 23" instruction which causes a skip if error 23 does not occur. If the pin has missed everything, the "AOJ" is executed and the spiral is continued.

The "SAVE H" instruction saves the position that the hand was in when it inserted the pin. Thus, to return to that position, the following instructions could be executed:

```
MOVE T
RESTORE H
```

The "RESTORE H" modifies the position T by the saved difference H.

The last two instructions double-check the pin placement by making sure that the pin remained in the hole after the hand released it. More is said about this type of checking in the section on touch sensing.

VISUAL FEEDBACK

Digitized TV input represents a great possibility for visually locating, inspecting, and aligning parts. Unfortunately the systemization of visual techniques has progressed much more slowly than originally expected. This general statement is also true with respect to the visual feedback used in the pump assembly. The primitives are special purpose techniques which work within a system containing detailed models of the expected scenes and a set of specific heuristics.

The TV camera contains a standard vidicon which produces a 256x333 array of intensities. Each intensity value is in the range of 0 to 15. The camera's pan, tilt, focus, filter wheel and lens turret are computer controlled. A major problem with such a system is the calibration of the camera with respect to the arm [Sobel] and [Gill].

The software primitives used include:

(1) a "beam" which locates the first discontinuity (black to white, white to black, etc.) on a ray through the picture.

(2) a "blob localization routine" which isolates a blob on a contrasting background by surrounding it with a box.

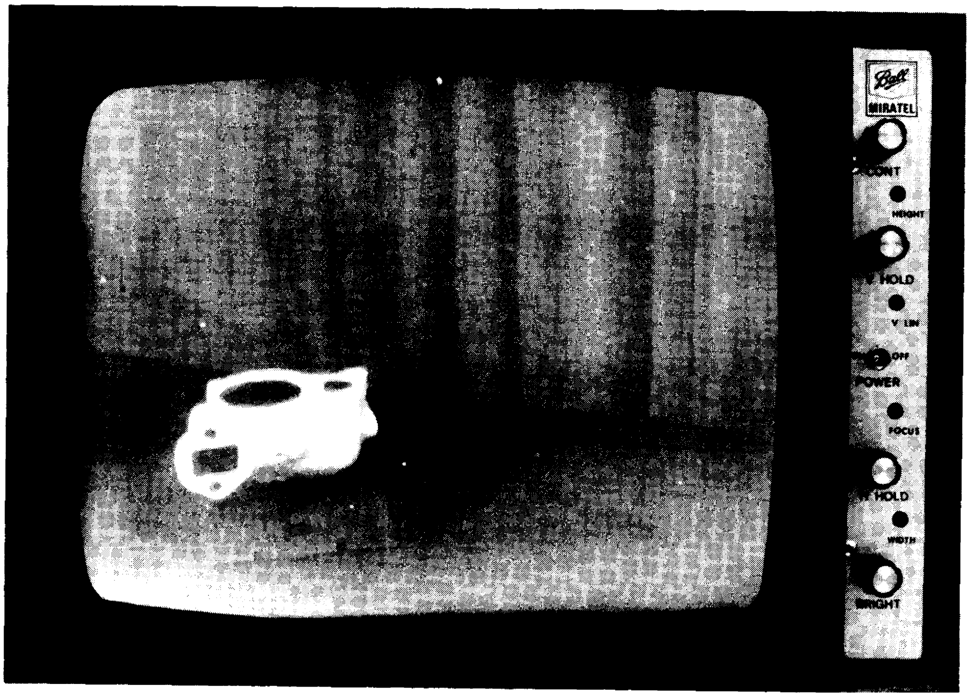
(3) a "convex blob characterizer" which determines the center, width, and height of a convex blob by bouncing around inside it,

One of the tasks achieved by visual feedback was the location of the pump base in terms of its X,Y,Z position and orientation. This was accomplished in two steps: (A) the general location (using the 25mm lens, see picture 8). and (B) the specific location (using the 50mm lens, see picture 9). The model used for the general location was fairly simple. It consisted of the following facts:

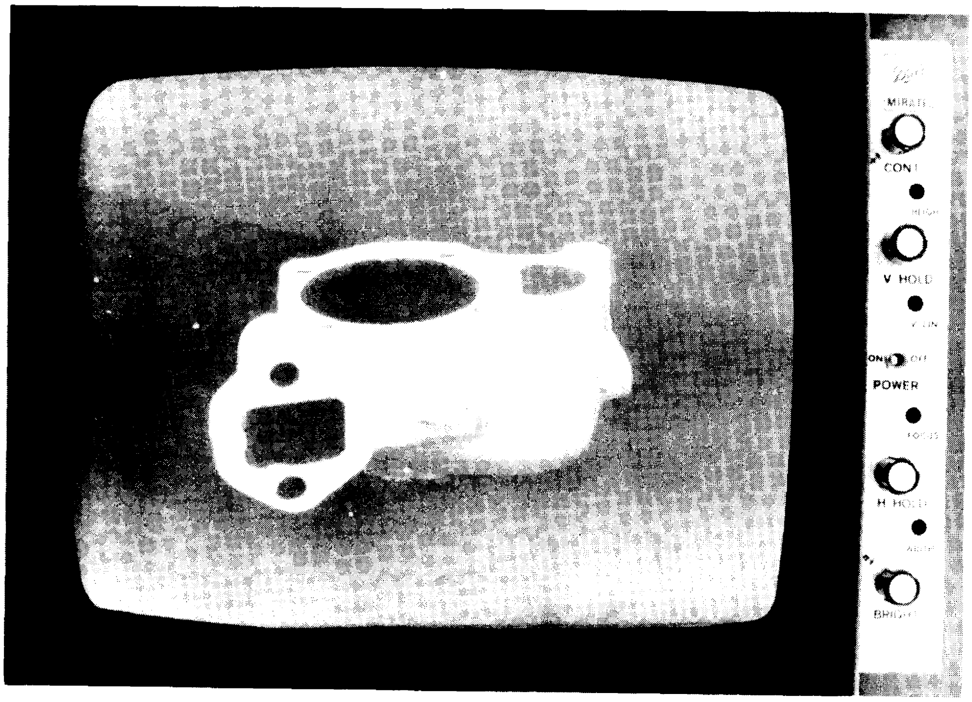
(1) the pump base would be a white blob on a black background (the white and black combination is not necessary --- any contrast in color or intensity would be sufficient)

(2) the pump base would appear within a certain portion of the table (ie. on a part of the simulated conveyor belt).

Therefore, to determine the general location, the TV was aimed and focused so that it covered the specified portion of the simulated conveyor belt, the blob location routine was applied within the appropriate part of the picture, and the support hypothesis was used to determine the position well enough to change lens and re-aim with the 50mm lens.



Picture 8.
General View of the Pump Base
on the Simulated Conveyor Belt (using 25mm Lens)

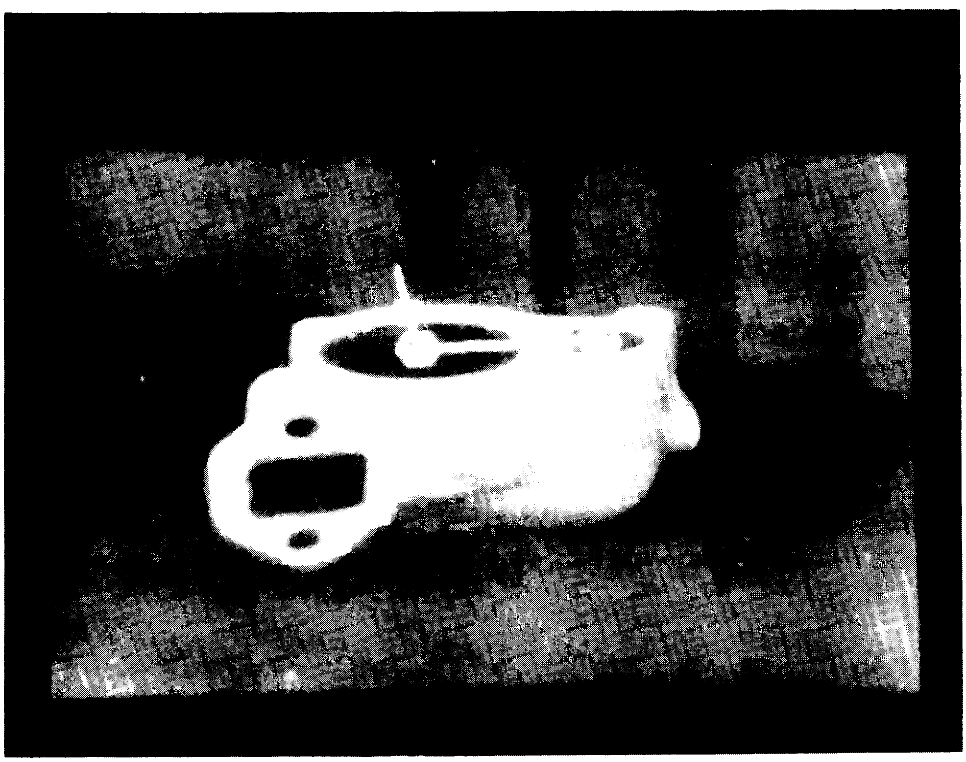


Picture 9.
Close-up View of the Pump Base (using 50mm Lens)

The "specific location" model was considerably more detailed. It consisted of a structured set of features (convex holes), their relative sizes, positions, and contrasts. The following steps were taken to locate the two large holes on top: beams were sent through the blob at promising positions and angles, the blob characterizer was applied whenever a discontinuity was noticed, and the holes were classified according to their relative sizes (based upon the general location information and the size of the largest hole after it had been found). When the two large holes on top had been located, the position and orientation of the pump base could be determined from the known 3-D measurements of it. Picture 10 shows the line of centers and a line indicating the orientation for the initial grasping position of the hand. This location information was then sent to the arm which used touch to determine the final grasping position. The touch sensing involved is discussed in the next section.

The second task accomplished by visual feedback was the inspection of the gasket's position after it had been put on. The location of the pump base with respect to the standard position was known and from this the position of the two large holes could be determined. To check the positioning of the gasket a picture was taken of the base just before the gasket was put on and another picture was taken just after it was put on. These pictures were "differenced." That is, a new picture was created by taking the absolute value of the difference between the intensities at all points. in theory only the gasket should appear in the difference. In practice other lower intensities arise because of shadows, slight image shifting, etc. Picture 11 shows the differenced picture with an overlayed display. Notice that the difference picture could again be interpreted as a white blob containing convex holes. The same convex blob characterizer was applied at the expected positions for the two large holes. If the centers were not within a certain tolerance (or they could not be found at all) the machine signalled an operator that the gasket was not on properly. In picture 11 the two crosshairs indicate the expected centers for the two large gasket holes. The solid dot indicates the observed center of the largest hole. In this case the observed center differed sufficiently from the predicted center to indicate that the gasket was not on correctly.

Picture 10.
The Pump Base with an Overlay Showing
the Computed Orientation



Picture 11.
The Differenced Picture of the Gasket with an Overlay Marking
the Expected Hole Centers and the Actual Center of the Large Hole

TOUCH SENSING

The hand has two fingers which can be opened or closed together. There is a microswitch on the inside of each of the fingers. These switches are binary in that they register only touching or not touching. They do not register force -- ie. how hard something is being grasped.

Even though the sensing mechanism is fairly primitive it can be combined with other techniques to provide some useful manipulation and feedback primitives. For example, position potentiometers measure the distance between the fingers. This distance can be used in conjunction with a simple model to provide feedback on what is being held in the hand. Typical models are "something of a given minimum thickness", "something whose thickness is within a given range", and "anything between the fingers". In the pump assembly, for example, after the hand inserts a pin in a screw hole, it opens and closes again to make sure that the pin is still there (see picture 12). In this case it is using the model of "anything between the fingers" because presumably only the pin could be there, If, for some reason, the pin fell into the large hole instead of seating in a screw hole, this' test would be sufficient to detect the mistake and the machine could notify an operator of the problem.

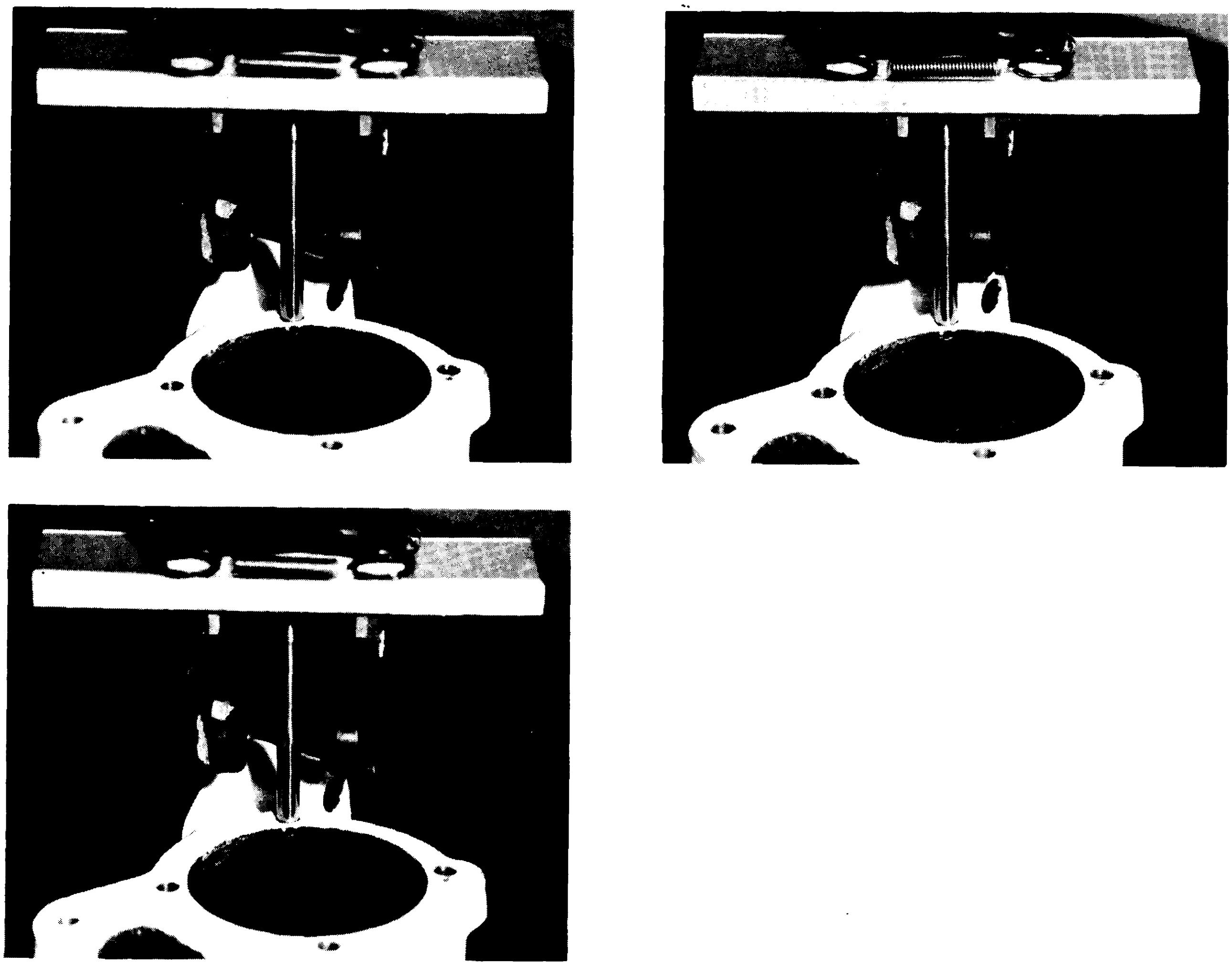
The signals from the two touch sensors are independent, making it possible to determine which touch sensor is being activated. This can be used, for example, to construct a "center-the-hand-over-an-object" procedure as follows:

(1) close the hand until one touch sensor (say sensor A) touches something.

(2) continue to close the hand, but move the arm so that the finger containing sensor A remains stationary --- stop when the second touch sensor touches something.

Picture 13 shows the hand centering on one side of the pump base. Notice that the hand is initially off-centered and ends up centered. Also notice that if the hand were simply told to close, it would move the whole pump base so that it ended up centered between the fingers. This may or may not be the desired result. This centering procedure was originally written as a combination of many primitives in the arm's language, but constant use prompted the implementation of a primitive called CENTER.

There are two other primitives, SAVE and RESTORE, which are related to touch sensing. The idea is that the arm can dynamically determine and save the difference between a predicted (or planned) position and an actual position. For example if the arm is told to insert a pin in a screw hole, it is given a specific, planned location for the hole. If the hole is slightly out of position and the hand successfully finds it by searching (as described in the force sensing section), the arm can SAVE the deviation of the hole from the predicted position. Therefore, when the hand returns to that hole, to take out the pin or insert a screw, it can RESTORE the deviation and avoid a second search.

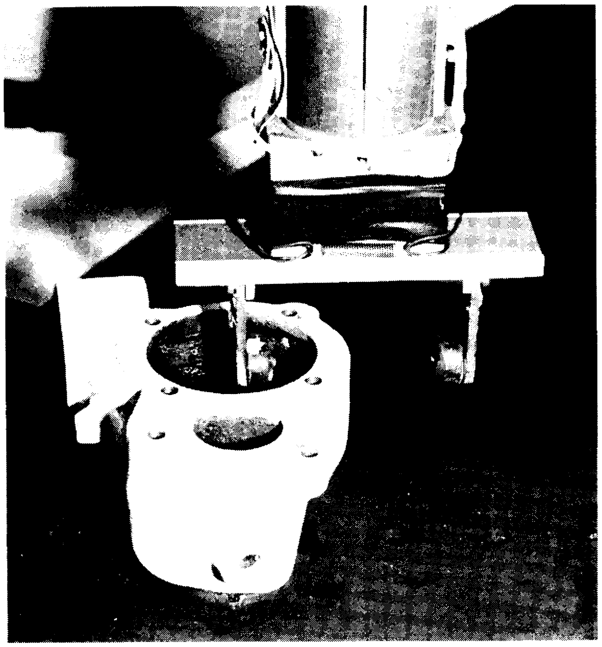


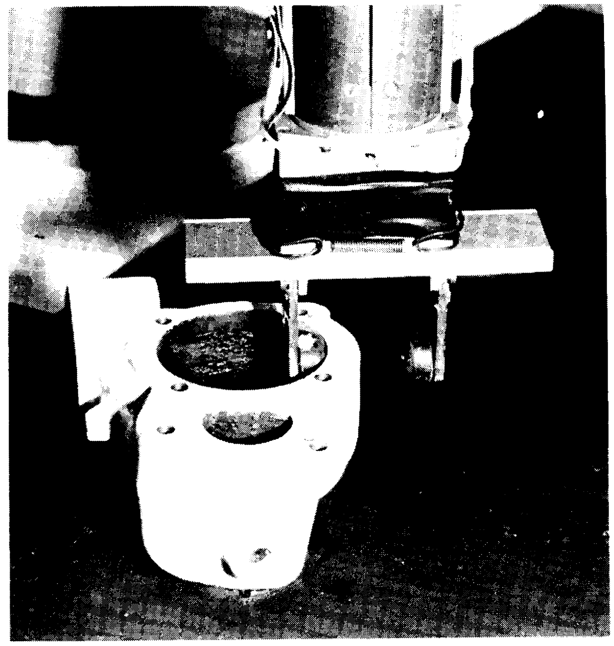
Picture 12.
Checking for the Presence of the Pin

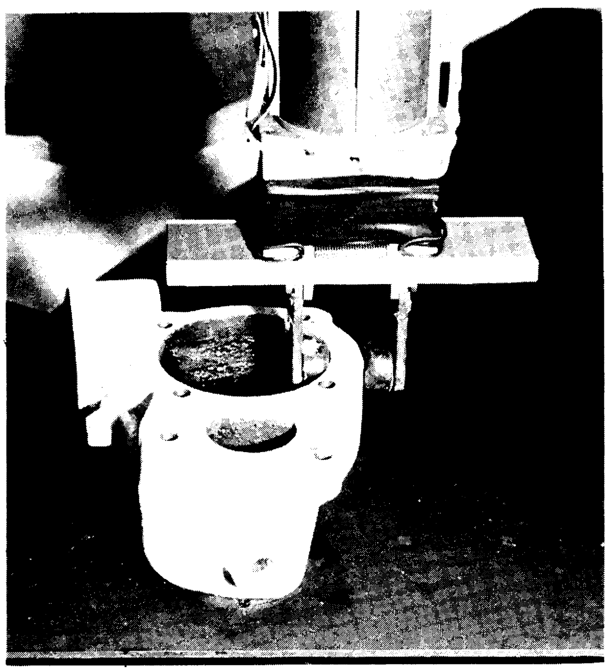
A. Hand Poised around the Pin

B. Hand after Closing on the Pin

C. After Checking that there is Something between the fingers, the Hand Opens Again







Picture 13.
Cent wing on the Side of the Pump Base

A. Hand Poised over the Pump Base, Not Centered about the Side

B. Hand Still Not Centered, but the Fingers have Closed so that the Left Finger is Touching the Side

C. Hand Now Centered after Moving to the Left so the Right Finger also Touches the Side

CENTER can be combined with SAVE and RESTORE to refine the positional information of an object. A model of the object is setup containing a set of grasping points and their relative position (X,Y,Z) with respect to some reference point on the object. To improve the location information for that object, the hand is asked to CENTER on one (or a series) of these relative grasping points. For each point the displacement of the object (i.e. the displacement of the reference point of the object) is SAVED. Each CENTER operation can only detect the displacement in one direction. Thus, two orthogonal CENTER's can be used to produce a 2-dimensional correction, etc. When a series of CENTER's are used, a RESTORE before each CENTER is used in order to make use of the latest information about the position of the reference point.

This type of two-dimensional localization with respect to a reference point was used to determine the final grasping position of the pump base, Vision determined. its position within one-third inch in X and one-fourth inch in Y. But this localization was not sufficient for the hand to pick up the base because of the base's irregular shape and the limited opening of the hand. Since the pump base was glued to a base plate, it was known to be upright. Thus, the Z component of the position was known quite accurately. The X and Y components needed to be improved. To do this a model of the base was set up as follows:

(1) the reference point was the center of the large hole

(2) the reference orientation was along the line of centers between the two large holes on top,

(3) two grasping directions and points (points A and B) were determined at right angles to each other.

Therefore, to determine the displacement in A's direction the hand was CENTERED on point A and any discrepancy was SAVED. This discrepancy was RESTORED when the hand CENTERED on point B. The combination of these displacements determines the pump base's X-Y displacement.

SAVE's and RESTORE's have been mentioned with respect to (1) remembering a specific point, such as a screw hole and (2) localizing an object, such as the pump base. These ideas can be combined in a straightforward way to provide dynamic position information for the pump base (with respect to its planned position). This is necessary because the base may not have been placed exactly in the aligning blocks or the hand may have moved it when trying to pull the screwdriver out of a screw. All that is needed to obtain this dynamic information is a model of the base which locates the screw holes with respect to the reference point. Each time a screw hole is found (inserting a pin, screwing a screw, etc.) the refined position can be SAVED. Thus finding one screw hole can help in finding all of the other parts on the base.

CONCLUSION

We have taken the first, primitive step toward integrating different types of sensory feedback into 8 general purpose, computer-cant rolled assembly system. We believe that this type of interaction, in some extended form, is necessary for performing sophisticated assembly tasks. The key factor in the applicability of this type of device is the ease with which it can be programmed. It will be important in the future to interact with design data bases in order to specify positions and motion const raint s automat ically. This will relieve the programmer from the task of defining the positions by "programming by doing," and will in effect generate a first cut at an assembly program.

We are currently designing a more powerful control language and are investigating tasks involving the coordination of two arms.

BIBLIOGRAPHY

[Dewar] R.Dewar, N.R.Lewis, L.Rossol, J.T.Olsztyn, "An Application of Computer Vision to Automatic Wheel Mounting," The First International Joint Conference on Pattern Recognition, Oct 1973.

[Feldman] J.A. Feldman, R.F. Sproul, "System Support for the Stanford Hand-Eye System," Second International Joint Conference on Artificial Intelligence, London, September 1971.

[Gill] A, Gill, "Visual Feedback and Related Problems in Computer Controlled Hand-Eye Coordination," Stanford Artificial Intelligence Memo 178, October 1972.

[McCarthy] John McCarthy, Arthur Samuel and Artificial Intelligence Project staff, Edward Feigenbaum and Heuristic Programming Project staff, "Project Technical Report," Stanford Artificial Intelligence Memo 143, March 1971.

[Paul] R.P.C.Paul, "Modelling, 'Trajectory Calculation and Servoing of a Computer Controlled Arm," Stanford Artificial Intelligence Memo 177, March 1973.

[Scheinman] V. D. Scheinman, "Design of a Computer Manipulator", Stanford Artificial Intelligence Memo 92, June 1969.

[Sobel] Irwin Sobel, "Camera Models and Machine Perception," Stanford Artificial Intelligence Memo 121, May 1970.