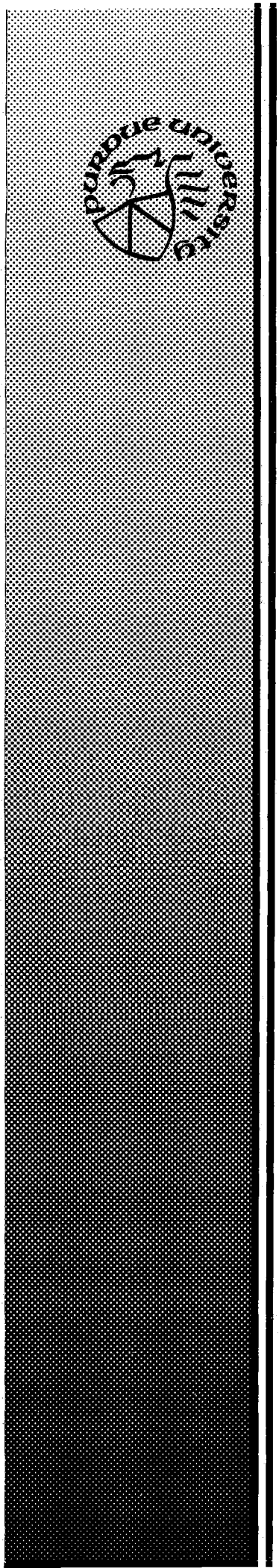
3-1-1990

Shape Recovery from Robot Contour Tracking with Force Feedback

Shaheen Ahmad
Purdue University

Chuan N. Lee
Purdue University



Shape Recovery from Robot Contour Tracking with Force Feedback

Shaheen Ahmad
Chuan N. Lee

TR-EE 90-21
March 1990

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

April 27th, 1990

Submitted to International Journal of the Robotics Society of Japan

SHAPE RECOVERY FROM ROBOT CONTOUR - TRACKING WITH FORCE FEEDBACK

Shaheen Ahmad, Chuan N. Lee[†]

Real-Time Robot Control Laboratory

School of Electrical Engineering

Purdue University

West Lafayette, IN 47907-0501

U.S.A.

Abstract

In this paper we describe a process for shape recovery from robot contour-tracking operations with force feedback. Shape recovery is an important task for self-teaching robots and for exploratory operations in unknown environments. An algorithm which directs a position controlled robot around an *unknown planar contour* using the *steady state contact force* information is described in this paper.

The shape recovery from the planar contouring is not a trivial problem. It is experimentally found that there is significant distortion of the original contour if direct kinematics is used to recover the object's shape, as we are unable to recover the exact position of the robot tool due to the errors present in the kinematic model of the arm

[†] Now with Adept Technology, Los Angeles, CA.

and the non-linearities of the drive train. Drive train errors can consist of the joint compliance, gear backlash and gear eccentricity. A mathematical model of the errors generated by the drive train has been previously addressed. In this paper a compensation process is explored for purposes of planar shape recovery. It is found through experimentation that the joint compliance is most conveniently compensated for in practice. Improvements in the shapes recovered from robot contouring are seen with our compensations. Experimental details and difficulties are also discussed.

1. Introduction.

There are many papers published in the area of force and compliance control [9] [10] [11] [16] [17] [18] [19] [20] [21] [22] [23]. *This paper does not address the problem of controlling the end-effector forces. Also, this paper does not address the problem of recognition of objects through probing.* It deals with measuring the static end effector force to recover the shape of an unknown planar contour present in the robot workspace.

Processes that involve the contouring of an object is common in the industrial environment. For instance, when a robot is used to deburr a workpiece using a grinding tool as the end-effector, it is required to be able to move the grinding tool along the contour of the workpiece without direct assistance from human operators. This implies the use of sensors, usually force sensors, to detect contact with the workpiece as well as to control the force with which the grinding tool is to be applied to the workpiece. The force information is also used to move the end-effector around the workpiece.

Contour processes have long been the subject of experimentation and research.*

*Numerous publication in this area exists, we do not cite or discuss the pros and cons of all the papers. We limit our discussion to a few representative of the body of papers that are published.

Most of the past work in this area is related to force control and deburring. Thiessen [1] described an early application of an ASEA robot performing deburring tasks at a cast iron factory. A force sensor was used with the robot to ensure sufficient deburring force. Plank and Hirzinger [2] extended Thiessen's work by devising an algorithm to vary the velocity of the grindstone based on the size of the burr encountered. This addition increased the robot's effectiveness as a deburring device. Stepien, Sweet, Good, and Tomizuka [3] also reported work on a deburring robot. Their emphasis was on the design of a force control system that can be used to correct the position of the robot until a desired contact force is applied on the object. Starr [4] used a PUMA 560 robot to follow edges while applying a constant contact force. Wampler [5] used optical proximity sensors in his surface-tracking experiment, the optical sensors were used to detect distances from edges, and the control algorithm he described was able to position the robot to a specified distance from the edges.

In a typical robot contouring operation, an unknown object, whose contour is not usually analytic, is set upon a rigid base. The object is assumed to be non-deformable under external forces. The robot arm with its end-effector as a probe is made to track along the contour of the object. A force sensor is used to determine when there is contact between the end-effector and the object. As the probe is moved along the contour of the object, the joint positions of the robot arm are recorded, and when they are converted to Cartesian coordinates, the points that form the outline of the object are obtained. This contour might be used for fast retracing the object without the use of force sensors. However, there are errors in the recorded points. These errors arise from the assumption that the robot joints are rigid and the robot gear train provide a constant gear ratio. These assumptions are not in general correct [17]. Also because the joint position sensors are mounted on the motor, discrepancies between the true joint position and the motor position will exist. The kinematic parameters of the arm such as its link lengths and offset angles are also not precisely known. This paper is

concerned with the recovery of shape probed by a robot end effector and the guiding of the probe around unknown contours using force feedback. The work most closely resembling this paper with respect to shape recovery is reported by Whitney and Edsall [6] who discuss the filtering of general errors from a robot contour. Sources of the errors and their models were not clearly discussed in their paper. Hemami and Godard [25] formulated the shape recovery problem as a problem of solving nonlinear partial differential equations. They assume a rigid robot model thereby ignoring nonlinearities introduced by drive train errors. Experiments were not performed to support their work; also, schemes for kinematic or dynamic tracking of an unknown objects were not discussed in their paper either. Most recent publication in this area by Blauer and Belanger [24] discusses estimation of some contact surface parameters related to the shape of the object by the use of force feedback. The shape of the part was assumed to be known. Experiments were not performed by the authors to verify their results. Perfectly rigid body model of the robot is also used in their derivations. Huang and McClamroch [26] formulated and solved part of the problem of moving an end-effector along a contour in minimal time. Problems related to shape recovery, drive train errors or tracking an unknown contour were not discussed in their paper. In real life robotic applications, a robot maybe required to trace a shape which may not be analytically defined and or it maybe unknown. Additionally the robot will exhibit joint flexibility and other errors making the shape recovery of unknown contours complex. In section two of this paper, the sources of errors introduced by a robot arm are described in mathematical terms. Section three describes a static contour tracing algorithm utilizing force feedback. Section four contains a description of the setup of the experiment for implementing the compensation of joint compliance errors. Section five describes and discusses the results of the experiments.

2. Sources of Error In Direct Shape Recovery

In this section, the sources of error are analyzed in mathematical terms. These errors prevent shape recovery from direct joint measurements. Considerations relating to robot kinematic errors are not presented here, several publications related to this issue exist see [12] [13] [14] [15]. The issues related to drive train errors are briefly presented in this paper.

2.1. Gear Backlash.

Backlash is the gap between the teeth of the two gears of a gear pair that causes motion to be lost. Backlash takes effect when the direction of motion of a gear pair changes. The amount of motion lost can be considered to be constant for a particular gear pair. It is important to realize the reason why this lost motion, as well as other drive train errors, are not detected by the encoders. This is because the joint encoders on robots are directly mounted on the motor and only sense the motion of the motor. Drive train errors occur without any motion of the motor. This is because the motion of the motor is used to drive the gears which then actuate the joints of the arm. Most industrial robot system do not incorporate joint sensors, as a result drive train errors are observed.

2.2. Predicting backlash.

When the end-effector of a robot arm comes in contact with a rigid surface, backlash in every joint must be taken up before the end-effector can exert any force that is not due to the weight of the arm. The backlash for each joint is practically a constant amount independent of the force eventually exerted by the arm. Therefore, the problem is not in predicting the amount of backlash in each joint but rather its direction. To predict the direction of backlash in each joint, the resultant torque at the joint must be computed from all the external forces acting on the link. These forces are due to

gravity and the force of contact with the surface.

Let m_i refer to the mass of the ith link. Let $\underset{\sim}{r}_i \in \mathbb{R}^{3\times1}$ be the position vector of the center of mass of link i with respect to O_i, the ith origin, and described in $(x_i, y_i, z_i)^t$ coordinates. The vector $\underset{\sim}{r}_i$ is therefore a constant vector since O_i is fixed in link i. Let ${}^iS_j \in \mathbb{R}^{4\times4}$ denote the 4×4 rotation matrix that is partitioned as:

$$ {}^iS_j = \begin{bmatrix} {}^iR_j & 0_{3\times1} \\ \underset{\sim}{0}_{1\times3} & 1 \end{bmatrix} \tag{1} $$

where ${}^iR_j$ is the 3×3 rotation part of the homogeneous transformation matrix ${}^iA_j$, $\underset{\sim}{0}_{1\times3} = [0, 0, 0]$, and $0_{3\times1} = [0, 0, 0]^t$,

Suppose there is one point in space labeled P. Next suppose there are two coordinate frames 1 and 2 related to each other by the homogeneous transformation matrix ${}^1A_2 \in \mathbb{R}^{4\times4}$, see Paul [8]. If the vector from O_1 pointing to P is named $\underset{\sim}{p}^*_{x_1y_1z_1}$ described in $(x_1,y_1,z_1,1)^t$ coordinates and the vector from O_2 pointing to P is named $\underset{\sim}{p}^*_{x_2y_2z_2}$ described in $(x_2,y_2,z_2,1)^t$ coordinates, then $\underset{\sim}{p}^*_{x_1y_1z_1}$ and $\underset{\sim}{p}^*_{x_2y_2z_2}$ are related by the equation: $\underset{\sim}{p}^*_{x_1y_1z_1} = {}^1A_2\, \underset{\sim}{p}^*_{x_2y_2z_2}$ where $\underset{\sim}{p}^*_{x_1y_1z_1} = (x_1,y_1,z_1,1)^t$ and $\underset{\sim}{p}_{x_2y_2z_2} = (x_2,y_2,z_2,1)^t$

If the origins of frame 1 and 2 were to become the same (while the orientation of the frames were still different), then we have the following equation:

$$ \underset{\sim}{p}^*_{x_1y_1z_1} = {}^1S_2\, \underset{\sim}{p}^*_{x_2y_2z_2} \tag{2} $$

Let us now consider the gravity load on the joints, consider joint 1. The weights of links 1 through n will contribute to the total torque vector at joint 1. Each link produces a torque equal to:

$$\underset{\sim}{\tau} = \underset{\sim}{r} \times \underset{\sim}{F} \tag{3}$$

where $\underset{\sim}{\tau} \in \mathbb{R}^{3\times1}$ is the torque vector, $\underset{\sim}{r} \in \mathbb{R}^{3\times1}$ is the position vector of the link center of gravity with respect to joint 1 (or whichever joint that is being considered) and $\underset{\sim}{F} \in \mathbb{R}^{3\times1}$ is the gravity vector of the link. Both $\underset{\sim}{r}$ and $\underset{\sim}{F}$ must be described in the same coordinate frame to produce the correct result. The resulting torque vector will also be described in the same coordinate frame as $\underset{\sim}{r}$ and $\underset{\sim}{F}$. The actual scalar torque u_i that acts at the ith joint is that component of the torque vector $\underset{\sim}{\tau}_i$ that is along the $\underset{\sim}{z}_{i-1}$ axis, the axis of rotation of the ith joint, i.e., $u_i = \underset{\sim}{z}_{i-1} \cdot \underset{\sim}{\tau}_i$.

Let us compute the contribution of link 1 weight to the joint 1 torque $\underset{\sim}{\tau}_{1,1} \in \mathbb{R}^{3\times1}$. Assume $\underset{\sim}{r}_1$ is the coordinate of center of gravity of link 1 measured with respect to O_1 the coordinate reference frame of link 1. The gravity force vector is $\underset{\sim}{F}_1 = m_1 \underset{\sim}{g}$, where $\underset{\sim}{g} \in \mathbb{R}^{3\times1}$ is the constant gravity vector described in coordinate frame O (i.e. the world coordinate frame). Then the contribution of the weight of link 1 to the torque at joint 1 is:†

$$\underset{\sim}{\tau}^*_{1,1} = {}^0A_1 \, \underset{\sim}{r}^*_1 \times m_1 \underset{\sim}{g}^* \tag{4}$$

where the vector $\underset{\sim}{\tau}^*_{i,j} \in \mathbb{R}^{4\times1}$ denotes the contribution of the weight of link j to the torque at joint i. For convenience of dealing with homogeneous transforms, we define the cross product of two (4×1) vectors $\underset{\sim}{u}^* = [\underset{\sim}{u}, 1]^t$ and $\underset{\sim}{v}^* = [\underset{\sim}{v}, 1]^t$ as follows: $\underset{\sim}{u}^* \times \underset{\sim}{v}^* = [\underset{\sim}{u}, 1]^t \times [\underset{\sim}{v}, 1]^t = [\underset{\sim}{u} \times \underset{\sim}{v}, 1]^t$, where $\underset{\sim}{u} \in \mathbb{R}^{3\times1}$ and $\underset{\sim}{v} \in \mathbb{R}^{3\times1}$ and usual definition of $\underset{\sim}{u} \times \underset{\sim}{v}$ is used in the above.

†Note vector $\underset{\sim}{r}_i \in \mathbb{R}^{3\times1}$ and vector $\underset{\sim}{r}^*_i \in \mathbb{R}^{4\times1}$, are related $\underset{\sim}{r}_i{}^* = [\underset{\sim}{r}_i, 1]^t$ eg $\underset{\sim}{g}^* = [\underset{\sim}{g}, 1]^t$, also $\underset{\sim}{\tau}^*_{ij} = [\underset{\sim}{\tau}_{ij}, 1]^t$ and $\underset{\sim}{\tau}^* = [\underset{\sim}{\tau}, 1]^t$

Returning to the above discussion, as another example the contribution of the weight of link 3 to the torque at joint 2 is

$$\underset{\sim}{\tau}^{*}_{2,3} = {}^{1}\mathbf{A}_3 \, \underset{\sim}{\mathbf{r}}^{*}_{3} \times m_3 \, {}^{1}\mathbf{S}_0 \, \underset{\sim}{\mathbf{g}}^{*} \tag{5}$$

where ${}^{1}\mathbf{A}_3 = {}^{1}\mathbf{A}_2 \, {}^{2}\mathbf{A}_3$ and ${}^{1}\mathbf{S}_0 = {}^{0}\mathbf{S}_1{}^{-1}$. To obtain the total torque at a joint i, all the contributions from all links i to n must be summed up:†

$$\underset{\sim}{\tau}^{*}_{i} = \sum_{j=i}^{n} m_j \left({}^{i-1}\mathbf{A}_j \, \underset{\sim}{\mathbf{r}}^{*}_{j} \times {}^{i-1}\mathbf{S}_o \, \underset{\sim}{\mathbf{g}}^{*} \right) \tag{6}$$

where $\underset{\sim}{\tau}^{*}_{i}$ is referenced to coordinate frame i-1. For purposes of computational efficiency, it is better to write, for instance, $\underset{\sim}{\tau}^{*}_{2}$ as:

$$\begin{aligned} \underset{\sim}{\tau}^{*}_{2} &= (m_2 \, {}^{1}\mathbf{A}_2 \, \underset{\sim}{\mathbf{r}}^{*}_{2} + m_3 \, {}^{1}\mathbf{A}_2 \, {}^{2}\mathbf{A}_3 \, \underset{\sim}{\mathbf{r}}^{*}_{3} + \ldots + m_n \, {}^{1}\mathbf{A}_2 \, {}^{2}\mathbf{A}_3 \ldots {}^{n-1}\mathbf{A}_n \, \underset{\sim}{\mathbf{r}}^{*}_{n}) \times {}^{1}\mathbf{S}_o \, \underset{\sim}{\mathbf{g}}^{*} \\ &= ({}^{1}\mathbf{A}_2 (m_2 \, \underset{\sim}{\mathbf{r}}_{2} + {}^{2}\mathbf{A}_3 (m_3 \, \underset{\sim}{\mathbf{r}}_{2} + \ldots + ({}^{n-1}\mathbf{A}_n m_n \, \underset{\sim}{\mathbf{r}}_{n}))))) \times {}^{1}\mathbf{S}_o \, \underset{\sim}{\mathbf{g}}^{*} \end{aligned} \tag{7}$$

The other external force acting on the arm is the force of contact with the surface. This force can be sensed by the force-torque sensor. If the sensed force-torque vector is written as ${}^{S}\underset{\sim}{\mathbf{F}} = [f_x, f_y, f_z, w_x, w_y, w_z]^t \in \mathbb{R}^{6\times 1}$ then the vector of joint torques resulting from external contact, $\underset{\sim}{\tau}_c = [u^c_1, u^c_2, \ldots, u^c_n]^t$, is given by $\underset{\sim}{\tau}_c = \mathbf{J}^t \, {}^{S}\underset{\sim}{\mathbf{F}}$ where $\mathbf{J}^t \in \mathbb{R}^{n\times 6}$ is the transpose of the Jacobian of the robot arm [8].

In order to find the resultant torque at each joint, the torque due to gravity and the torque due to contact must be added together. The direction of the backlash is given by the sign of the total torques experienced at the joint. Given $\underset{\sim}{z}_{i-1}$ is the axis of rotation of joint i and if θ_{b_i} is the amount of backlash in joint i, then the error due to backlash in that joint is:

$$\text{Backlash error in ith joint} = \text{sgn}(\underset{\sim}{z}_{i-1} \cdot \underset{\sim}{\tau}_{i} + u_i^c)\theta_{b_i} \tag{8}$$

2.3. The effects of joint compliance

The analysis of joint compliance begins with the assumption that the joint displays linear behavior under torsion from incremental forces. Note that the externally applied torques must be smaller than the maximum motor torques or the motor will move under the influence of the external torques. Let us model the scalar torque u due to compliance by:

$$u = K\delta \tag{9}$$

where u is the incremental torque, δ is the incremental angle of deflection resulting from the torque, and K is the spring constant or the stiffness.

Assuming that all backlash information needed to perform the calculations in Section 2.2 have been obtained and that all the joint compliances have been obtained experimentally, all that is needed to recover from these two errors would be to compute the total error due to both sources for each joint and to subtract this error from the joint angle read from the joint encoders. If the angle at the ith joint is denoted θ_i, the incremental torque computed at the ith joint $u_i^K = \underset{\sim}{z}_{i-1} \cdot \underset{\sim}{\tau}_{i} + u_i^c$ and the stiffness of the ith joint is K_i, then the following *assignment operation* should be performed for each joint angle:

$$\theta_i := \theta_i - \text{sgn}(u_i^K)\,\theta_{b_i} - \frac{(u_i^K)}{K_i} \tag{10}$$

When inverse kinematics is performed with these corrected angles, the Cartesian position and orientation obtained will be free from a large portion of the drive-train errors. Errors due to gear eccentricity are present but their effects are small and are not considered in this paper, see [7] for further detail.

3. A contour-tracking algorithm.

3.1. Force generation mechanism

In order to understand our contour tracing algorithm, it will be necessary to clarify the mechanics of the force generation. It was experimentally found that most robots have compliant joints [7], [27], [28], [29]. As a result, we may model the robot operations in the xy plane as having a stiffness matrix $\mathbf{K}\in\mathbb{R}^{2\times2}$ along the xy directions. This stiffness varies with the configuration of the robot joints. If the end effector is demanded to move in the xy plane and is obstructed at point P, a force $\underset{\sim}{F} \in\mathbb{R}^{2\times1}$ will be generated at the contact.

$$\underset{\sim}{F} = \mathbf{K}_P \begin{pmatrix} x_d - x \\ y_d - y \end{pmatrix} = \mathbf{K}_P \begin{bmatrix} e_x \\ e_y \end{bmatrix} \tag{11}$$

where $(x_d,y_d)^t$ is the demanded position and $(x,y)^t$ is the actual position of the end effector, see Figure 1. The stiffness of the manipulator at position **P** is $\mathbf{K}_p$. The manipulator compliance **K** measured at the end effector of an arm with six joints operating in six dimensional space is given as:

$$\mathbf{K} = (\mathbf{J}^t)^{-1}\mathbf{K}_\theta\mathbf{J}^{-1} \in \mathbb{R}^{6\times6} \tag{12}$$

where $\mathbf{K}_\theta$ is the diagonal joint stiffness matrix,

$$\mathbf{K}_\theta = \mathrm{diag}(K_i) \in \mathbb{R}^{n\times n} \qquad i = 1...n \quad \text{and usually} \quad n = 6 \tag{13}$$

and K_i is the ith joint stiffness, and $\mathbf{J}\in\mathbb{R}^{6\times6}$ is the manipulator Jacobian between the cartesian coordinate frame and the joint coordinate frame. For our planar xy manipulator example $\mathbf{K}\in\mathbb{R}^{2\times2}$ and $\mathbf{K}_\theta\in\mathbb{R}^{2\times2}$ and $\mathbf{J}\in\mathbb{R}^{2\times2}$.

Suppose the robot tool is located against an edge (point P_c, see Figure 2) with zero or some small nominal contact force. A very *small movement* is demanded from the actuators to move the tool to point D. If we assume the object cannot be deformed

then the end effector will be located at XA. Assuming the demand vector $\underset{\sim}{P}_D = (\Delta_x, \Delta_y)^t$, then we can define

$$r^2 = (\Delta_y)^2 + (\Delta_x)^2 \tag{14}$$

and, the length of the vector $\left|P_c\ XA\right| = r\cos(\theta_2 - \theta_1)$ (15)

where, $\theta_1 = \tan^{-1}\left(\frac{\Delta_y}{\Delta_x}\right)$ and it *represents the direction of the demanded move.* The angle the two dimensional edge makes is θ_2. In that case the steady state actual movement of the tool is:

$$\Delta x_a = r\cos(\theta_2 - \theta_1)\cos\theta_2 \tag{16}$$

$$\Delta y_a = r\cos(\theta_2 - \theta_1)\sin\theta_2 \tag{17}$$

The errors in the movement in terms of the actuator displacements are then:

$$e_x = \Delta_x - \sqrt{\Delta_x^2 + \Delta_y^2}\cos(\theta_2 - \theta_1)\cos\theta_2 \tag{18}$$

and,
$$e_y = \Delta_y - \sqrt{\Delta_x^2 + \Delta_y^2}\cos(\theta_2 - \theta_1)\sin\theta_2 \tag{19}$$

If $\mathbf{K}_{XA} = \mathbf{K}(x_{XA}, y_{XA})$, the corresponding forces that are generated at the contact point are then computed as:

$$\underset{\sim}{F} = \mathbf{K}_{XA}\begin{bmatrix} e_x \\ e_y \end{bmatrix} \tag{20}$$

During a static contouring operation, Δ_x and Δ_y are known and $\underset{\sim}{F}$ can be measured with a force sensor. We wish to determine θ_2 the direction of *the unknown* edge which we desire to follow, this can be calculated as:

$$\tan \theta_2 = \left(\frac{\Delta_y - [0,1] \cdot \left(\mathbf{K}_{XA}^{-1} \underset{\sim}{F} \right)}{\Delta_x - [1,0] \cdot \left(\mathbf{K}_{XA}^{-1} \underset{\sim}{F} \right)} \right) \tag{21}$$

This relation will hold if the contact force $\underset{\sim}{F}$ is lower than some F_{max} which determines the upper limit of the linear region. This force F_{max} is determined by the maximum allowable actuator torque, configuration of the arm and other nonlinearities in the system.

In practice, the robot is operated in those regions of the workspace where the motion qualities are acceptable. We assume in those regions $\det(\mathbf{K}_{XA}) \neq 0$ i.e., $\mathbf{K}_{XA}$ is nonsingular.

The effect of friction: There are three types of friction that can influence the contouring: (a) viscous friction which is proportional to the velocity of the tip of the end effector. (b) static friction which must be broken before motion can occur. (c) coulomb friction which acts in the opposite direction of motion, it acts during the time the end effector is in motion. Therefore, friction effects where the end-effector comes to rest along the contour and how long it takes to reach the rest position. As equation (20) utilizes the steady state forces to determine the local direction of the contour, it is uneffected by friction, because the rest position is always on the contour.

The effect of the bias force: As it is necessary to maintain a bias force to ensure contact, as without contact it would not be possible to compute the surface tangent. Notice that equation (21) is calculated when the end effector is at rest and K_{XA} is calculated for that end effector rest position, therefore, K_{XA} is independent of the bias force. The experimental usefullness of equation (21) will be seen in section 4.2, where it is used to compute the surface gradient during a contour exploration.

3.2. The Contour Tracing Algorithm

We now describe the contour tracking algorithm in detail. The basic purpose of the algorithm is to guide the robot around an *unknown irregularly shaped* object so as to determine its contour. The surface need not be continuous, i.e., its gradient may be undefined at points. 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; 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 lose 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 can arise in practice if the robot controller used is only capable of positional control. Then, 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 may cause many present controllers to "lock up". The algorithm developed addresses all these problems and a flow chart of this algorithm is shown in Figure 3. The algorithm is divided into five major sections.

3.2a. *Initialization*

Before the robot can even begin to approach the object from a known position, existing bias forces sensed at the sensor must be recorded. Also, the approximate location of the object needs to be known so that the end effector may approach it.

3.2b. Approach Phase

The robot is moved towards the object until a threshold force is exceeded to ensure contact has occurred. The *contact* force threshold ($F_{contact\_min}$) is experimentally set. *It must be sufficiently high so as to prevent force sensor noise from falsely recording a valid contact condition.*

$$\text{if } \left[\left(F_{contact} = \sqrt{(F_x^2 + F_y^2)}\right) > F_{contact\_min}\right] \textit{ then contact\_established;} \qquad (22)$$

where F_x and F_y are the sensed contact forces along the x and y axis. Once the contact has been made, the algorithm enters into the tracing phase.

It is important to realize that forces sensed by a force sensor installed on a robot or a table are extremely noisy. If the force sensor is installed on the wrist it will act as an "accelerometer" and yield measurement of the acceleration forces. The impact of the robot gears as they make or break contact can also influence the sensor readings! If the sensor is mounted on a table, the forces transmitted to the table due to robot motion prior to contact are also sensed. Problems may also arise as the sensing structure may have its own dynamics which may be excited by the impact forces during the contouring operation. As a result extreme care is needed to filter the force sensor data and carefully set appropriate thresholds.

3.2c. *The contact threshold force*

Once contact is established 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. This would cause the robot to jam if it is position controlled. As a result, it may not accept a new set point until the position error is zero. Let us define this force as F_{jam_min}, then the contact force has to be maintained within the following range:

$$F_{contact\_min} < \sqrt{(F_x^2 + F_y^2)} < F_{jam-min} \tag{23}$$

In practice 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 position discretization. This can be remedied by increasing the spread between the values. The jamming force is a function of the contact surface friction and it has to be experimentally determined.

3.2d. *Moving around the contour*

The probe must be moved from position $\underset{\sim}{P}_{i-1}$ to $\underset{\sim}{P}_{i}$. Given the incremental movement is Δr, the new position vector $\underset{\sim}{P}_{i}$ must be computed as:

and
$$\underset{\sim}{P}_{i} = \underset{\sim}{P}_{i-1} + (\Delta r \cos\theta_2,\ \Delta r \sin\theta_2)^t \tag{24}$$

where the surface gradient $\tan\theta_2$ is computed by expression (21) given above. Once the move is completed the contact force is checked; if the contact force is below the minimum contact force, a new move $\underset{\sim}{dP}'_i = (\Delta r \cos\theta'_2, \Delta r \sin\theta'_2)^t$ 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 4. If the move $\underset{\sim}{dP}'_i$ does not ensure a minimum contact force, the angle between $\underset{\sim}{dP}'_i$ and the direction of the edge is further halved until contact is made. In the worst case the new direction of movement may lie in between the opposite direction of the previously computed surface tangent and the last move, see figure 4. This enables the robot to move around very sharp corners.

3.2e. *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 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, see Figure 4.

4. Experimental setup.

The experimental setup (see Figure 5) used to implement the compensation of errors due to joint compliance consist of the following: (1) a five axis Naka-Nihon (NND) robot and controller, (2) a Lord Force-Torque sensor, and (3) an Omnibyte 68000 single board computer in a Multibus card-cage for running the software. The NND robot controller is purely position controlled device. It also contains no trajectory generation capability, but these limitations are not important for this experiment.

Backlash compensation was not implemented because of experimental difficulty. Conceptually, the compensation can be performed simply by obtaining through experimentation the values of the compliance and backlash of each joint and then to subtract the computed errors from the measured joint angles as in equation (10). Firstly, when a joint is experiencing lost motion, i.e., when the gear teeth in the joint are positioned between each other without any contact, the joint is not necessarily experiencing zero torque. There is a friction force that must be overcome to force the joint to move even when the gear teeth in the joint are not in contact. This friction force is a function of many factors, for instance, how well the joint has been machined or how much it has been lubricated. Secondly, the robot joint flexibility errors dominate any backlash errors as will be seen from the results. This is because our robot joints were relatively flexible.

These two factors combine to make it extremely difficult to distinguish between the deflection of a joint due to backlash as opposed to the deflection of the joint due to

compliance. Both these deflections are accompanied by finite forces that have relatively small magnitude. The Lord Force-Torque controller has insufficient precision to measure extremely small forces, and besides, it is quite susceptible to noise when measuring small forces. The transition from backlash to compliance is not clear without much more experimentation and online measurements. Therefore, the larger of the two effects, namely compliance, will be considered alone.

The Naka-Nihon robot is a five-axis robot. It has one joint with vertical axis followed by three joints with horizontal axes followed by another joint with vertical axis. This set of joints only allows the orientation of the end-effector on the plane of the arm. This is sufficient for our experiment. A stiff probe is fixed to the end-effector and the inverse kinematics was developed such that the robot always points the probe downwards.

4.1. Experiment (a). To Demonstrate the Presence of Joint Flexibility

The robot is moved around a circular object while exerting a fixed force at points around the contour. The surface normal $\underset{\sim}{m}$ is computed at each point of contact from the steady state force readings. The robot is moved at right angles to the outward surface normal $\underset{\sim}{m} \in \mathbb{R}^{2\times 1}$ (along the $\underset{\sim}{dx} \in \mathbb{R}^{2\times 1}$ direction) as $\underset{\sim}{m} \cdot \underset{\sim}{dx} = 0$. The joint positions are stored for each contact point. Figure 6a shows the shape recovered from the robot joint positions. Shape distortion is quite apparent. Compensation for joint flexibility is then applied, as outlined in section 2, Figure 6b shows the resulting contour. Significant improvement in the recovered shape is seen. This verifies that joint flexibility is a dominant factor in the recovery process. Similar experiment was carried out for a rectangular plate placed on the xy plane. Uncompensated and compensated recovered shapes are shown in Figure 7a and Figure 7b, respectively. The compensated contour is seen to approach the true dimensions. Clearly, some errors remain in the compensated contour. Although vast improvement is seen, the remaining errors are possibly due to

(i) nonlinearity in the joint stiffness (ii) backlash errors and (iii) inexact kinematics. No attempt was made to compensate for the last four effects due to experimental difficulty.

4.2. Experiment (b). To Demonstrate the Effectiveness of the Tracing Algorithm for Shape Recovery

The full algorithm described in section 3.1 is now used, in particular equation (21) is used to compute surface gradients. The robot is moved around a circular disk as in experiment (a). The shape recovered from robot joint position is shown in Figure 8a. Shape recovered in this experiment is significantly improved over that of Figure 6a. Insignificant improvements are observed due to the flexibility compensation, the output of which is shown in Figure 8b. The results of this experiment shows that contact force information is necessary to recover the effects of joint flexibility for accurate shape reconstruction.

Discussion of Results and Conclusion

From experiment (a), it is seen that joint flexibility is present in robot drive train. As a result of this, if the measured joint positions are used to compute the position of the end effector, it would be different from the actual position of contact. The difference in the position is a function of the joint angles and the contact force and other robot parameters. Therefore it is not desirable to compute surface normals and or recover contact position without contact force information. If the steady state contact force is not used erroneous results will be produced because (i) large impact forces may be recorded which is a function of the kinetic energy of the arm. The sensing structure may also be excited during impact, thus further corrupting the recovery process. (ii) Inertial forces due to the motion of the arm can produce transient forces which are not related to the contact force.

A compensation scheme was proposed for shape recovery. This was given in equation (10) it was able to improve contours which did not consider joint flexibility effects.

A kinematic algorithm which is able to guide a robot around a planar contour using force feedback was described in this paper. It was experimentally verified to improve the shape recovery. At the time of writing this paper we do not know of any scheme which is practically verified that is able to dynamically control the contact force and follow an unknown object. Such an algorithm would be practically important in many industrial applications.

Our experience has shown that due to the presence of joint flexibility and backlash and other nonlinearities it is necessary to correct motor-based joint readings for the purposes of computing next trajectory points in a tracing operation or for computing feedback signals, or for estimating surface parameters.

Acknowledgement

We are grateful to Milo Fresein of the Ransburg Corporation of Indianapolis who donated the Lord-Force-torque sensor for our research at the Real-time Robot Control Laboratory, School of Electrical Engineering at Purdue University.

6. List of References

[1] Thiessen, C., "Improving Grinding Performance of an ASEA-Robot by Two Dimensional, Sensor Controlled Path Correction", *Control Problems and Devices in Manufacturing Technology 1980, Proceedings of the 3rd IFAC/IFIP Symposium*, Oct. 1980, T.R.M. Ellis, ed., pp. 265-270

[2] Plank, G. and Hirzinger, G., "Controlling a Robot's Motion Speed by a Force-Torque-Sensor for Deburring Problems", *Information Control Problems in Manufacturing Technology 1982, Proceedings of the 3rd IFAC/IFIP Symposium*, Oct. 1982, D.E. Hardt, ed., pp. 97-102.

[3] Stepien, T. M., Sweet, L. M., Good, M. C., Tomizuka, M., "Control of Tool/Workpiece Contact Force With Application to Robotic Deburring", *Proc. 1985 IEEE International Conference on Robotics and Automation.* pp. 670-679.

[4] Starr, G. P., "Edge-Following with a PUMA 560 Manipulator Using VAL-II", *Proceedings 1986 IEEE International Conference on Robotics and Automation,* Apr 7-10, 1986, pp. 379-383.

[5] Wampler, C., "Multiprocessor Control of a Telemanipulator with Optical Proximity Sensors", *The International Journal of Robotics Research,* Vol. 3, No. 1, Spring 1984, MIT, pp. 40-50.

[6] Whitney, D. E. and Edsall, A. C., "Modeling Robot Contour Processes", *Robotics Research: The Second International Symposium,* ed. Hanafusa, Inoue, MIT Press, Cambridge, Massachusetts, 1985, pp. 163-170.

[7] Ahamd, Shaheen, "Analysis of Robot Drive Train Errors their Effects and Compensation," IEEE Journal of Robotics and Automation, Vol. 4, No. 2, April 1988, pp. 117-129.

[8] Paul, R. P., "Robot Manipulators: Mathematics, Programming and Control," MIT Press, 1981.

[9] Kazerooni, H. Sheridan, T. B., Houpt, P. K., "Robust Compliant Motion for Manipulators, Part I: The Fundamental Concepts of Compliant Motion," *IEEE Journal of Robotics and Automation,* Vol. RA-2, No. 2, June 1986, pp. 93-92.

[10] Kazerooni, H., Houpt, P. K., Sheridan, T. B., "Robust Compliant Manipulators, Part II: Design Method," *IEEE Journal of Robotics and Automation,* Vol. RA-2, No. 2, June 1986, pp 93-105.

[11] Yoshikawa, T., "Dynamic Hybrid Position/Force Control of Robot Manipulators - Description of Hand Constraints and Calculation of Joint Driving Force," *IEEE Journal of Robotics and Automation,* Vol. RA-3, No. 5, October 1987, pp. 386-392.

[12] Zvi, R., Mooring, B. W., Ravani, B., "An Overview of Robot Calibration," *IEEE Journal of Robotics and Automation,* Vol. RA-3, No. 5, October 1987, pp. 377-385.

[13] Wu, C. H., "A Kinematic CAD Tool for the Design of Robot Manipulators," *Int. J. Robotics Res.* Vol. 3, No. 1, Spring 1984.

[14] Hayati, S., "Robot Arm Geometric Link Calibration," *Proc. of 22nd IEEE Conference on Decision and Control,* pp. 1477-1483, Dec. 83.

[15] Veitschegger, W. K. and Wu, C. H., "Robot Accuracy Analysis Based on Kinematics," *IEEE Journal of Robotics and Automation,* Vol. RA-2, No. 3, September 1986, pp. 171-179.

[16] Mason, M. T. "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Trans. Sys. Man. Cybernetics,* SMC-11, No. 6, June 1981, pp. 418-432.

[17] Raibert, M. H. and Craig, J. J. "Hybrid Position/Force Control of Manipulators," ASME Journal of Dynamics Systems Measurement and Control, No. 102, June 1981, pp. 126-133.

[18] Hogan N., "Impedance Control of Industrial Robots," *Journal of Computer Integrated manufacturing,* Vol. 1, No. 1, pp. 97-113, 1984.

[19] Hogan, N., "Impedance Control: An Approach to Manipulation, Part 1: Theory, Part 2: Implementation, Part 3: Applications," *ASME J. of Dynamic Systems Measurement and Control.*

[20] Wu, C. H. and Paul, R. P., "Manipulator Compliance Based on Joint Torque Control," *Proceedings of the IEEE Conference on Decision and Control,* pp. 88-94, Dec. 1980.

[21] Salisbury, J. K., "Active Stiffness Control of Manipulators in Cartesian Coordinates," *Proceedings of the 19th IEEE Conference on Decision and Control,* pp. 95-100, Dec. 1980.

[22] Whitney, D. E., "Force-Feedback Control of Manipulator Fine Motions," *ASME Journal of Dynamics Systems Measurement and Control,* pp. 91-97, June 1977.

[23] Ahmad, S. and Guoh, H., "Feedforward Compensation Approach for Active Stiffness Control of Manipulators," TR-EE 87-11, Purdue University.

[24] Blauer, M. and Belanger, P. R., "State and Parameter Estimation for Robotics Manipulators Using Force Measurements," IEEE Transaction on Automatic Control, Vol. AC-32, no. 12, December 1987, pp. 1055-1066.

[25] Hemami, H. R. and Godard, R., "Recognition of Geometrical Shape by Robotic Probe," Journal of Robotic Systems, 1987, Vol. 4(2), pp. 237-257.

[26] Huang, H. P., McClamroch, N. H., "Time Optimal Control for a Robotic Contour Following Problem," IEEE Journal of Robotics and Automation, Vol. 4, no. 2, April 1988, pp. 130-140.

[27] Rivin, E. I., "Effective Rigidity of Robot Structure: Analysis Enhancements," Proceedings of the 1985 American Controls Conference, Boston, June 1985.

[28] Good, M., Strobel, K., Sweet, L. M., "Dynamics and Control of Robot Drive Systems," General Electric Company, 1983.

[29] Spong, M. W., Khorasani, K., Kokotovic, K., "An Integral Manifold Approach to the Feedback Control of Flexible Joint Robots," IEEE Journal of Robotics and Automation, Vol. 30, No. 4, Aug. 1987.

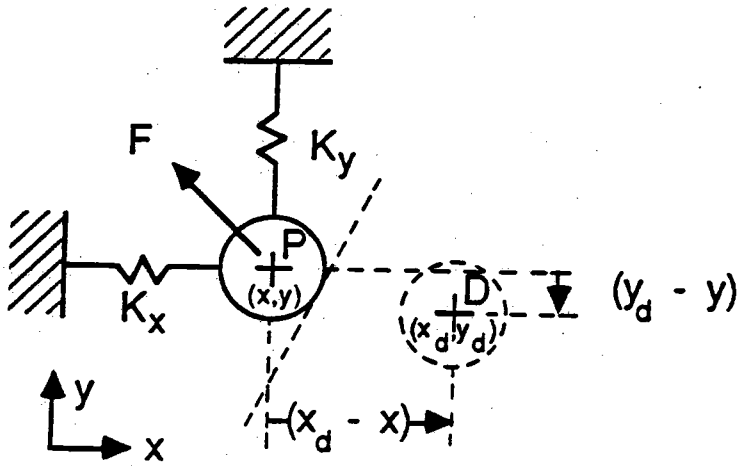


Figure 1. Small Force Generation Model with Joint Flexibility

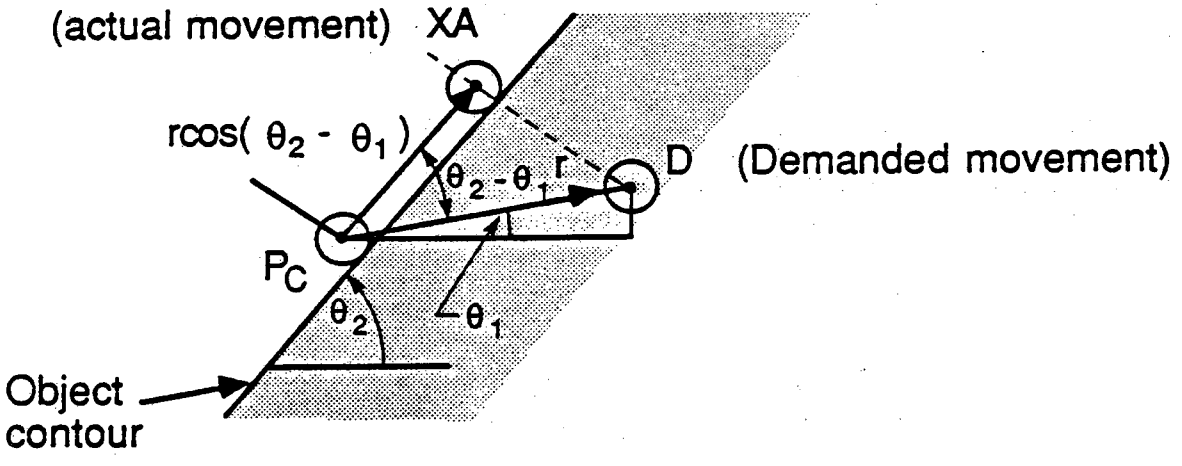


Figure 2. Displacement Along the Contour.

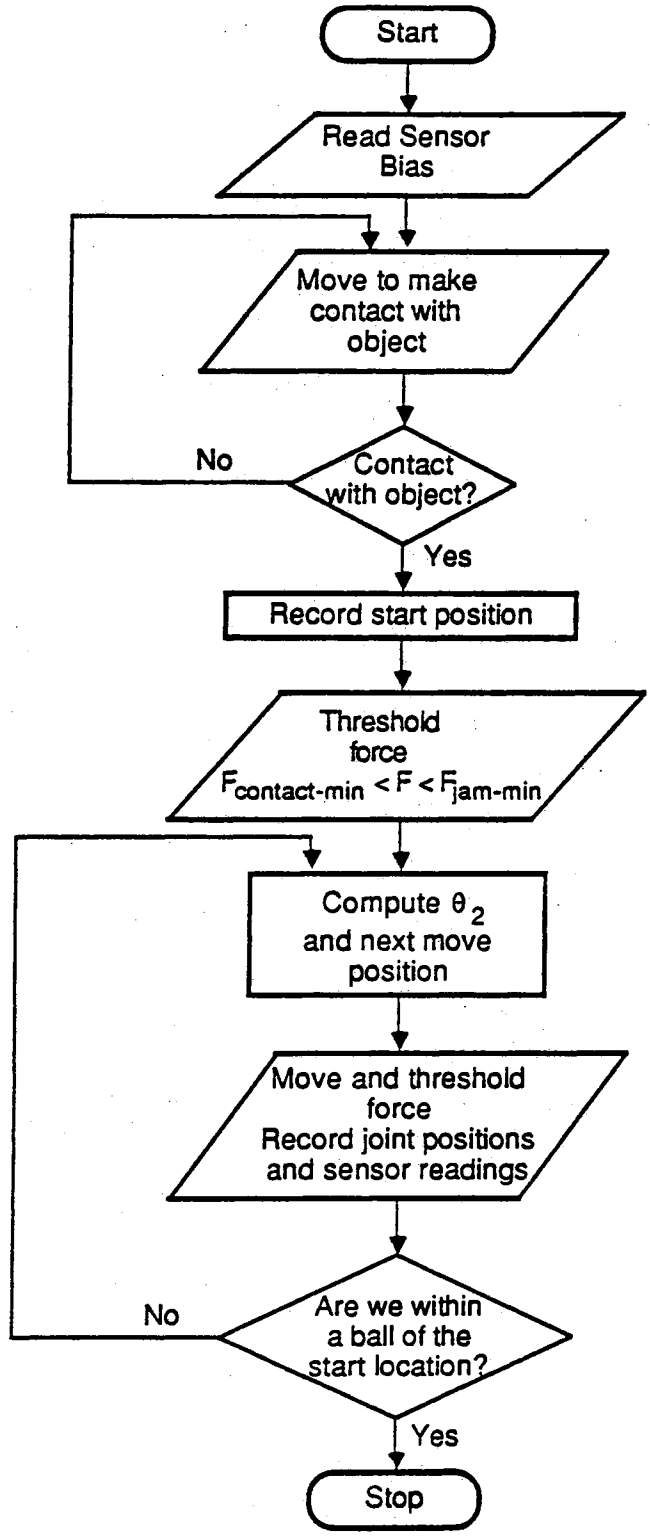


Figure 3. Algorithm to Trace an Unknown Contour.

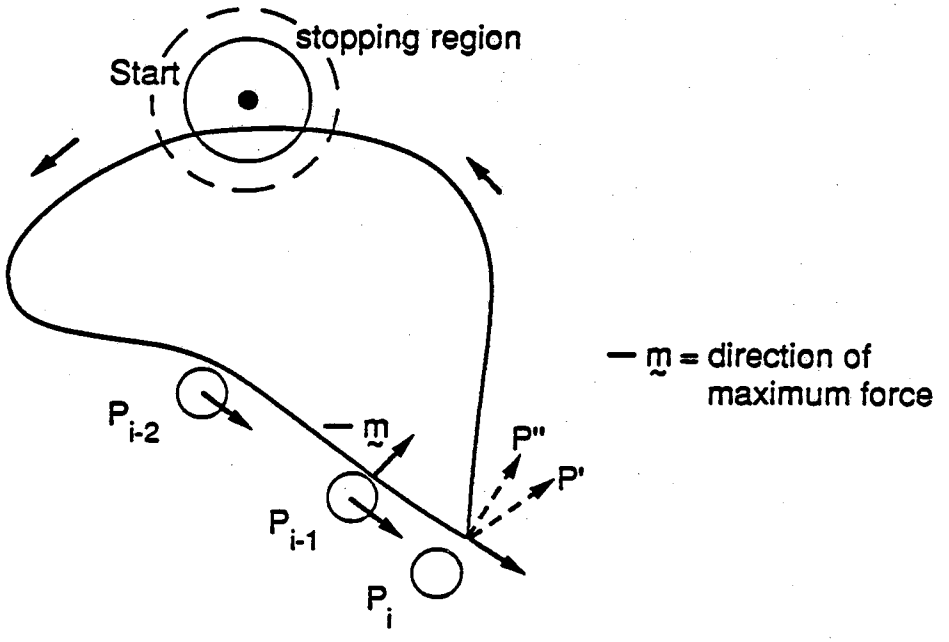


Figure 4. Moving Around Discontinuities Along the Contour.

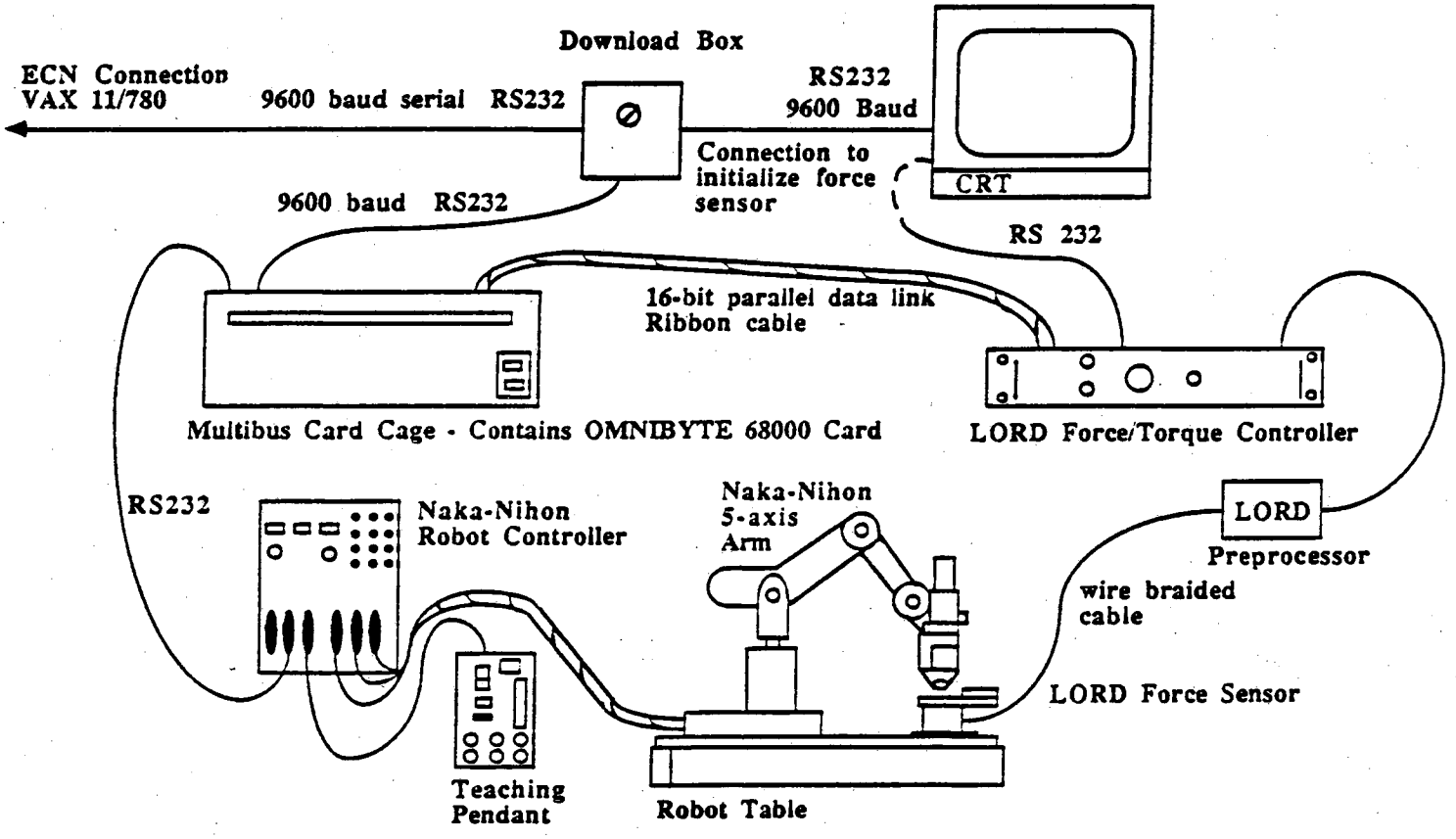


Figure 5. Contour Tracing Experimental Apparatus.

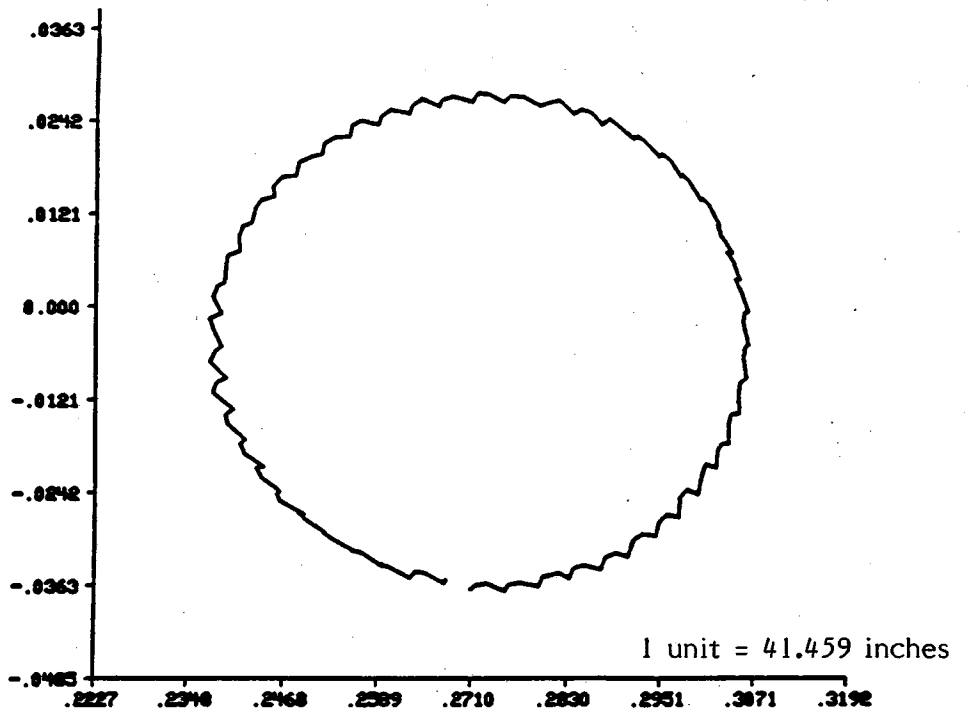


Figure 6a. Circular Object Recorded from Joint Angles.

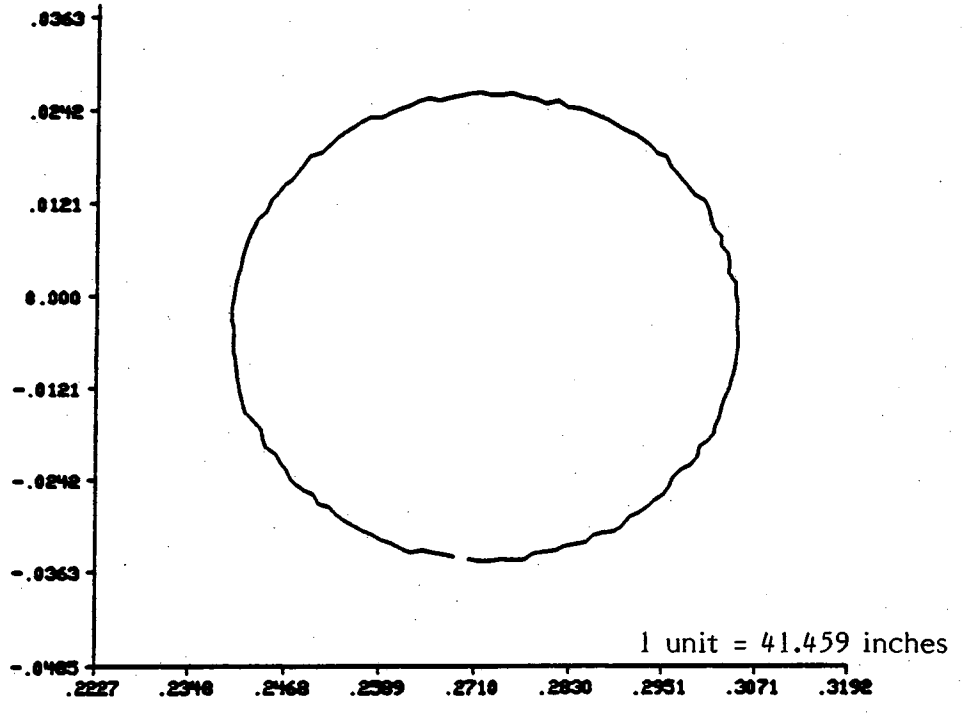


Figure 6b. Circular Object After Compensation for Joint Flexibility (using equation (10)).

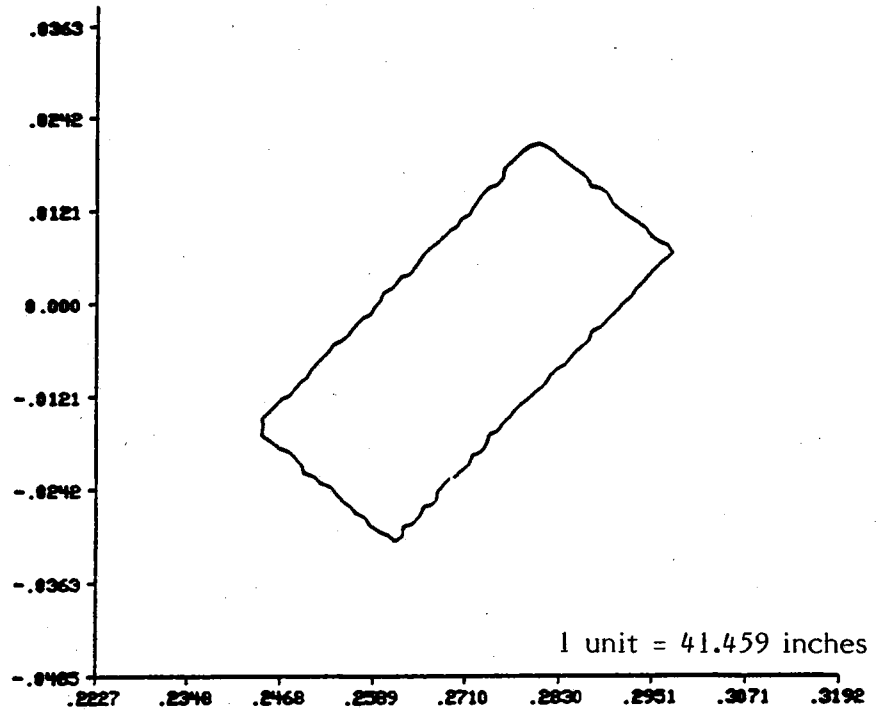


Figure 7a. Rectangular Object Before Compensation.

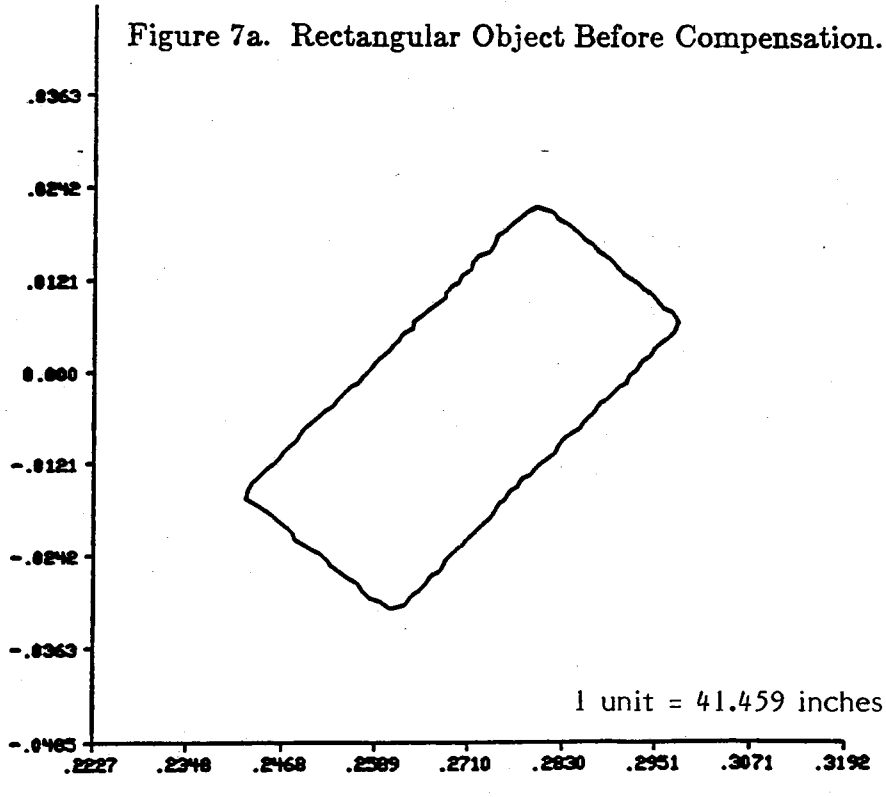


Figure 7b. Rectangular Object After Compensation for Joint Flexibility (using Equation 10)

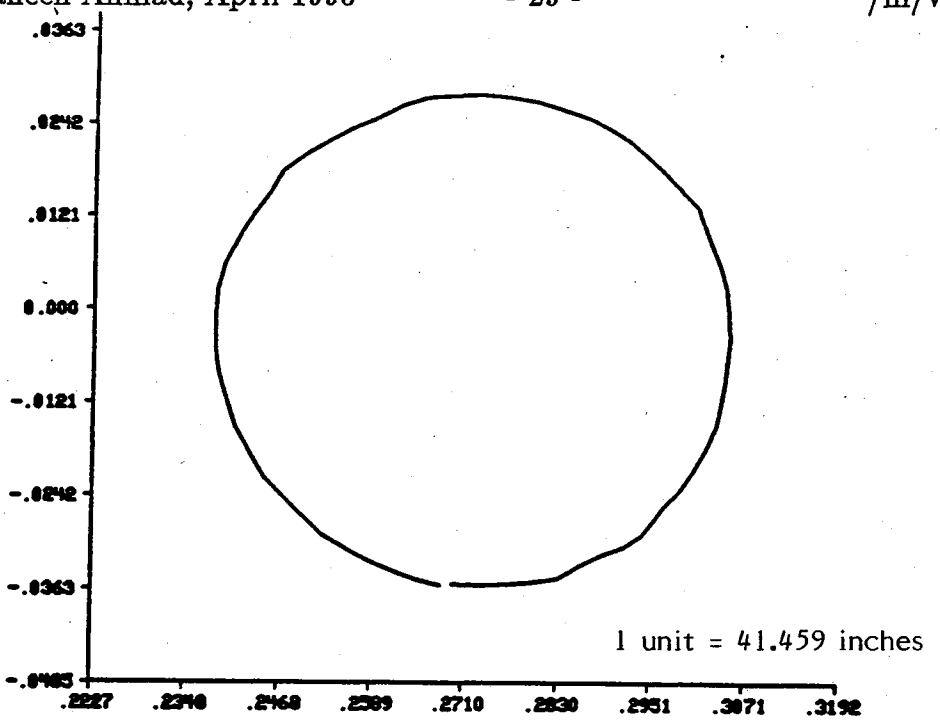


Figure 8a. Uncompensated Shape Recovered form the Tracing Algorithm, with Flexibility Compensation (equation (21)).

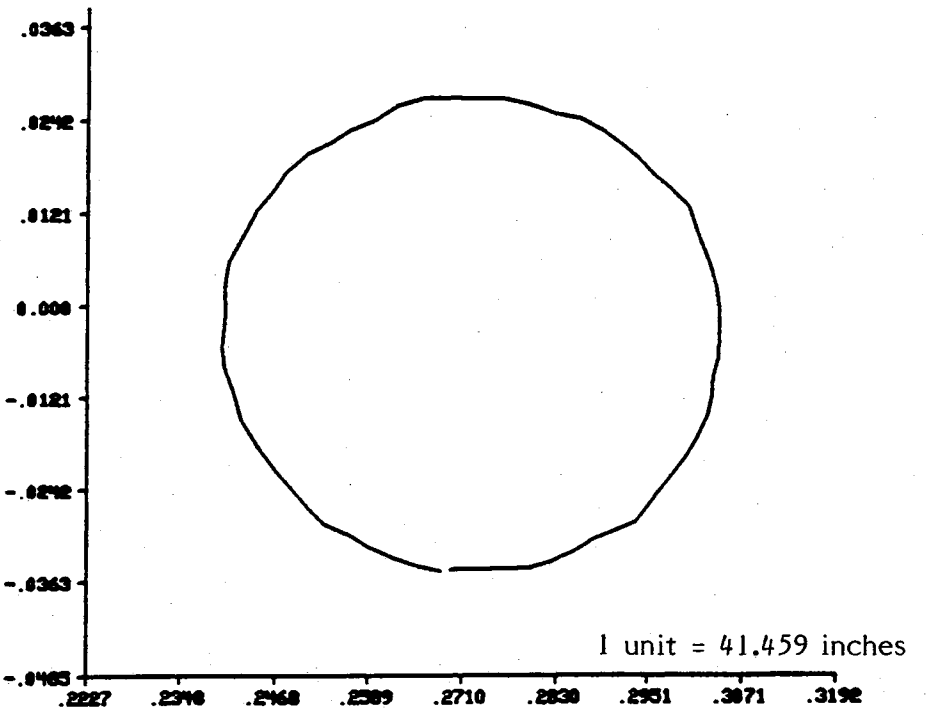


Figure 8b. Compensated Shape.
(Using equation 10)