4-1-1987

Contour Tracking with Force Feedback

Shaheen Ahmad
Purdue University

Contour Tracking with Force Feedback

Shaheen Ahmad

TR-EE 87-12
April 1987

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Presented at Robots 11, Chicago, IL, April 1987

CONTOUR TRACKING WITH FORCE FEEDBACK

Shaheen Ahmad
Real-time Robot Control Laboratory
School of Electrical Engineering
Purdue University
West Lafayette
Indiana 47906
USA

Abstract

In this paper we describe an algorithm to allow a manipulator to track a complex contour without having to teach or program any points on the contour. This is an important problem in many manufacturing situations, when a robot tool such as a deburring tool must follow a complex work piece contour. In such instances it is tedius to teach or program the manipulator to follow such contour additionally caliberation of the manipulator or the workpart may be difficult. The algorithm we present has been exprimentally demonstrated utlizing a force sensor, a five degree of freedom manipulator and a 68000 single board computer.

Introduction

In many manufacturing tasks such as deburring, application of sealents and adhesives, etc. The manipulator is required to track a contour of an object. In such instances the workpiece must be accurately positioned and the contour of the object must usually be taught [Paul 81] (see Figure 1) or it must be generated from a CAD database, or a manufacturing blueprint (drawing). In such cases a caliberation process is necessary to align the actual coordinate frame of the part with that of the programmed coordinates. This must be carried out every time a part is worked on. This requires additional setup overhead time reducing productivity. Expensive jigs are also required to locate the part such that it is correctly positioned.

In this paper we describe an alternative method for tracking object contours with force sensor feedback. This does not require prior teaching of object frame or additional positions describing its contour.

Section II of this paper describes the contour tacking algorithm. Section III of this paper describes the experimental results, Conclusion is found in Section IV.

II. The Contour Tracking Algorithm

In this section we describe the contour tracking algorithm in detail. The basic purpose of the algorithm is to guide the robot around an irregularly shaped object so as to determine or follow its contour. The object can be inaccurately placed in the robots workspace. Once the shape information is obtained, it can be stored for use at a later time. This could be useful for such tasks as removing flashing from a die cast part. A human could remove the flashing off of one part, have the robot learn the part's shape, then use the stored shape information to have the robot remove the flashing off any similar part at any time.

The algorithm is rather simple conceptually. In a sentence, the robot just moves at a right angle to the contact force. Problems arise in that the robot must somehow approach the object without knowing exactly where it is. It must not loose contact until it has completely traced around the object, and it must not apply too much force to the object. It must also determine when the tracing around the object is complete.

Additional problems arises if the robot controller is only capable positional control. That is the robot cannot servo to a force when it is in contact. In such a case it is necessary to move the robot in small but appreciable distances to increase or decrease the force. This increases the risk of losing contact with the object, or applying too much force. The application of too much force can cause the robot actuators to saturate and the arm to jam and hence fail to reach its end position, and this causes many commercial robots controllers to "lock up" and not respond to any more commands. The algorithm developed addresses all these problems and a flow chart of this algorithm is shown in Figure 2.

The Contour Tracking Flowchart Description

As can be seen in Figure 2, the algorithm is divided into six major sections. These sections are described below:

Initialization

Before the robot can even begin to approach the object, the control system must initialize a variety of parameters. First, it must move the arm to a safe position. Then initialize the force sensing system and record any existing bias forces sensed at the sensor. The bias force would include weight of any structure which the force sensor supports. Such as the robot wrist, if a wrist force sensor is used. Weight of objects placed on the table if a force sensing table is used.

Approach Phase

Once at the starting position, the robot then approaches the object by moving small incremental distances some what equal to the tolerence with which part is positioned in the robot workspace. The robot is moved in this fashion until a threshold force is exceeded at that point to ensure an actual contact. The *contact* force threshold ($F_{contact\_min}$) and its value must be experimentally determined. This is because most manufacturing environments are noisey, for example when the manipulator accelerates or stops forces are experienced at the wrist if a wrist sensor is used, or at the force table if a force table is used. Multitude of other events cause force signals to be registered. As a result the contact threshold must be set sufficiently high enough to keep force sensor noise signal from falsely causing a valid contact condition. The contact test is then:

$$\text{if} \quad \left(\left(F_{contact} = \sqrt{(F_x^2 + F_y^2)} \right) > F_{contact\_min} \right) \quad then \quad contact\_established; \tag{1}$$

where F_x and F_y are the contact forces measured by the force sensor. Once contact has been made, the algorithm enters into the tracing phase.

The Contact Threshold Force

The robot must now be moved to a new position while maintaining contact. The contact force must be appropriately selected such that excessive force is not exerted which may result in saturation of the robot actuators. In larger robots large contact forces may damage the robot and the workpiece. The actuators are saturated the robot may be jammed, and if it is a position controlled device it may not accept a new set point until the position error is zero. There is a range of forces $[F_{jam}, F_{jam-max}]$ in which the robot may jam depending on the configuration of the arm, frictional forces and the geometry of the contour. The contact force must then be selected in such a way that:

$$\frac{1}{2}\left(\mathrm{F}_t + \mathrm{F}_{contact\_min}\right) < \sqrt{(\mathrm{F}_x^2 + \mathrm{F}_y^2)} < \mathrm{F}_{jam} \tag{2}$$

where $\mathrm{F}_{jam} < \mathrm{F}_{jam\_max}$.

The desired contact force must therefore be experimentally found. This obviously depends on the material of the tool tip and the workpiece. The touch force F_t is set at considerably less than the actually jamming force F_{jam}. As the robot must able to move from one contact point to the next contact point without jamming up (and requiring human intervention).

If the contact threshold and the jamming threshold are too close together, the robot may not be able to position the force in between them due to its minimum travel distance. This can remedied by increasing the spread between the values, or by reducing the incremental robot moves to smaller distances.

This thresholding action is illustrated in Figure 3. As the position of the probe moves from $\mathbf{P}_{i-1}$ to $\mathbf{P}_i$ the robot tries to keep the contact force value between $\mathbf{F}_{\min}$ and $\mathbf{F}_{\max}$ before it tries to move at a right angle to the force.

Record Position

When the robot has completed an incremental move while keeping in contact with the object its new probe position is recorded. This position must be compensated for as the robot and its tool flexibility alters actual location of the object contour, as sensed at the joint position sensor.

Conditions for Terminating the Contour Tracking Motion

The starting position on the contour is defined as the first point at which contact was made. The robot is continually moved to the right while maintaining a contact. A stopping region is defined as being the circle with the center as the starting point. When the tool enters this circle the contour tracking operation is terminated. This is shown in Figure 4.

Moving Around the Contour

The probe must be moved from position $\mathbf{P}_{i-1}$ to $\mathbf{P}_i$. In this experiment the current force information is used to compute the incremental move to point $\mathbf{P}_i$. Given the radial step size for the movement is Δr, then

$$dP_{x,i} = \Delta r \frac{F_x}{\sqrt{(F_x^2 + F_y^2)}} \tag{3}$$

$$dP_{y,i} = - \Delta r \frac{F_y}{\sqrt{(F_x^2 + F_y^2)}} \tag{4}$$

where F_x and F_y are forces monitored from the current contact. Then,

$$\mathbf{P}_i = \mathbf{P}_{i-1} + d\mathbf{P}_i \tag{5}$$

where

$$d\mathbf{P}_i = (dP_{x,i}, dP_{y,i})^t \tag{6}$$

Once the move is completed the contact force is checked, if the contact force is below the minimum contact force a new move $d\mathbf{P}_i'$ is computed. The new incremental move bisects the angle between the initial direction of the incremental movement and the direction of the maximum force. This is shown in Figure 5. If the move $d\mathbf{P}_i'$ does not ensure a minimum contact force, the angle between $d\mathbf{P}_i'$ and the direction of the maximum force is further halved until contact is made.

Effects of Probe and Robot Compliance

As mentioned earlier the robot used in this experiment is a position controlled device as a result compliance is essential for fine force resolution. If K_x is the cartesian stiffness of the end-effector and $dx_{\min}$ is the minimum cartesian movement, then,

$$\mathbf{F}_{\min} = K_x dx_{\min} \tag{7}$$

Additional compliance can be added by a compliant probe (K_{probe}) as:

$$K = \frac{K_x K_{probe}}{K_x + K_{probe}} \cong K_{probe} \quad \text{if } K_x > K_{probe} \tag{8}$$

where K is the altered stiffness of the robot and the tool as seen at the tip of the probe. Figure 6, shows the effects on positional error with a compliant probe and a stiff probe.

4. Results of the Experiment

The algorithm that was developed to trace the path of the object on the force table worked quite well. The probe maintained contact with the object and never became lost. The process of tracing the entire object was fairly slow due to the nature of the robot communication as it took long for the robot to execute the move commands. When the object was retraced with previously calculated and stored joint positions, the process of tracing the perimeter of the object was cut to roughly one half of the original time. A photograph of the experimental workstation is shown in Figure 7.

Plots of Object Outline

Four different objects were traced and the of their outlines were plotted. Figure 8 shows three of the objects. Figures 9, 10 and 11 show the outlines of the objects from the stored data. These outlines are a by-product of the algorithm, since the algorithm's purpose is to teach the robot how to trace around an object. Although the Cartesian points plotted may have errors in them, the angular positions stored are still correct because they truely represent the position of the contact with the object, at that specified force.

Analysis Of Results

There are three kind of errors most notable in the outlines of the plotted objects. The first error is the shrinking of the object, primarily in the x dimension. The shrinking of the dimensions of an object is generally caused by the compliance of the probe and the robot arm joints. It may also be caused by backlash in the gears, creating the effect of the robot having to move the joint further to achieve the same pressure. However, the dimensional distortion due to backlash would be small. The reason the object shrinks more severly in the x dimension is because this axis is aligned with the radial dimension of the arm. There are three joint along the arm in this direction (for our robot), causing much more compliance in x than along the y -axis.

The second type of error is a broad edge distortion along the sides. This type of error is contributed mostly to parametric errors in the robot. The length information of the arm is approximate, as is the joint angle zero positions and as is the arm gear ratios. Along with the eccentricity of the gears this will lead to nonlinear distortion in the radial direction of the robot. This will occur more noticeably in the x direction, once again because it is in the radial direction of the arm, the plane in which most of the robot joints operate.

The third type of error is jagged lines along all of the object edges. There are two major considerations for this error. The first and more noticeable was due to the thresholding of the force. One point may have had the maximum pressure against it when its neighbor may have had the minimum force. Since there is compliance, this force difference may cause small random fluctuations in distance. Another source for this error is due to the resolution of the plots themselves. Many jagged edges appear that way because the plot has moved over one pixel distance. Analysis on robot train errors can be found in Ahmad [Ahmad 87], analysis of errors in robot link parameters can be found in Hayati [Hayati 83].

5. Conclusion

In this paper we described an algorithm to trace a contour around a complex object. In many manufacturing tasks objects can now be traced without accurately locating the workpiece or knowing its shape.

Acknowledgements

Jim Gallo and Correy Ustanik were the first students to perform this experiment, the results presented in this paper are reprinted from their laboratory reports.

References

[Ahmad 87]
Ahmad Shaheen, "Analysis of Robot Drive Train Errors Their Static Effects and Their Compensations," TR-EE-87-4, Purdue University, West Lafayette, Indiana 47907.

[Hayati 83]
Hyati, S., "Robot Form Geometric Link Parameter Estimation," Proceedings of the 22nd IEEE Conference on Decision and Control, December 1983.

[Paul 81]
R. P. Paul, "Robot Manipulator," MIT Press, 1981.

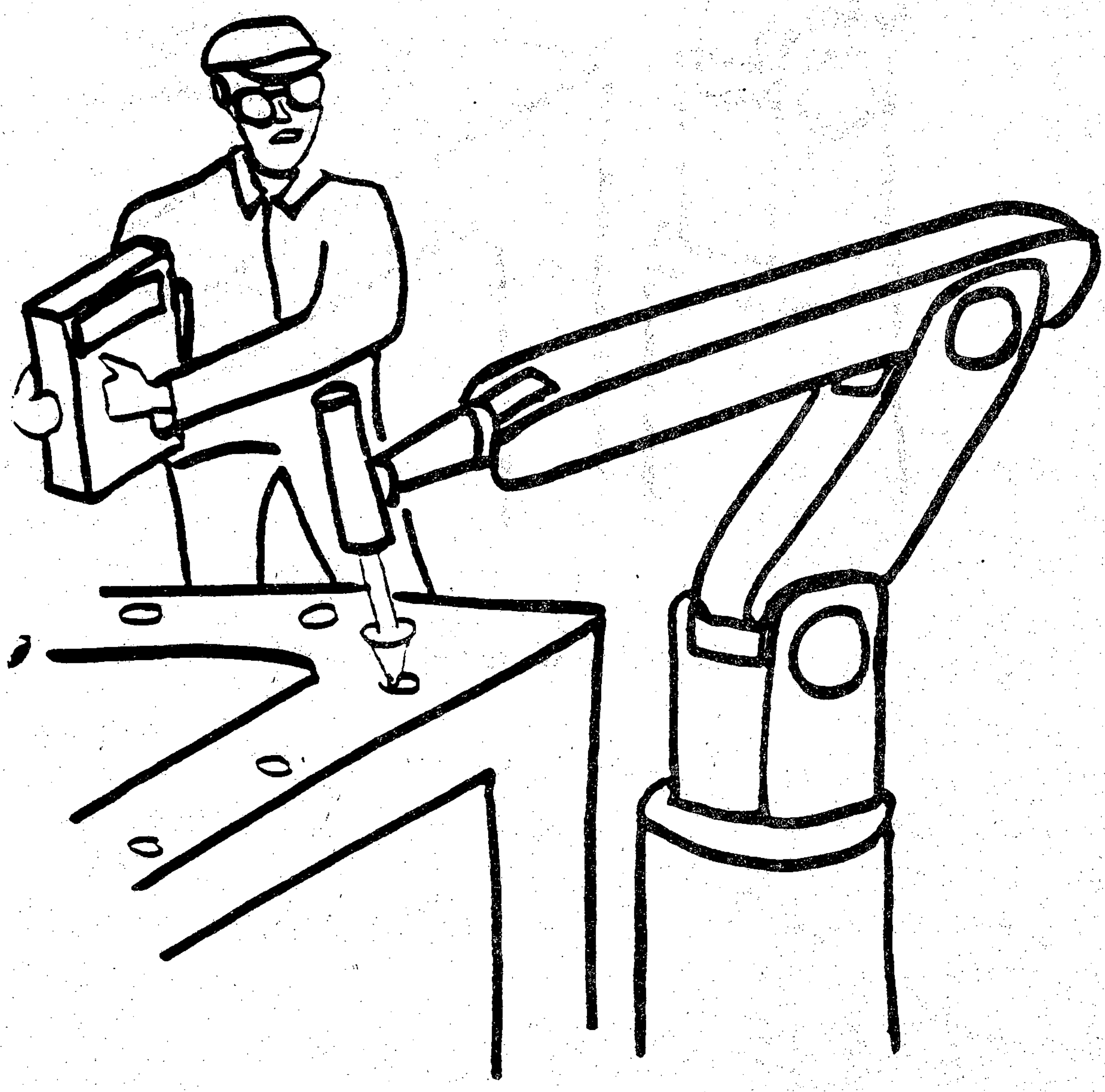


Figure 1
Robot Programming by Teaching

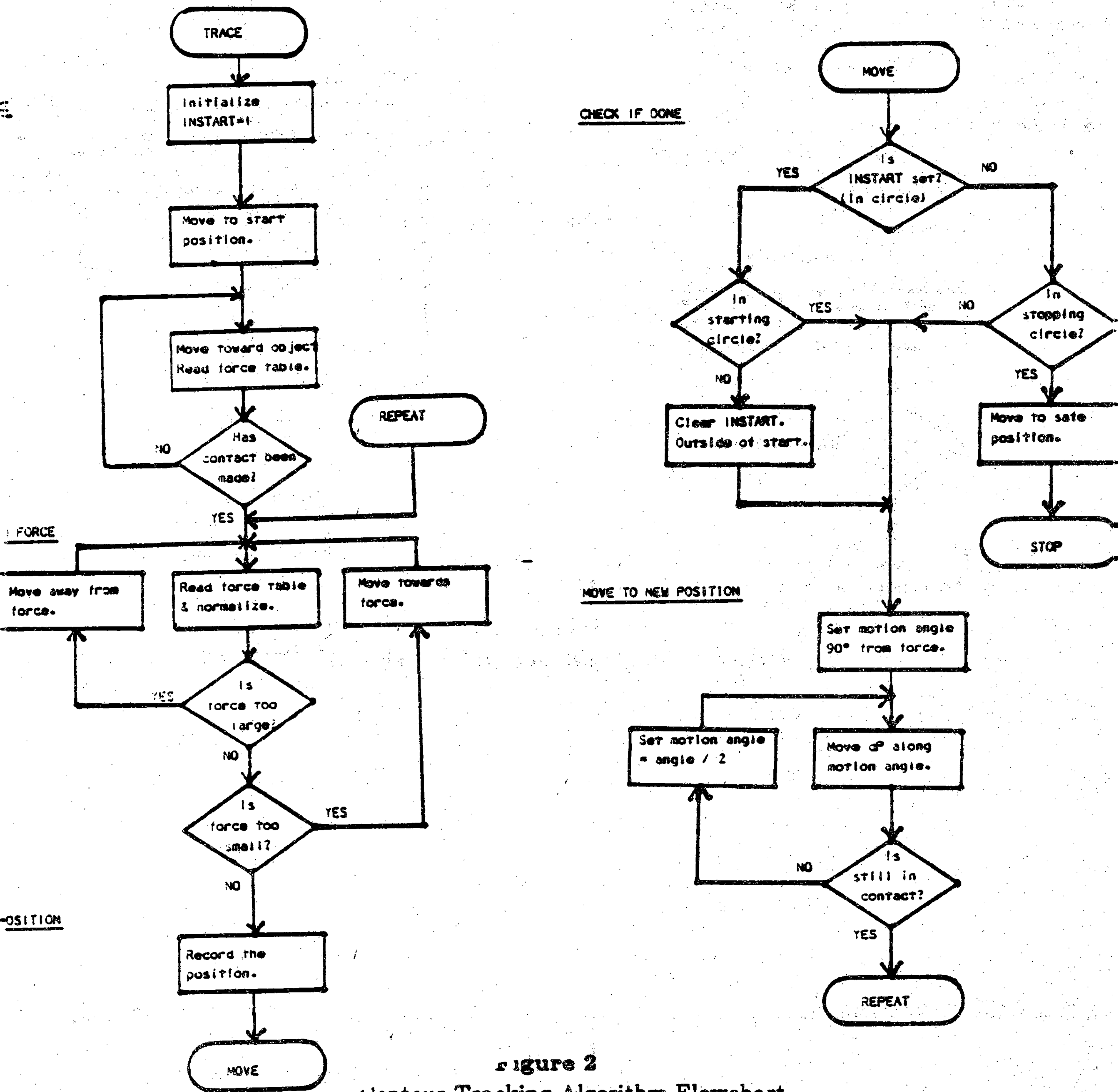


Figure 2
The Contour Tracking Algorithm Flowchart

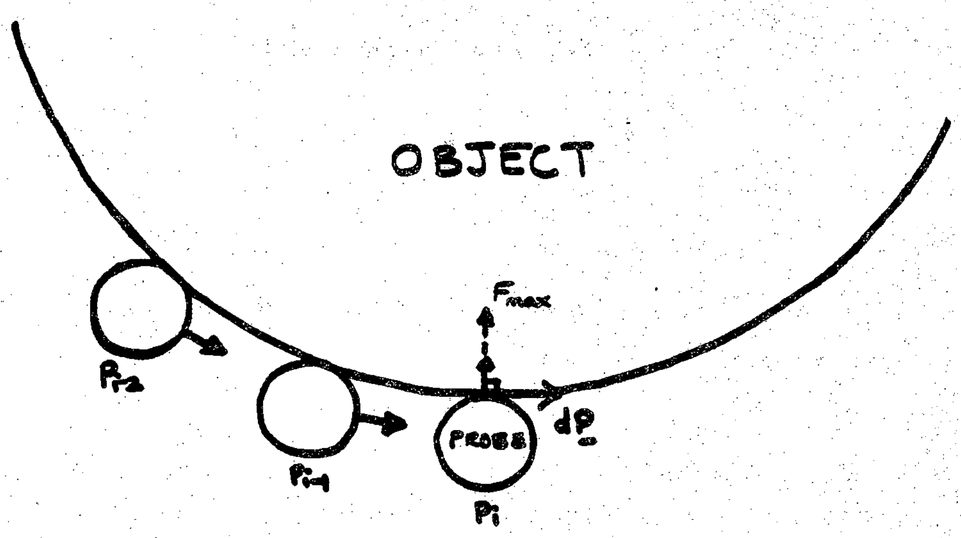


Figure 3
Thresholding the Contact Force

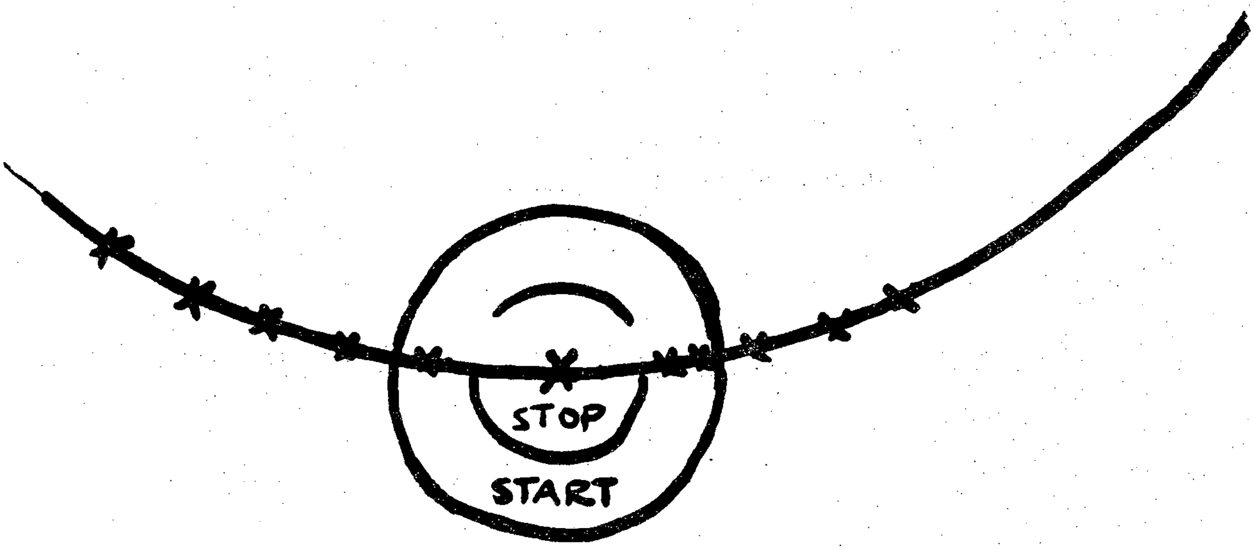


Figure 4
Starting and Stopping Regions

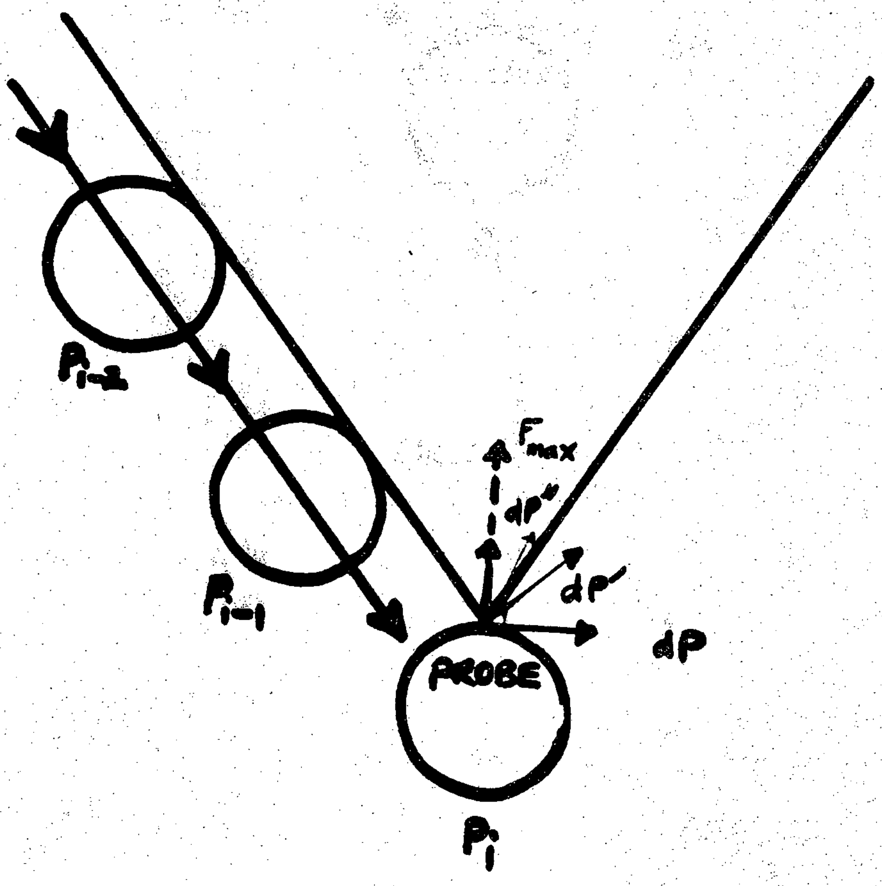


Figure 5
Movement to New Position

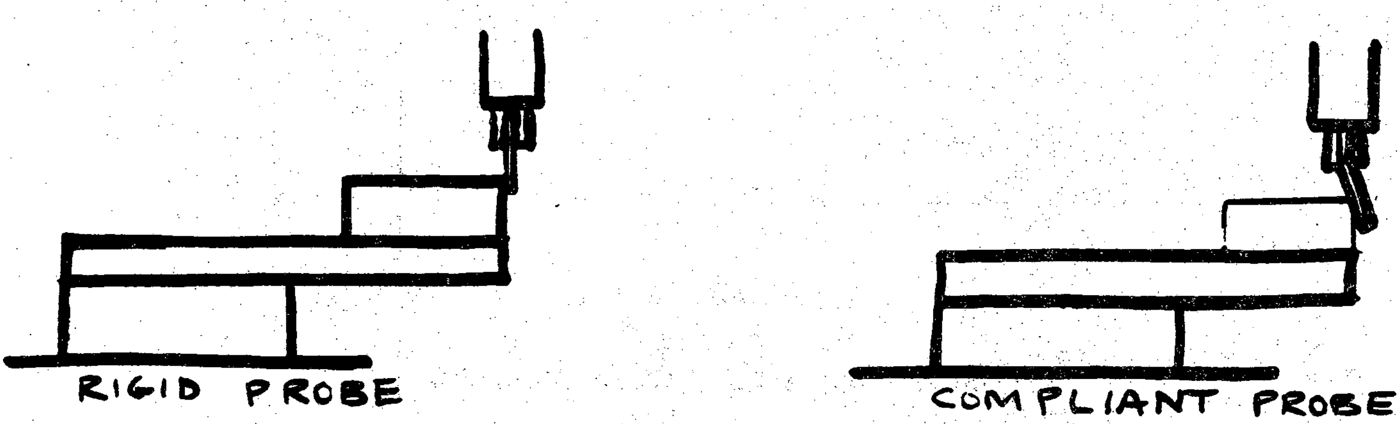


Figure 6
Positional Error With A Compliant Probe

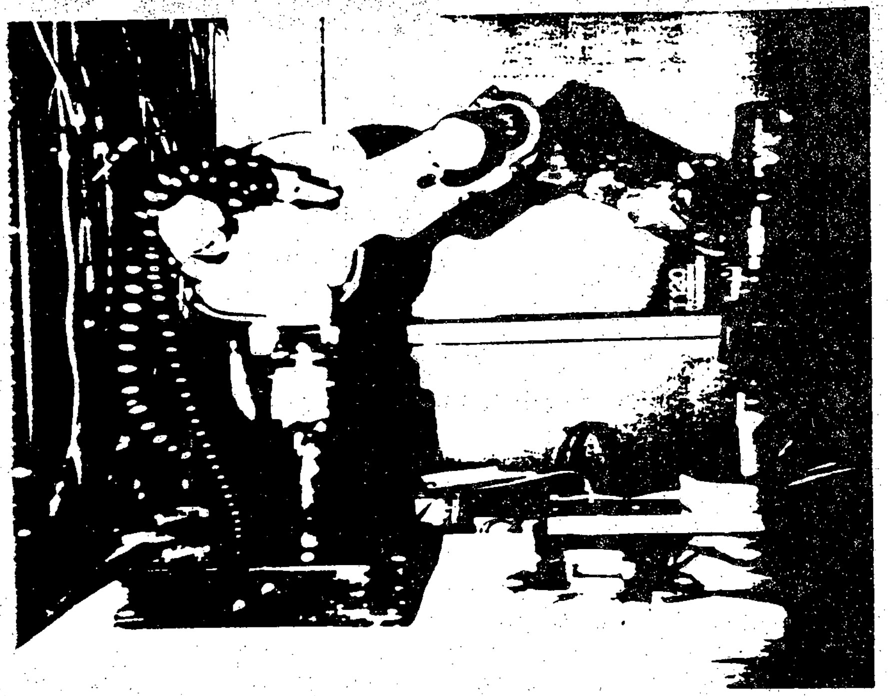


Figure 7
Photograph of Experiment Workstation

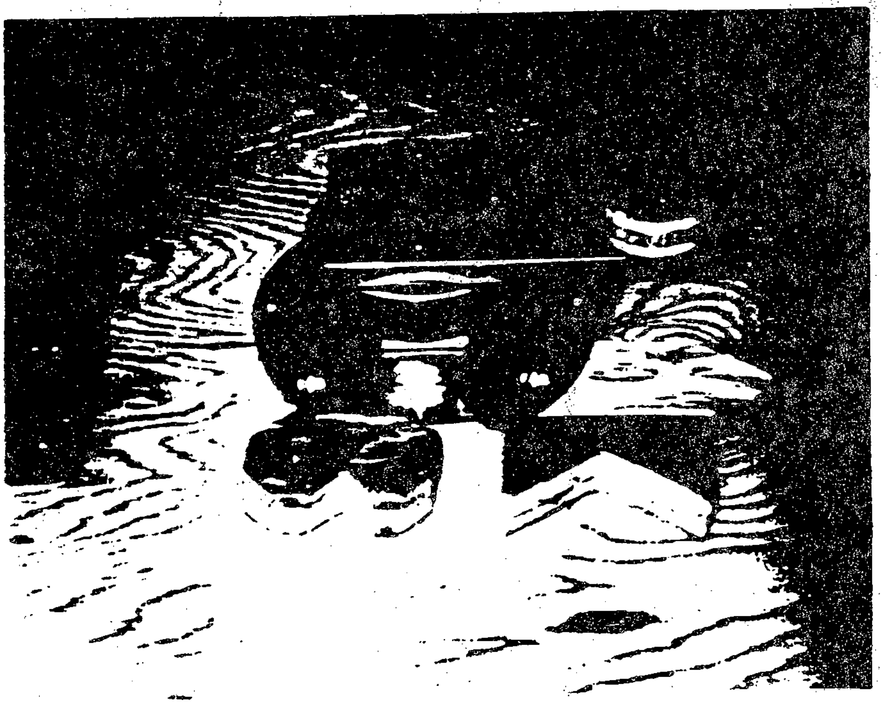
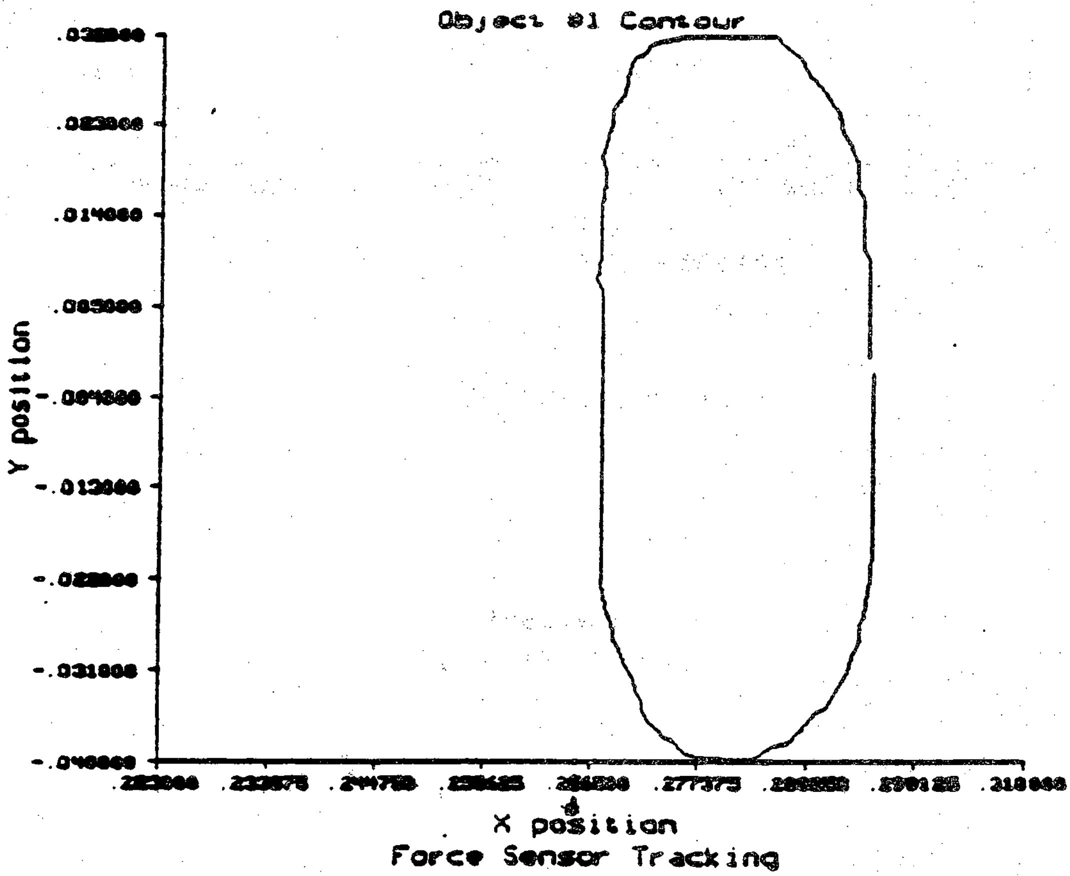


Figure 8
Photograph of the Objects



Object #1 Contour

Figure 9
Object #1 Contour

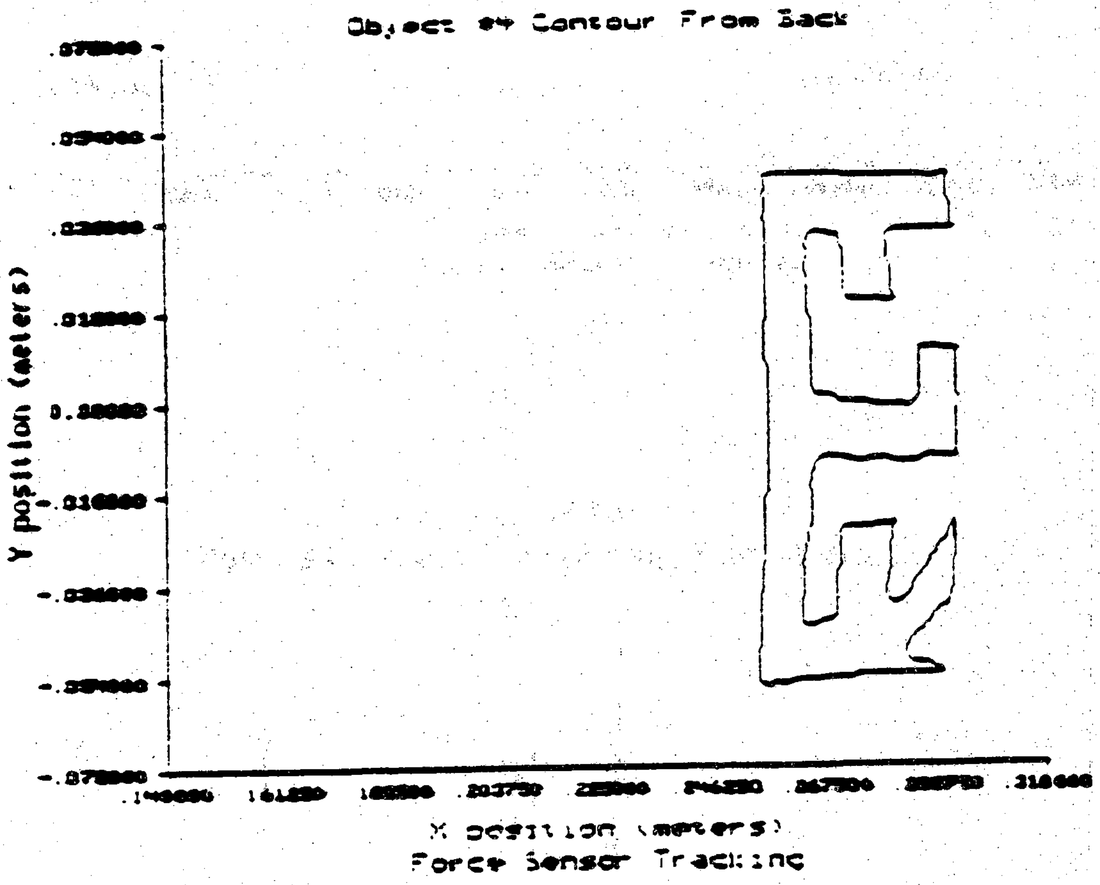


Figure 10
Object #2 Contour (Traced From Front & Back)

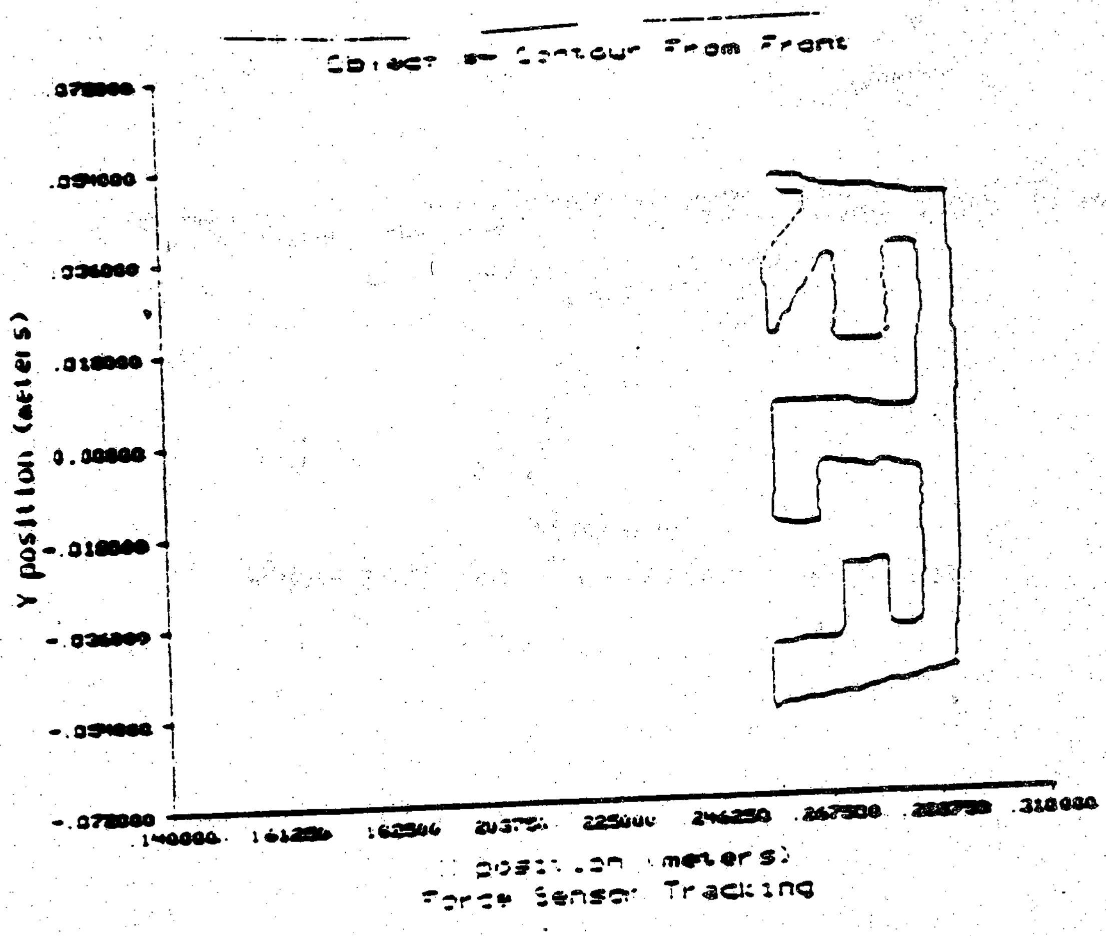


Figure 11
Object #3 Contour (Traced From Front & Back)