MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Memo No. 308

August 1974

FORCE FEEDBACK IN PRECISE ASSEMBLY TASKS

Hirochika Inoue

ABSTRACT

This paper describes the execution of precise assembly tasks by a robot. The level of performance of the experimental system allows such basic actions as putting a peg into a hole, screwing a nut on a bolt, and picking up a thin piece from a flat table. The tolerance achieved in the experiments was 0.001 inch. The experiments proved that force feedback enabled the reliable assembly of a bearing complex consisting of eight parts with close tolerances. A movie of the demonstration is available.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contracts N00014-70-A-0362-0005

1. Introduction

Many industries have a great potential demand for advanced automation in assembly lines. In many modern factories, a great deal of unskilled human labor is forced to be mated with substantially automated fabrication machines or N/C machine tools because no versatile assembly machine is yet available. From the viewpoint of robotics research, advanced assembly automation provides a fertile application field.

So far, several laboratories have developed primitive robots that can carry out simple assembly tasks. However, almost none of these robots operate within practical precision ranges. An exception is a machine that was developed at the Central Research Laboratory of Hitachi. It is not a general purpose robot, but the use of force feedback allows the machine to perform an assembly task with very close tolerances faster than unskilled human labor.

Force feedback is a key to the performance of precise assembly tasks. Fortunately, at the MIT A.I. Laboratory, there is available a simple robot that can detect forces and torques acting on its wrist. Only small modifications of this system were needed to begin a basic experimental study.

This is a report of a study on the execution of precise assembly tasks by a robot. The level of performance of the experimental system allows such basic actions as putting a peg into a hole, screwing a nut on a bolt, and picking up a thin piece, such as a washer that is lying on a flat table. Not only they are good examples for the study of force feedback, but they also are an important part of a repertoire of basic tasks required in a wide range of practical machine assemblies. The tolerance achieved in the experiments was 0.001 inch. Presumably, this sort of tolerance is a minimum requirement for many practical applications, since it corresponds to the resolution of a vernier caliper, one of the common standards of measurement in many machine shops. This study demonstrates an assembly of a bearing complex consisting of eight parts with close tolerances, of up to 0.001 inch. These experiments show that force feedback enables the reliable performance of assembly tasks with close tolerances.

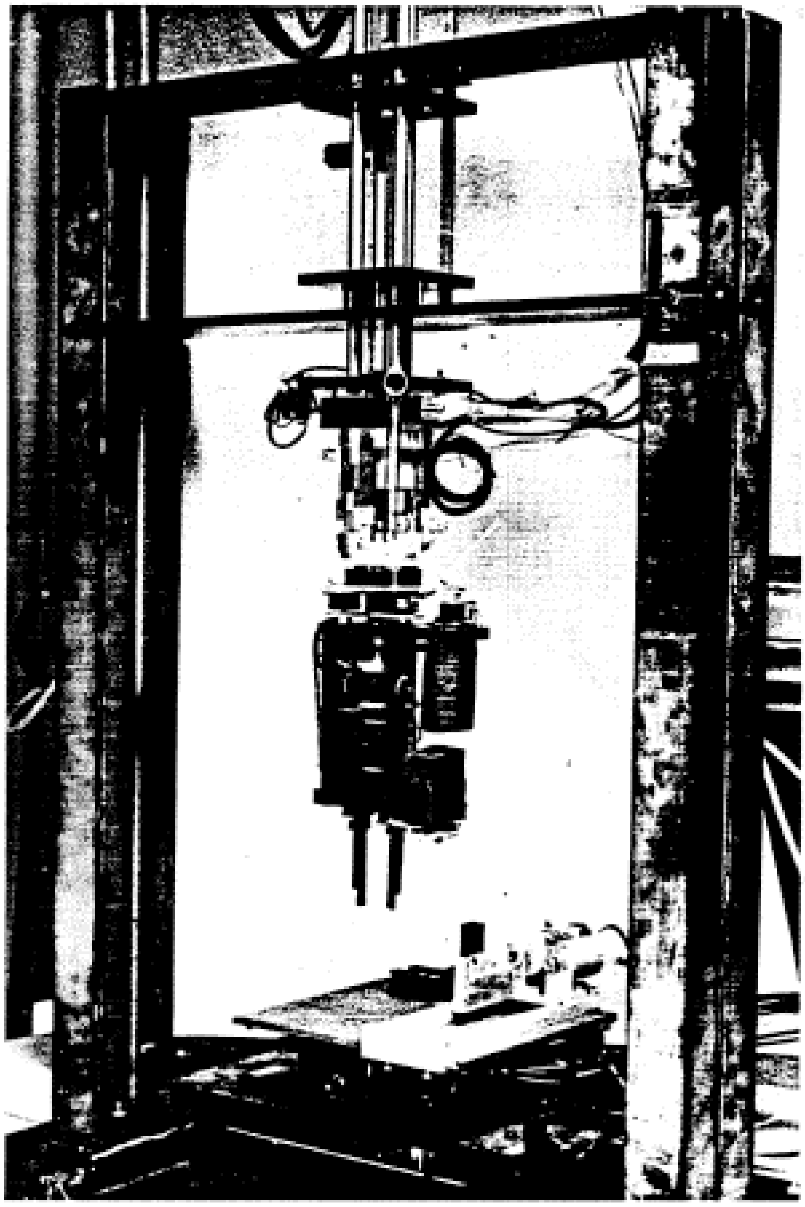


Figure 1. Whole view of the Little Robot System.

2. The Little Robot System

This study employed the Little Robot System developed by D. Silver. It is shown in figure 1. It provides for the ITS user a medium size, five degree of freedom, seven axis robot which is controlled through the programming language LISP by the PDP-6 computer.

The heart of the robot is a force sensor complex located at its wrist. It consists of six L.V.D.T.s (Linear Variable Differential Transformers), and it allows the measurement of forces and torques that are acting on the wrist. Figure 2 shows the geometrical arrangement of the six L.V.D.T.s, where angle "a" is a rotational displacement between the robot axis and the force sensor complex axis. The relationship between force and torque components in the w-xyz coordinate system and the force components acting at the position of each L.V.D.T. is described by the following equations.

Fx = -f2*cos(45+a) - f4*cos(45-a) + f5*cos(45+a)

Fy = -f2*sin(45+a) + f4*sin(45-a) + f5*sin(45+a)

Fz = f1 + f3 + f6

Tx = f1*d*cos(a) - f3*d*sin(a) + f6*d*sin(a)

Ty = f1*d*sin(a) + f3*d*cos(a) - f6*d*cos(a)

Tz = f2*d - f4*d + f5*d

where, Fx, Fy, Fz	forces in x, y, z axis
Tx, Ty, Tz	torques about x, y, z axis
f1, f2,,,, f6	forces at each L.V.D.T.

Force and torque components in the hand coordinates are obtained in turn from the following equations.

$Fx' = Fx*\cos(b) - Fy*\sin(b)$

$Fy' = Fx*\sin(b) + Fy*\cos(b)$

$Fz' = Fz$

$Tx' = (Tx - L2*Fy)*\cos(b) - (Ty + L2*Fx)*\sin(b)$

$Ty' = (Tx - L2*Fy)*\sin(b) + (Ty + L2*Fx)*\cos(b) + L1*Fz$

$Tz' = Tz - L1 * (Fy*\cos(b) + Fx*\sin(b))$

In the current Little Robot System, an approximation technique is used to calculate the forces and torques, since the stress-strain matrix of the sensor has not been analysed in detail. Consider small displacements (dx", dy", dz") in x", y", z" axis as well as rotations (da", db", dc") about the x", y", z" axis. The displacements in the L.V.D.T.s (t1, t2,... . t6) are related to dx", dy", dz" and da", db", dc" as follows:

$dx" = -(t2 + t4)/ \sqrt{2}$

$dy" = (t4 + t5)/ \sqrt{2}$

$dz" = (t3 + t6)/ 2$

$da" = (t3 + t6 - 2*t1)/ 2$

$db" = (t3 - t6)/ 2$

$dc" = (t2 + t5)/ 2$

By the coordinate transformation from the sensor axes (x", y", z") to the robot axes (x, y, z), we can obtain the displacements (dx, dy, dz) in x, y, z axis as well as the rotations (da, db, dc) about the x, y, z axis. Next, we must determine overall stiffness (k1, k2,... k6) in x, y, z axis and about the x, y, z axis. Then we can get force and torque components approximately, as follows.

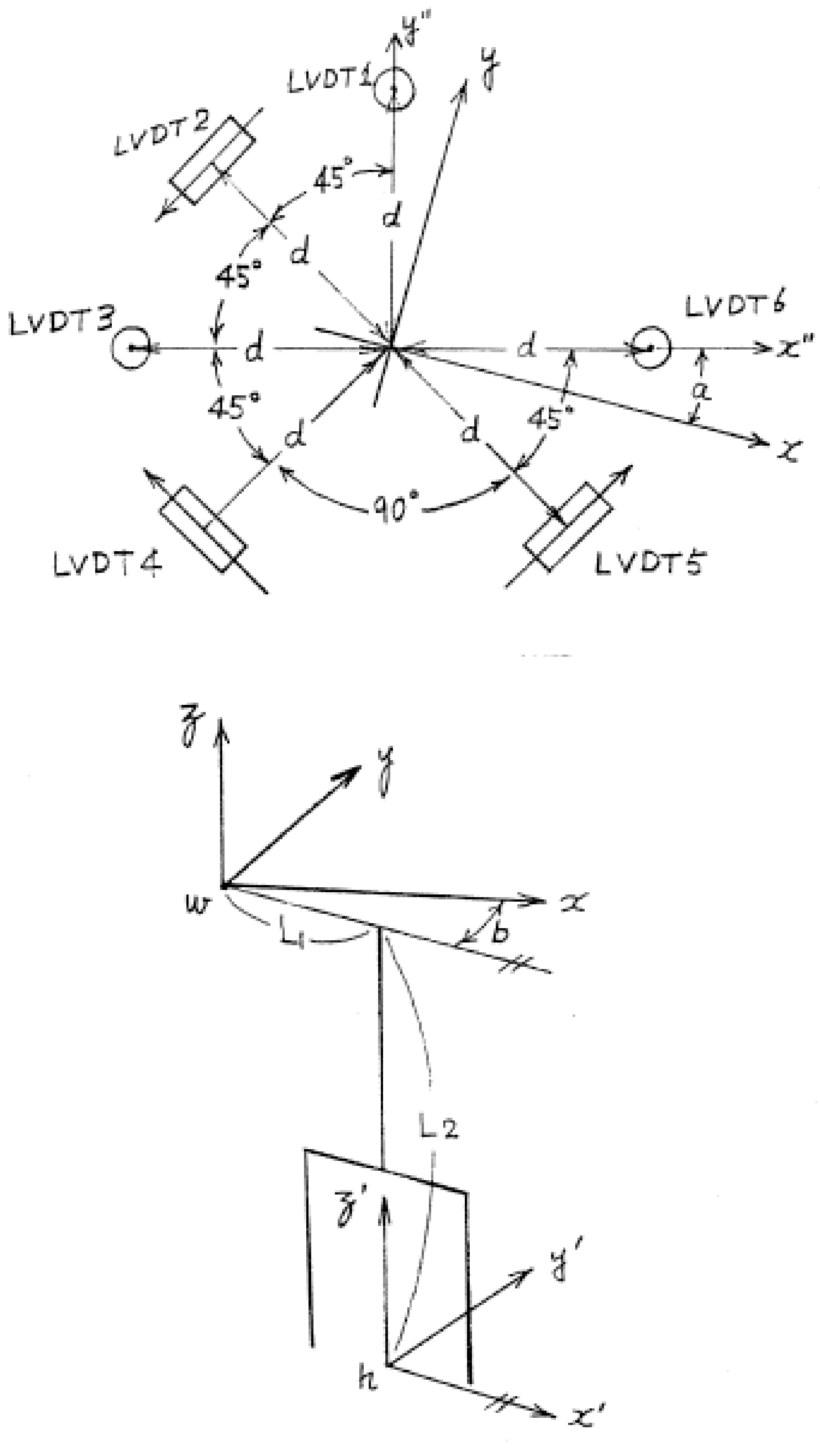


Figure 2.   Force sensor complex and the coordinate systems.

$$F_x = k1 * dx , \quad F_y = k2 * dy , \quad F_z = k3 * dz$$

$$T_x = k4 * da , \quad T_y = k5 * db , \quad T_z = k6 * dc$$

Servoing of the robot is done by the PDP-6 computer every 1/60 second. LISP in the PDP-10 governs the PDP-6 via five special LISP functions.

The LISP functions are:

SETM (SET Mini) of two arguments, sets PDP-6 variables.

GETMF (GET Mini Floating point) of one argument, reads PDP-6 variables in floating point mode.

GETM (GET Mini fixed point) reads PDP-6 variables in fixed point mode.

WAIT of one argument, evaluates its argument every 1/30 second and will "sleep" until evaluation returns T.

SEQ (Sort of EQual) of three arguments, returns T when the first argument is equal to the second within the tolerance of the third, otherwise NIL.

The PDP-6 variables are shown in Table 1. Note that the prefixes X, Y, Z, R, G, V, T mean X-axis Y-axis, Z-axis, Rotation about Z, Grip, Vice and Tilting-part-holder, respectively. Mode switches FRE select the control mode. For example, when XFRE is 0, the X-axis acts as a position control servo, and when XFRE is -1, it acts as a force control servo. A more detailed description can be found in AI Memo No. 273: "The Little Robot System." by D. Silver.

	suffix	prefix	meaning	unit	type
I	POS	X,Y,Z,R,G,V,T	current position	inch/radian	floating
N	DLT	X,Y,Z,R,G,V,T	position error	pot. unit	fixed
P	FRS	X,Y,Z, G,V	calibrated force	LVDT unit	fixed
U	TRC	X,Y,Z	calibrated torque	LVDT unit	fixed
T					
O	DES	X,Y,Z,R,G,V,T	position destimation	inch/radian	floating
U	FOS	X,Y,Z,R,G,V	force destination	LVDT unit	fixed
T	FRE	X,Y,Z,R,G,V	mode switch		0 or -1
P	GAN	X,Y,Z,R,G,V,T	position servo gain		fixed
U	FGN	X,Y,Z,R,G,V	force servo gain		fixed
T					

Table 1. PDP-6 Variables

What follows is a brief introduction to the basic functions used in this study. They are intended to help the reader understand the program which will apppear in sections 3 and 4.

Position control (absolute); X=, Y=, Z=, R=, G=, V=, P=

eg: `(DEFUN X= (X)   (SETM XDES X) (SETM XFRE 0))`

Position control (relative); DX=, DY=, DZ=, DR=, DG=, DV=, DP=

eg: `(DEFUN DX= (DX) (X= (PLUS (GETMF XPOS) DX)))`

Force control; FX=, FY=, FZ=, FR=, FG=, FV=

eg: `(DEFUN FX= (FX) (SETM XFOS FX) (SETM XFRE -1))`

Check of position control; ?X, ?Y, ?Z, ?R, ?G, ?V, ?P

eg: `(DEFUN ?X ( )  (SEQ (GETM XDLT) 0 Threshold-P))`

Check of force control; ?FX, ?FY, ?FZ, ?FR, ?FG, ?FV

eg: `(DEFUN ?FX ( )  (SEQ (GETM XFRS) (GETM XFOS) Threshold-F))`

Miscellaneous; XOF, YOF, ZOF, DOF, and HOF gets X, Y, Z coordinate, diameter and height of the object specified by the argument,respectively.

Position control functions as well as force control functions deliver their arguments to the PDP-6, when they are evaluated. From that moment, the servoing loop in the PDP-6 controls the robot to the specified destination in the specified mode, until another function changes the control. The function WAIT checks a control situation and waits until its argument evaluates to T. The servoing is still in effect after WAIT is terminated.

For example,

```
    (FZ= fz)(WAIT '(?FZ))  (DR= 3.14)(WAIT '(?R))
```

This program moves the hand down until it lands on the table, i.e, a force is felt in the Z-direction. When (WAIT '(?FZ)) returns control, the program proceeds, and rotates the hand 3.14 radians. During and even after this rotation, the hand keeps the contact force "fz" against the table, because (FZ= fz) is still in effect. This should be kept in mind, as the reader examines the programs in sections 3 and 4.

3. Applications of Force Feedback in Machine Assembly

This section describes how force feedback techniques are applied to the process of precise machine assembly. Three basic actions are selected for study. They are;

(1) putting a peg into a hole with close tolerance.

(2) screwing a nut on a bolt.

(3) picking up a thin piece from a flat table.

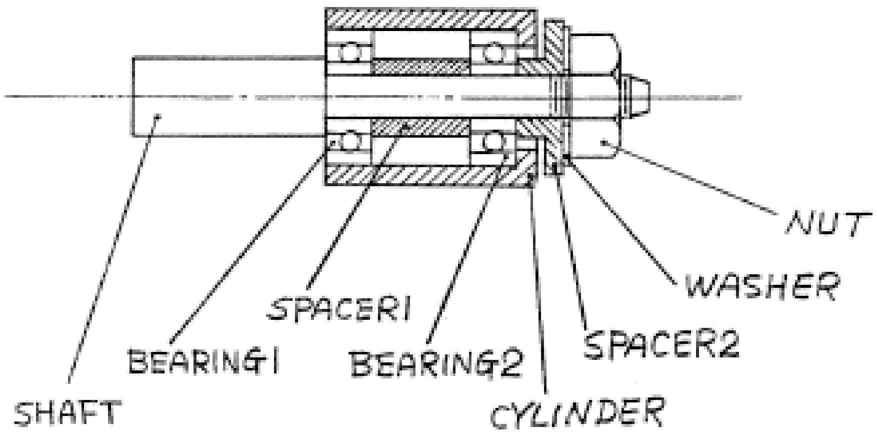
Each of these is required to perform a wide variety of machine assemblies. If one analyzes the assembly of any machine, one can easily find that the above-mentioned actions are frequently needed.

The object used in the experimental assembly task is a specially designed bearing complex, shown in figure 3. It consists of two radial ball bearings, two spacers, a cylinder, a washer, a nut and a shaft with threaded end. Important dimensions of the parts are specified in figure 4. The tolerances are 0.001 inch for the SHAFT-BEARING pair and the BEARING-CYLINDER pair, 0.002 inch for the SHAFT-SPACER1 pair, 0.01 inch for the SHAFT-SPACER2 pair, and 0.06 inch for the SHAFT-WASHER pair.

3.1 Peg-into-Hole Assembly.

The main procedure of the peg-into-hole assembly is broken into three consecutive phases: they are "drop-into", "mate" and "push-into", in that order.

"Drop-into" is a phase to partially drop a shaft into the mouth of the cylinder. In the next phase "mate" adjusts the relative position



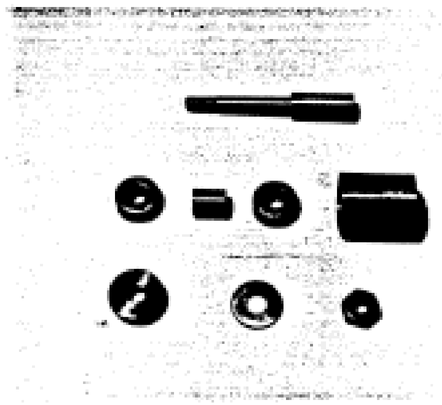
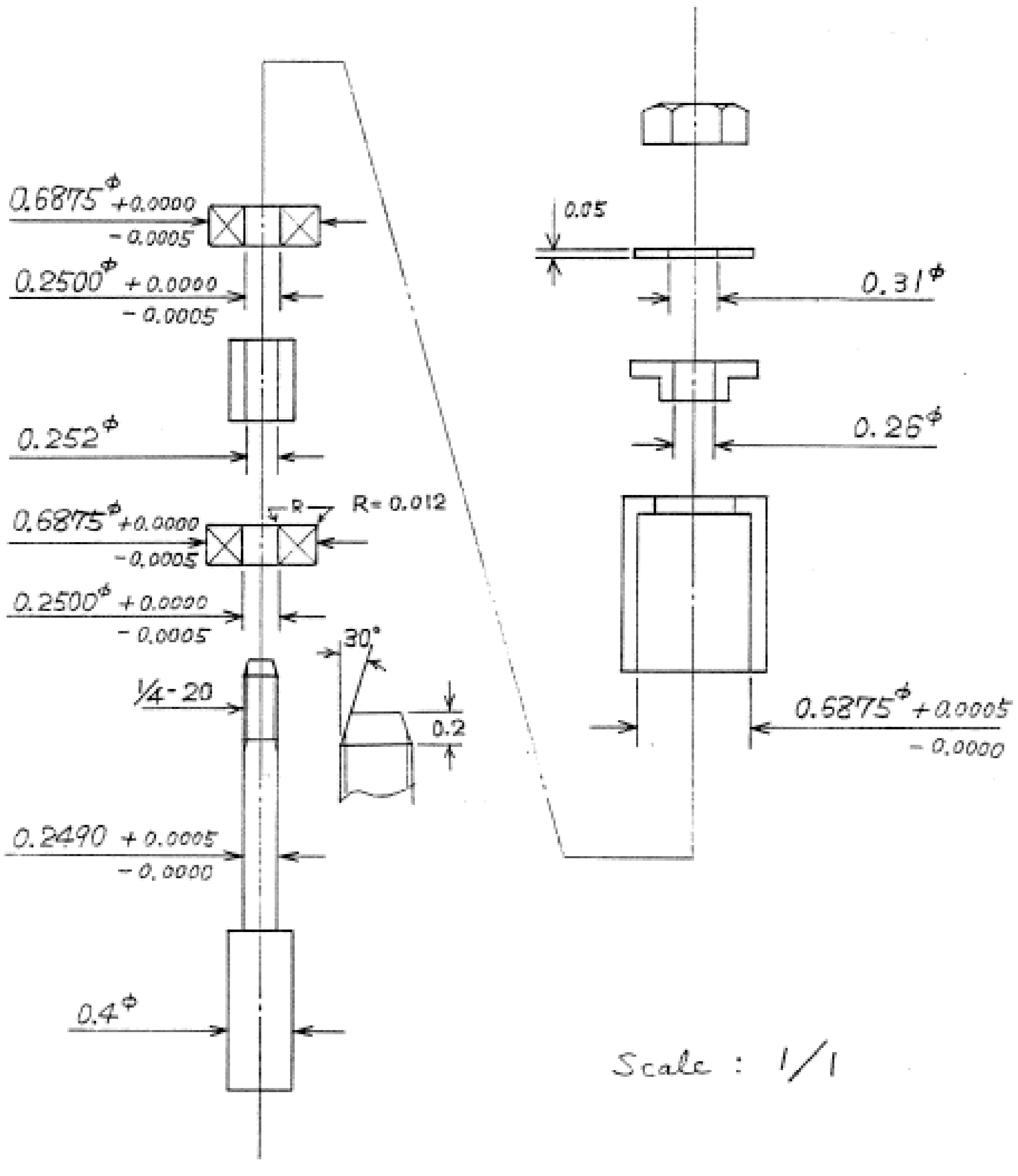


Figure 3: The Object to be assembled.



0.6875$^{\phi}$ +0.0000 −0.0005
0.2500$^{\phi}$ +0.0000 −0.0005
0.252$^{\phi}$
0.6875$^{\phi}$ +0.0000 −0.0005
R R R= 0.012
0.2500$^{\phi}$ +0.0000 −0.0005
1/4-20
30°
0.2
0.2490 +0.0005 −0.0000
0.4$^{\phi}$
0.05
0.31$^{\phi}$
0.26$^{\phi}$
0.5275$^{\phi}$ +0.0005 −0.0000

Scale : 1/1

Figure 4. Size and clearance of the parts.

and orientation between the two, until the shaft sits in the mouth of cylinder correctly. In the third phase "push-into", the shaft is pushed into the cylinder smoothly until it arrives at the bottom. Every phase requires some type of force feedback to control the delicate physical relationships between the parts.

The clearance and the shoulder shape determine the difficulty of assembly. In the field of machine design, the standards of fit are well defined, depending on the machining accuracy, however we do not yet have any standards of fit in automatic assembly. Presumably, they should be defined by considering positioning error, clearance between parts, shoulder shape and assembly algorithm. In this study, the concept of "loose fit" and "close fit" are introduced, and each fit is related to its appropriate stereotyped assembly procedure.

3.1.1 Loose Fit

An example of "loose fit" is the SHAFT-SPACER2 pair. The SHAFT is tapered at one end, and the clearance is 0.01 inch at the body and 0.068 inch at the tapered end. This assembly task begins from the initial state in which SPACER2 is located above the SHAFT in the same orientation as the SHAFT.

At first, we must consider the positioning repeatability of the hand. Suppose the errors in X, Y are " $\pm e/2$ ". Then, the hand must be located within the ($e * e$) error square enclosing the reference point. If the hand is holding SPACER2 at its center, SPACER2 is also located within the same error square. On the other hand, the tolerance defines a circle whose diameter is equal to "c". If the tolerance circle completely covers the error square, the fit is so loose that the SHAFT will go into the SPACER2 without changing their initial orientations. Thus, the condition of "loose fit" is defined by $c > \sqrt{2} * e$

In actual experiments, we cannot assume that the hand always holds SPACER2 at its center. One usually finds a small offset between the hand and SPACER2, especially with a parallel jaw type hand. The offset is mainly the result of the grasping action and sometimes it is much larger than the positioning error of the servo-mechanisms. If a hand has parallel jaws, like the one in the Little Robot System, the offset occurs only in the direction perpendicular to the squeezing axis.

(a) DROP-INTO: From the initial position, without changing the orientation, SPACER2 is moved down until it lands on SHAFT. There are three possibilities when SPACER2 is brought into contact with SHAFT. Cases 1, 2, and 3 of figure 5 occur when the offset is negative, nearly zero and positive, repectively. To guide the SHAFT into SPACER2 , a different search tactics must be taken depending on the contact situation, and it is neccessary to decide which case has in fact occurred. However, if a small shift " +dX " has been added to the initial position in advance, the contact situations become as in the cases 1', 2' and 3'. This time, the same simple search in the "-X" direction will be applicable in all three cases.

Thus the procedure is as follows:

(1) make a small advance shift toward "+X" .

(2) move SPACER2 down until it lands on SHAFT.
and keep a small force against it.

(3) slide SHAFT toward "-X" until it drops into SPACER2.

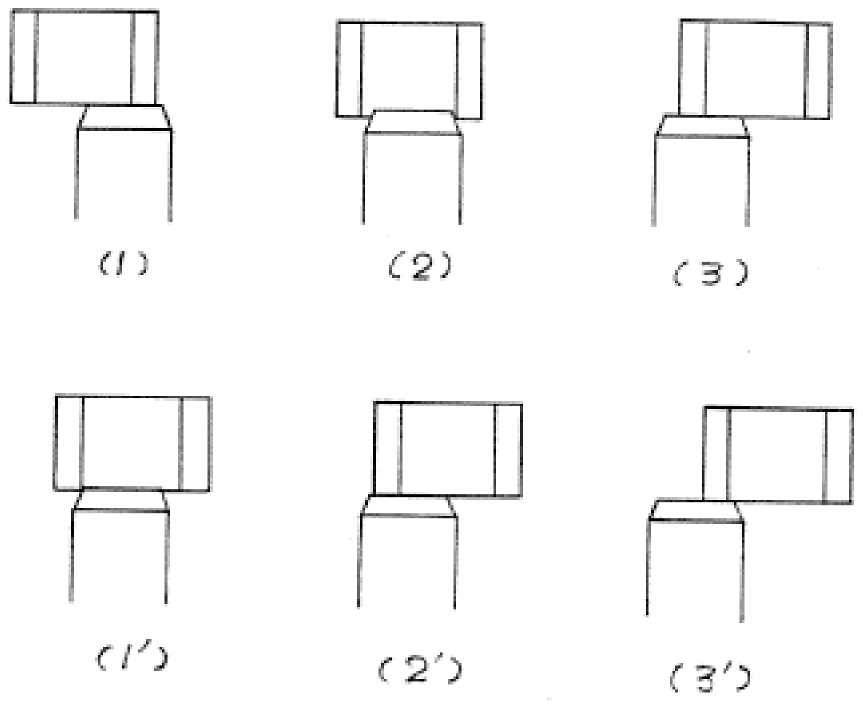


Figure 5. Landing situations.

(4) stop Z-motion.

The program for this procedure is:

```
     (DEFUN DROP-INTO-L ( )
            (DX= shift) (WAIT ' (?X))
            (FZ= small-contact-force)(WAIT '(?FZ))
            (FX= small-sliding-force)
            (WAIT '(OR (?FX) (SEQ (GETM FZ) 0 Threshold)))
            (DZ= 0.0)   )
```

Note; (WAIT '(?FX)) waits until the SHAFT contacts the left wall of the hole. (WAIT '(SEQ (GETM FZ) 0 Threshold)) checks the occurrence of the "drop". Either event will complete the desired action.

(b) MATE: When DROP-INTO has been completed, SHAFT should be at least partially in the hole of SPACER2. So, the following procedure can guide the position of SHAFT exactly to the center of SPACER2.

(1) move SHAFT toward "+Y" until it finds an edge of SPACER2

and get the Y-position of SHAFT ==> Y1

(2) move SHAFT toward "-Y" until it finds another edge of SPACER2

and get the Y-position of SHAFT ==> Y2

(3) move SHAFT to (Y1 + Y2)/ 2

this motion locates the SHAFT in the actual center along the Y axis.

similarly;

(4) find "+X" edge ==> X1

(5) find "-X" edge ==> X2

(6) move SHAFT to (X1 + X2)/ 2

What follows is a program that performs this procedure.

```
(DEFUN MATE-V ( )
     (PROG (X1 Y1)
       (FY= small-edge-finding-force-to-"+Y") (WAIT '(?FY))
       (SETQ Y1 (GETMF YPOS))
       (FY= small-edge-finding-force-to-"-Y") (WAIT '(?FY))
       (Y= (//$ (+$ Y1 (GETMF YPOS)) 2.0)) (WAIT '(?Y))
       (FX= small-edge-finding-force-to-"+X") (WAIT '(?FX))
       (SETQ X1 (GETMF XPOS))
       (FX= small-edge-finding-force-to-"-X") (WAIT '(?FX))
       (X=  (//$ (+$ X1 (GETMF XPOS)) 2.0)) (WAIT '(?X)) ))
```

(c) PUSH-INTO-V: SPACER2 is moved down along the SHAFT until it arrives at the end. During this process, the X and Y position of the SHAFT should accommodate to that of SPACER2. This is easily done by controlling X and Y forces to be zero. The program for this is:

```
(DEFUN PUSH-INTO-V ( )
       (FX= 0) (FY= 0) (FZ= inserting-force) (WAIT '(?FZ)) )
```

The sequence of DROP-INTO-L, MATE-V, and PUSH-INTO-V gives a simple stereotyped action for the peg-into-hole assembly with loose fit.

```
(DEFUN PEG-HOLE-L ( )
       (DROP-INTO-L) (MATE-V) (PUSH-INTO-V) )
```

Actually, we can omit the function MATE-V, because during the following function PUSH-INTO-V, the X, Y position of the SHAFT adapts to the center of SPACER2. Thus we have an even simpler stereotyped action for this task.

```
(DEFUN PEG-HOLE-L2 ( ) (DROP-INTO-L) (PUSH-INTO-V) )
```

3.1.2 Close Fit.

The other class of fit is "close fit", defined by ($c < \sqrt{2} * e$). The BEARING-CYLINDER pair, shown in figure 7, is a good example of this. The function DROP-INTO-L will not reliably perform the drop-into process of a pair with close fit, because the tolerance circle does not cover

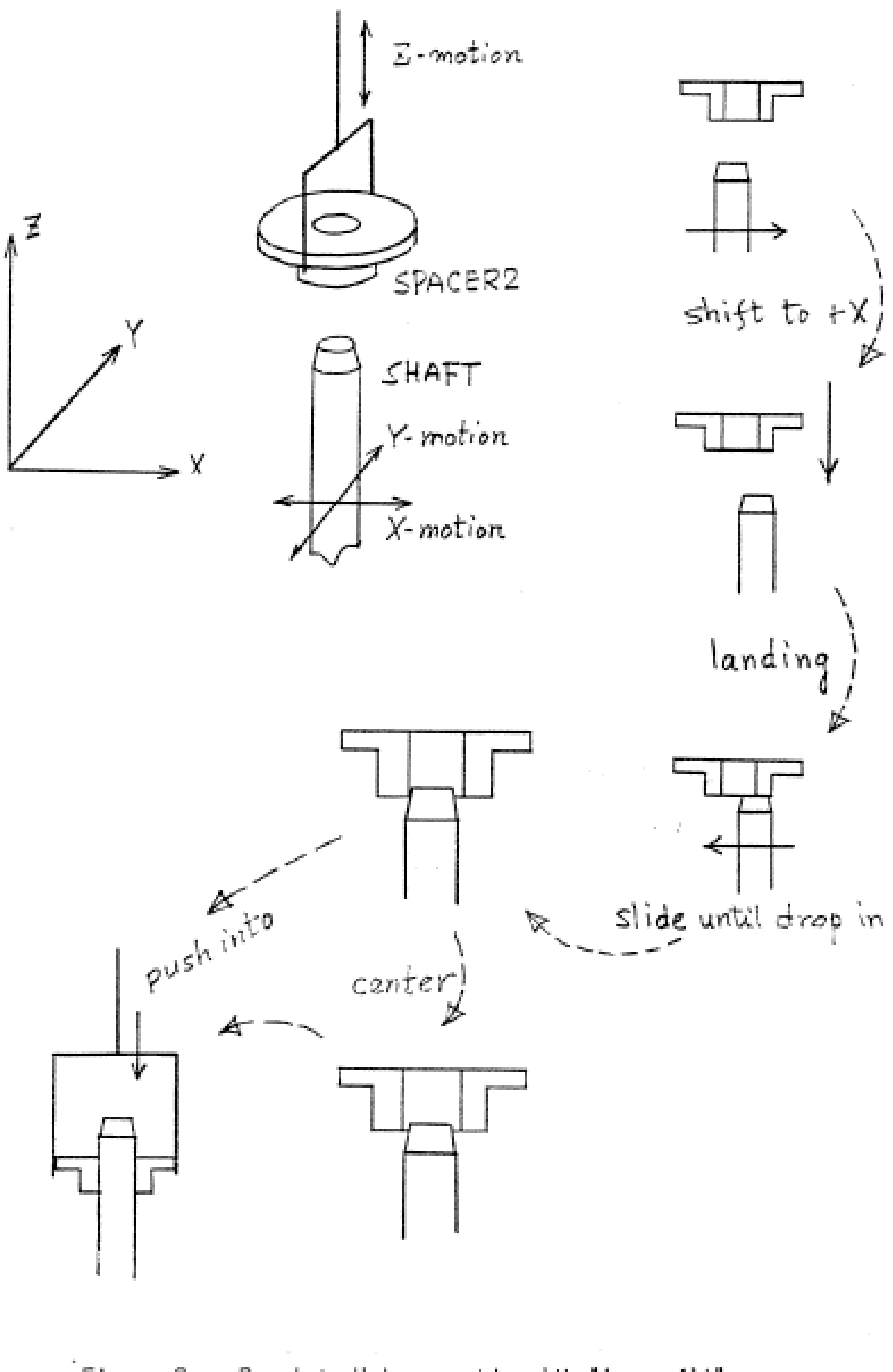


Figure 6.   Peg-into-Hole assembly with "loose fit".

all of the error square. The simplest technique for solving this difficulty is that of tilting the peg or the hole. By tilting, the tolerance circle is equivalently enlarged up to the size of hole, which is much larger than the error square. For example, the diameter of the tolerance circle of the BEARING—CYLINDER pair is 0.001 inch without tilting, and 0.688 inch with tilting. Note that $\sqrt{2}$ *e of the system is about 0.009 inch. So, if we apply the tilting technique to the pair with close fit, we can treat it as if it were a loose fit. The following is a program to do the drop-into process for a pair with close fit.

```
(DEFUN DROP-INTO-C ( )
       (DR= 0.1) (DY= shift) (WAIT '(AND (?R) (?Y))
       (FX= landing-force) (WAIT '(?FX))
       (DX= 0.0) )
```

After the DROP-INTO-C is completed, CYLINDER must be aligned to the orientation of SHAFT and Y, Z position of BEARING must be adapted to CYLINDER. The procedure is:

(1) move BEARINGs toward "+Z" until it finds an edge of CYLINDER, and get the Z-position ==> Z1

(3) move BEARINGs toward "-Z" until it finds another edge of CYLINDER, and get the Z-position ==> Z2

(3) move to (Z1 + Z2) / 2

(4) move BEARINGs to +Y direction until it contacts the CYLINDER

(5) keep above contact and apply small force in +X direction

(6) null the Z-force and rotate CYLINDER to R = 0.0

The program for this is:

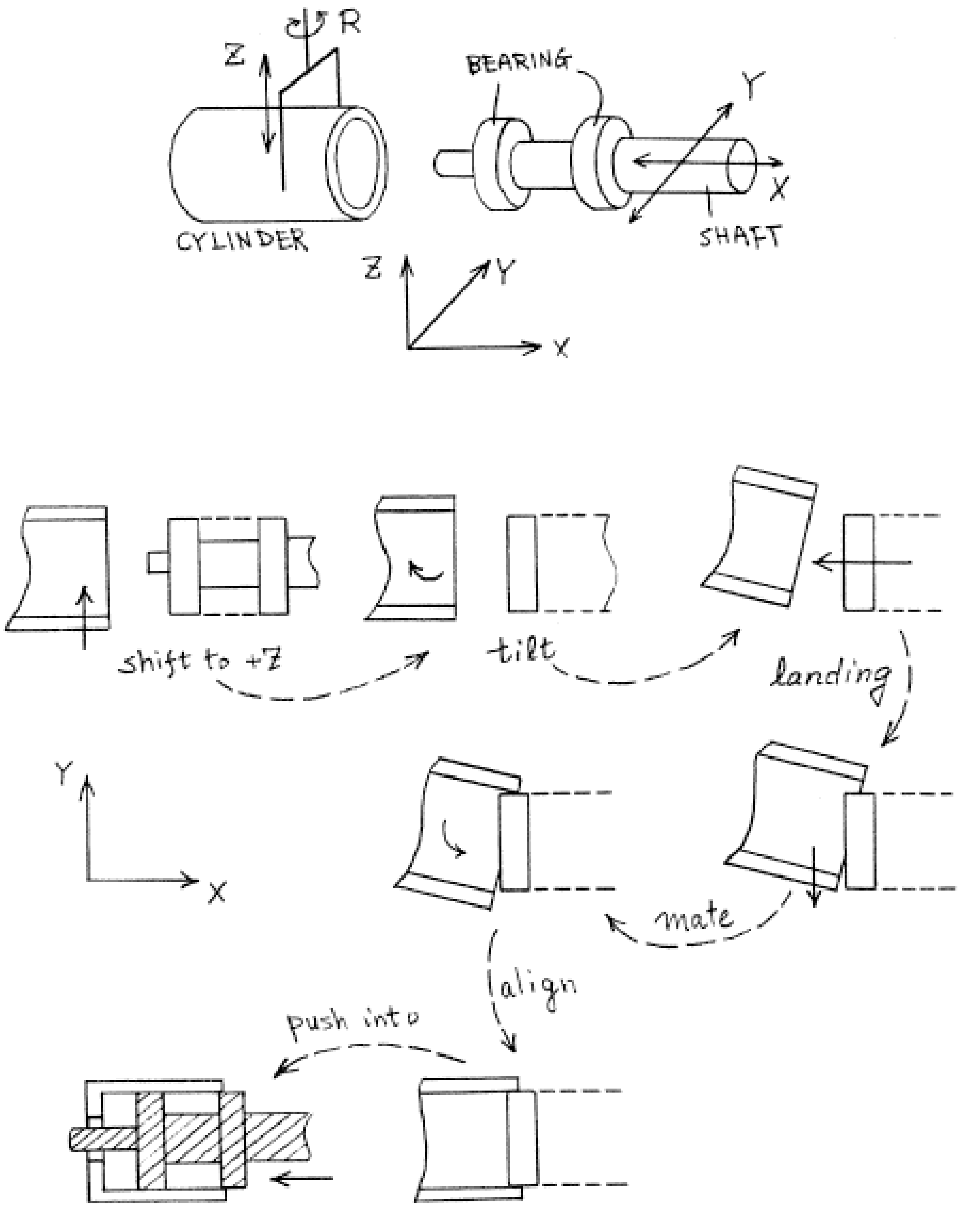


Figure 7. Peg-into-Hole assembly with "close fit".

```
(DEFUN MATE-H ( )
     (PROG (Z1)
       (FZ= small-edge-finding-force-to-"+Z") (WAIT '(?FZ))
       (SETQ Z1 (GETMF ZPOS))
       (FZ= small-edge-finding-force-to-"-Z") (WAIT '(?FZ))
       (Z= (//$ (+$ Z1 (GETMF ZPOS)) 2.0))
       (FY= small-contact-force-in-"Y-axis") (WAIT '(?FY))
       (FZ= 0)  (R= 0.0) (WAIT '(?R))
       ))
```

The next thing to do is the "push-into" action. It is the same as PUSH-INTO-Y except for the direction.

```
(DEFUN PUSH-INTO-H ( )
       (FY= 0) (FZ= 0)
       (FX= inserting-force) (WAIT '(?FX)) )
```

A stereotyped sequence of peg-into-hole assembly for close fit is defined as follows.

```
(DEFUN PEG-HOLE-C ( )
       (DROP-INTO-C) (MATE-H) (PUSH-INTO-H) )
```

3.2 Screwing a Nut.

The action for screwing a nut on a bolt is also broken into three consecutive phases; "drop-into", "mate", and "screw-into". For the "drop-into" and "mate" processes, DROP-INTO-L and MATE-V, defined in section 3.1, are applicable. In order to screw a nut, the nut must be turned in a clockwise direction while exerting a small downward pressure. During this screwing phase, the X, Y position of the nut must accommodate to that of the screw, so X, Y force are controlled to be zero. When the robot feels the specified fastening torque, it stops the screwing motion.

```
(DEFUN SCREW-INTO ( )
       (FZ= small-down-force) (FX= 0) (FY= 0)
       (FR= fastening-torque) (WAIT '(?FR)) )
```

Thus, a procedure for screwing a nut is:

```
(DEFUN SCREW-NUT ( )
       (DROP-INTO-L) (MATE-V) (SCREW-INTO) )
```

Again in this function, we can also ignore the function MATE-V, because during the next function SCREW-INTO, the X, Y position of the screw adapts to the center of the nut. Thus we have an alternate, simpler function for this task.

```
(DEFUN SCREW-NUT-2 ( )
       (DROP-INTO-L) (SCREW-INTO) )
```

3.3 Picking Up a Thin Piece.

If one analyzes the assembly of any machine, one will find that the robot has to handle many small, thin pieces such as washers. It is hard to pick up such thin objects from a flat table. If the commanded position of the hand is slightly higher than the table, the robot will miss the washer, if it is a little bit lower than the table, it may exert a large force against the table. Presumably, in the latter case serious damage can occur to the robot or the table.

When picking up a thin piece, force feedback is necessary to check whether the hand makes contact with the table or not. The following is a program to pick up a thin piece on a table.

```
(DEFUN PICKUP (IT)
       (G= (+$ (DIA IT) 0.2))
       (Z= (+$ (ZOF IT) (HOF IT) 0.2) (WAIT '(AND (?Z) (?G)))
       (FZ= small-landing-force) (WAIT '(?FZ))
       (FG= grasping-force) (WAIT '(?FG)) )
```

In order to decrease the offset between the hand and the object and to allow for errors in the positioning of the part, the following centering action is very effective.

```
      (DEFUN PICKUP-2 (IT)
              (G= (+$ (DIA IT) 0.2))
              (Z= (+$ (ZOF IT) (HOF IT) 0.2)) (WAIT '(AND (?Z) (?G)))
              (FZ= small-landing-force) (WAIT '(?FZ))
              (FG= small-grasping-force) (WAIT '(?FG))
              (DG= 0.2) (WAIT '(?G))      ;open the hand 0.2 inch.
              (DR= 1.57) (WAIT '(?R))     ;turn the hand 1.57 radian.
              (FG= grasping-force) (WAIT '(?FG))  )
```

4. Outline of Assembly Demonstration

Figure 8-1 shows the initial environment of the demonstration of machine assembly. This demonstration does not use any vision program, so all information about the position of the parts must be specified. The assembly sequence is:

(1) put BEARING1 onto SHAFT

(2) put SPACER1 onto SHAFT

(3) put BEARING2 onto SHAFT

(4) assemble CYLINDER with BEARINGs in SHAFT

(5) put SPACER1 onto SHAFT, vertically

(6) put WASHER onto SHAFT, vertically

(7) put NUT on SCREW and torque down the NUT

To carry out the above demonstration, three LISP functions are defined. ASSY-L performs the peg-into-hole assembly of loose fit, as described in section 3.1.1. ASSY-C does the peg-into-hole assembly of close fit, as described in section 3.1.2. ASSY-N puts a NUT on the screw and torques it down, as described in section 3.2.

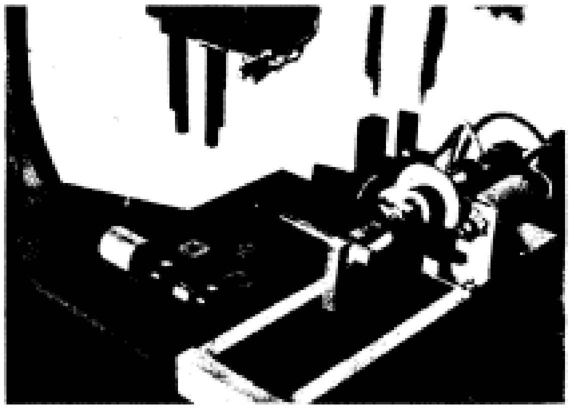
```
(DEFUN ASSY-L (PART1)
        (GO-ABOVE PART1)      ;move hand above the part
        (PICKUP-2 PART1)      ;pick it up
        (BRING-TO VSHAFT)     ;bring it to shaft
        (PEG-HOLE-L)          ;peg-into-hole of loose fit
        (RELEASE)  )          ;release


(DEFUN ASSY-C (PART1)
        (GO-ABOVE PART1)      ;move hand above the part
        (PICKUP PART1)        ;pick it up
        (BRING-TO HSHAFT)     ;bring it to shaft
        (PEG-HOLE-C)          ;peg-into hole of close fit
        (RELEASE)  )          ;release
```

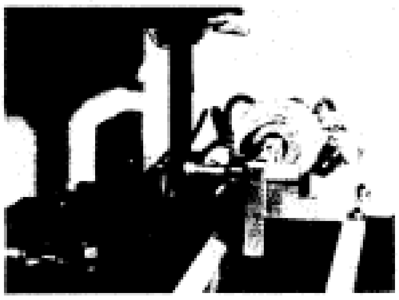
```
(DEFUN ASSY-N (NUT)
       (GO-ABOVE NUT)       ;move hand above the part
       (PICKUP-2 NUT)       ;pick it up
       (BRING-TO VSHAFT)    ;bring it to the screw
       (SCREW-NUT)          ;screw a nut
       (RELEASE)  )         ;release
```

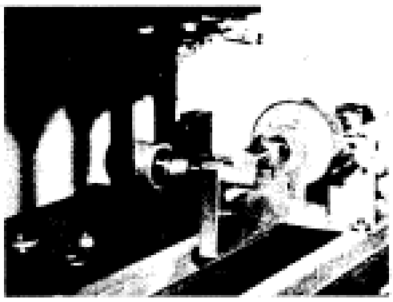
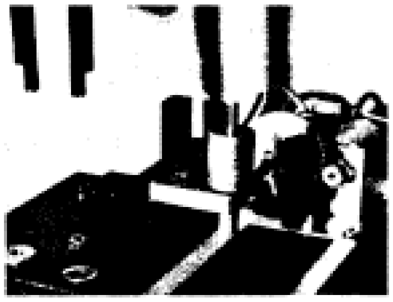
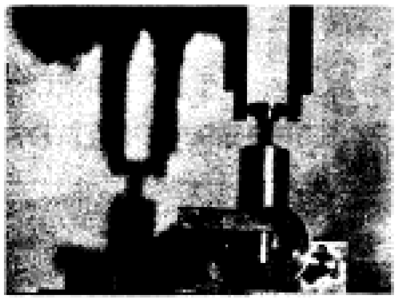
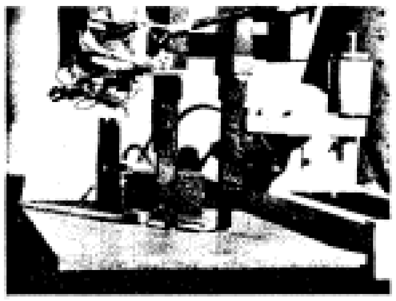
The following program ASSEMBLY carries out the demonstration completely.

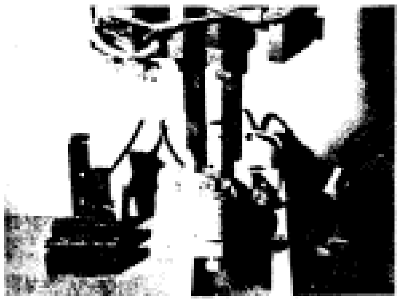
Pictures in figure 8 show the experiment.

```
(DEFUN ASSEMBLY ( )
       (P= 1.57) (WAIT '(?P)) ;tilt the shaft horizontally
       (ASSY-C BEARING1)
       (ASSY-C SPACER1)
       (ASSY-C BEARING2)
       (ASSY-C CYLINDER)
       (P= 3.14) (WAIT '(?P)) ;tilt the shaft vertically
       (ASSY-L SPACER1)
       (ASSY-L WASHER)
       (ASSY-N NUT) )
```







5. Conclusions and Discussion.

Mating parts with close tolerance requires force feedback to assure good alignment. This paper describes a study of force feedback in a typical assembly task. The performance of the system includes such tasks as putting a peg into a hole, screwing a nut on a bolt, and picking up thin pieces such as washers. It should be noted that each of these is a basic and required task for a wide range of machine assemblies. Tolerances of 0.001 inch were achieved and experiments proved that force feedback enabled the robot to perform these assembly tasks quite reliably.

In the peg-hole assembly, the concept of loose fit and close fit were introduced. For each fit, a simple stereotyped action is presented. the stereotype for close fit uses a simple tilting technique to improve the reliability. The closest tolerance pair (CYLINDER-BEARING) assembly was assembled using this tilting technique. However, the BEARING-SHAFT and SPACER1-SHAFT assembly could be done without the tilting technique. When screwing a nut, the nut is first brought into contact with the top of screw. From that moment, force feedback guides the nut into the screw and turns it down to the prescribed torque. Force feedback also allows the robot to pick up a thin piece, even from a flat table. In this process, after the hand is felt to have arrived at the table, the robot closes its hand while keeping the contact force with the table. By using this tactic, the robot can easily pick up a thin piece whose thickness is 0.05 inch.

This study does not use any vision programs, so all the geometrical information about the parts must be prescribed. The positional tolerance of each part is

$\pm$ ((width between two fingers) - (diameter of the part)) / 2

When the part is located within this tolerance, the assembly can be carried out quite successfully. If the robot can see the environment, the parts need not be so carefully positioned. In a hand-eye system, the following information should be determined by a vision program.

(1) identification of the parts

(2) X, Y, Z position of the parts

(3) R-orientation of the parts

(4) maximum diameter of the parts

(5) thickness of the parts.

The force sensor complex of the Little Robot System, consisting of six L.V.D.T.s, has a maximum range of about 1 pound and a resolution of about 0.25 ounce. When the system is first started up, gravity offset calibration is done automatically. However, one cannot avoid the small hysteresis and drift that arise from the mechanical behavior of the force sensor complex. This reduces reliable measurement. In order to make the sensory system more sensitive and reliable, it seems neccessary to develop a more sophisticated force signal processing program, that compensates for the drift and the hysteresis.

Servoing is done by a machine language program running in the PDP-6, every 1/60 second. The terminating condition is checked by a time-shared LISP program in the PDP-10. Under some circumstances, this arrangement causes the robot to miss the termination condition. To prevent such timing errors, I believe that the terminating conditions should be checked in the servoing loop on a real time basis, and that the termination should be reported to the higher level language via interrupt facilities.

Lastly, I would like to add a trivial comment: Force feedback enables the robot to guide a peg into a hole quite reliably, given that the parts do not slip in the hand. From a practical point of view, it is also important to develop a general purpose hand that prevents "slip" or that at least detects its occurrence.

Bibliography

1. Silver, D. (1973): The Little Robot System, MIT A.I. Memo 273
2. Minsky, M. (1972): Mini Robot Proposal to ARPA, MIT A.I. Memo 251
3. Inoue, H. (1971): Computer Controlled Bilateral Manipulator, Bull. of Japan Society of Mechanical Engineers,Vol.14, No.69.
4. Shirai, Y. and Inoue, H. (1973): Guiding a Robot by Visual Feedback in Assembly Tasks, Pattern Recognition, Vol.5, pp99-108.
5. Paul, R. (1972): Modelling ,Trajectory Calculation and Servoing of a Computer Controlled Arm; Stanford AIM 177.
6. Bolles, R. and Paul, R. (1973): The Use of Sensory Feedback in a Programmable Assembly System, Stanford AIM 220.
7. Goto, T. et. al. (1974): Precise Insert Operation by Tactile Controlled Robot, 2nd International Conference on Industrial Robot Technology.
8. Nevins, J.L. et. al. (1974): Exploratory Research in Industrial Modular Assembly, The Charles Stark Draper Lab. R-800.
9. Moon, D.A. (1974): MACLISP REFERENCE MANUAL, PROJECT MAC, M.I.T.