THE UNIVERSITY OF CALGARY

"Performance Limitations of a Force-Controlled Industrial Manipulator"

by

Yajaira Herrera

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL AND MANUFACTURING ENGINEERING

CALGARY, ALBERTA

FEBRUARY, 1999

ABSTRACT

In this thesis, system performance is investigated in the case where forces are applied by a robotic manipulator to a constraint. The forces are controlled using the hybrid position/force control method. Force control is achieved by adjusting the end effector position normal to the contact surface based on the feedback received from a wrist-mounted force sensor. The relationship between force control accuracy and system stiffness is derived and then verified experimentally. A computer algorithm is proposed for tracking and measuring flat and curved surfaces with unknown dimensions based on the force control approach. The algorithm is coded into a robot control language and tested experimentally by characterizing the performance of a two degree of freedom Adept manipulator. Factors affecting the performance of the system are identified. It is shown that the robotic manipulator tracks the constraint faster when the force tolerance to reach the desired normal force to the constraint is greater than the force resolution of the system. Therefore, there is a tradeoff between tracking speed and tracking accuracy. It is also shown that the more deformation the constraint experiences, the less accurate the constraint measurements are.

ACKNOWLEDGEMENTS

Many people have contributed to the fulfillment of this thesis. My supervisor, Dr. P. B. Goldsmith, was always available to answer my questions and discuss with me technical problems that arose during the completion of the research. He was also very patient and understanding when my mind was more focussed on my personal problems than on my work.

I would like to thank Mike for his promptness and unconditional help whenever I had mechanical problems with the robot in the laboratory. He also dedicated his time to cutting all the materials I used for the experiments.

My parents and sister always showed their loving support and helped me endure this stage of my career.

Finally, but not least, all my appreciation goes to Norm. He was always there for me when I needed guidance, support, a friendly hand, and love above all. He had the hard task to put myself back together when I needed it the most.

FOR NORM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 WHAT IS HYBRID POSITION/FORCE CONTROL?

Industrial robots are widely used in different branches of industry. Some of their applications are spray painting, arc and spot welding, injection moulding, tool loading, handling and palletizing. All these applications are simple and repetitive, and what they all have in common is that these robots follow only position commands for the execution of these tasks.

On the other hand, applications such as grinding, deburring, deflashing, polishing, and assembly require the control of not only the position of the manipulator, but also the force applied by the robot on the constraint. Hybrid position/force control is the control of the force in constrained directions and the simultaneous control of the position in unconstrained directions.

It is suggested here that hybrid position/force control can be implemented using either the direct method or the indirect method. The direct method entails the direct control of the torques exerted by each joint actuator to control simultaneously the force and the position, whereas the indirect method requires commanding positional corrections in constrained directions to control the force. The indirect method is suitable

for industrial robots because their controllers are designed to accept position commands. It is hypothesized that indirect hybrid position/force control is a feasible method to control the force applied by a robotic manipulator using industrial controllers. This study examines the performance of the indirect method experimentally.

1.2 RESEARCH PURPOSE

The main objective of the research underlying this thesis is to determine the performance of the indirect hybrid position/force control method and develop algorithms for contour tracking. In order to achieve these, the following subgoals are proposed:

- Formulate indirect hybrid position/force controls.
- Identify the factors limiting the performance of the system.
- Identify tradeoffs in performance characteristics.
- Design and test force control algorithms for contour tracking of a surface.

1.3 METHODOLOGY

This research is divided into five chapters. In Chapter 2, a brief review of the literature pertaining to the simultaneous control of forces and positions is presented. Chapter 3 defines two methods to control simultaneously the position and the force of any robot arm. These methods are direct hybrid position/force control, and indirect hybrid

position/force control. It also deals with the dynamics of the system under study, and identifies theoretically each of the factors affecting the performance of such a system.

In Chapter 4, the experimental setup is described. The feasibility of the indirect hybrid position/force control method is studied via various experiments which use wood, wax, and foam as the constraints. Such experiments consist of:

- The determination of the stiffness coefficient of the materials used.

- A one degree of freedom force control experiment that consists of the application of a constant force by the manipulator along the constraint axis.

- A non-contact experiment which determines the real position resolution of the system and therefore, the force resolution of the system.

- Contour tracking force control experiments that investigates the performance of the system when the manipulator tracks the contour of a flat and a curved surface.

The purpose of the experiments is to determine if indirect hybrid position/force control is feasible and to identify all the limitations affecting the performance of the system. Finally, the conclusions of the research are stated in Chapter five.

1.4 PROPOSED CONTRIBUTIONS

The main contributions of this research are:

- An explanation of the differences between direct hybrid position/force control and indirect hybrid position/force control.
- A theoretical formulation of the indirect force control method, a technique for which industrial application notes mainly exist.
- A demonstration of the feasibility of indirect hybrid position/force control.
- An identification of each one of the factors limiting the performance of the system for both the force control and the contour tracking experiments.
- An experimental investigation of robot and constraint stiffness.
- A verification that when using rigid constraints, the manipulator is the compliant element in the system due to the flexible joints.
- An alternative to the problem of controlling force in robotic manipulators with industrial controllers.
- An experimental determination of the position resolution of the system, from which the force resolution was computed.
- An experimental verification of the computed force resolution.
- Computer algorithms required to execute the one degree of freedom force control experiments and the contour tracking of either a flat or a curved surface.

The contour tracking algorithm of a curved surface provides customize inputs such as interpoint speed control, force applied by the robot, and force tolerance that allow the manipulator to track the contour of the constraint at a specified velocity when applying a constant force within the chosen force tolerance.

The contour tracking algorithm of a curved surface provides the means to graph the contour of the object, and determines information related to its dimensions such as arc length, etc.

CHAPTER 2

LITERATURE REVIEW

Broad is the literature related to position and/or force control of manipulators, in this review only those papers related to the scope of this research are presented.

All of these papers are oriented toward the derivation of position/force control algorithms to be implemented in diverse manipulator controllers specially designed for that purpose. These papers support the Direct Hybrid/Position Force Control method explained in Chapter 3, Section 3.1.1. Based on the literature survey, we conclude that research on indirect hybrid position/force control is rare. There are several papers that present industrial applications of the indirect method, but they neglect a theoretical formulation. This is the main reason why we are conducting this study.

Controlling the force applied by a robot manipulator has been the effort of many researchers who have proposed several methods in order to attain this goal, amongst other the main ones are:

Joint compliance: [Paul, 1976] present a control scheme based on compliance which is controlled in a Cartesian coordinate system. It changes the servo modes of the manipulator among position, velocity and force, controlling the joints individually. [Kazerroni, 1989] proposed a nonlinear approach for the control and stability analysis of

manipulative systems in compliant maneuvers. He shows that there must be some initial compliance either in the robot or in the environment for the stability of the robot. [Mason, 1981] presents a theory of force control based on forward models of the manipulator and the task geometry. Such models provide a precise semantics for compliant motion primitives in manipulation programming languages. [Chiaverini, 1994] discussed stable force/position regulations of robot manipulators in contact with an elastically compliant surface using a PID controller. [Whitney, 1977] stated that vector motion commands in the environment coordinate system can be deduced from vector force measurements reflected into the same coordinate system. It uses Jacobian matrices to multiply the measured forces and produce the required velocities directly in the environment coordinates.

Impedance control and Active stiffness: [Hogan, 1985] stated that force and position cannot be controlled simultaneously, but by controlling the amount of impedance in the manipulator, the contact forces can be regulated. In other words, positions are commanded and impedances are adjusted to obtain the proper force response. The controller attempts to implement a dynamic relation between manipulator variables such as output force and input displacement, whose parameters (inertia, damping ratio and stiffness coefficients) can be specified by the operator. [Anderson, 1988] suggested a method that improves the robot's adaptability to uncertain environments combining inverse dynamics, impedance control and hybrid control. [Goldenberg, 1991] implemented the robust hybrid impedance control method where inertia and damping are

introduced in the force controlled subspace to improve the dynamic behavior, and impedance control is used in the position controlled subspace. [Salisbury, 1980] designed an active stiffness control of a manipulator in Cartesian coordinates based on the known constraint conditions in the assembly task, the stiffness specifications can be changed using a high level language to match tasks and prevent part jamming.

[Spong, 1989] suggested a stiffness control method that uses a position control scheme similar to the one proposed in this thesis. The system that Spong and Vidyasagar described is governed by two kind of forces, one related to the mass and acceleration of the links and joints of the robotic manipulator and the other related to position and stiffness coefficient of the constraint. Their position control scheme is based on a desired position that is commanded slightly inside the constraint, which is really an unknown position. They also used two different kinds of gains, the stiffness of the manipulator and the stiffness of the constraint, in their PD control law. The main differences between their work and the control law proposed in this thesis are that we propose only a proportional control law and that we do not take into account the acceleration of the joints. We chose a P control law because the industrial controller of the robotic manipulator has already a PID control structure.

Hybrid position/force control: [Raibert, 1981] proposed a hybrid controller to control the position of the manipulator along the tangential axis and the force exerted by the end effector on an object along the normal axis. [Yoshikawa, 1981] and [Yoshikawa,

1988] proposed the dynamic hybrid control method, which takes the manipulator dynamics into account and is an extension of the hybrid control approach proposed by Raibert and Craig. [Ohto, 1991] suggested a hybrid position/force control scheme for manipulators with position controllers. They assumed that the dynamic model of the manipulator is exactly known, that the joint variable vectors are measurable, and that the external force/moment vector acting at the end point of the manipulator is also measurable by a force/torque sensor. Then, they designed a joint force/torque vector such that it would compensate all the nonlinear and coupling terms and computed force/torques, that uses a feedback control law with trajectory commands, position and velocity error feedback matrices. [Matsuno, 1991] designed a controller for the hybrid position/force control of flexible manipulators taking into account elastic deformations, dynamic equations of joint angles, vibrations of the flexible link and constraint force.

Another research branch is dedicated to contour tracking control using applied forces on constraints. In 1994, [Hu, 1994] proposed a descriptor system model to control forces and position tracking of robotic manipulators. They presented a method to globally linearize a nonlinear descriptor system arising from a manipulator in contact with a constraint surface. [Yao, 1998] used adaptive robust control to solve the motion and force tracking control of manipulators in contact with unknown stiffness environment. [Harris, 1988] presented global conditions for tracking based on a modified computed torque controller and local conditions for feedback stabilization using a linear controller. [Qu, 1993] proposed a first input-output robust control design for the trajectory following

problem of a robotic manipulator. It requires position and velocity feedback to ensure global stability.

The applications of force control are numerous and relevant. The main difference between these applications and the results obtained from the experiments conducted with the industrial robotic manipulator used is that these researchers designed both different control laws and special end effectors to carry out their experiments. [Bone, 1991] proved that by controlling the normal force a consistent chamfer depth can be obtained when using an active end effector for robotic deburring. He also developed with Elbestawi [Bone, 1991] an active end effector control of a low precision robot in deburring. [Kazarooni, 1986] developed a control strategy for precision deburring to guarantee burr removal, while compensating for robot oscillations and small uncertainties in the location of the part relative to the robot. He also presented with Her [Kazarooni, 1991] a method for robotic deburring of two-dimensional planar parts with unknown geometry using compliance control. [Whitney, 1990] implemented a nonlinear force control law and a light vision system for an automated robotic weld bead grinding system able to plan and execute a multi-pass grinding sequence for removing a weld bead. [Bausch, 1996] designed an application of a brush-based active robotic deburring system that is based on indirect hybrid/position force control. Other papers are oriented toward the medicine field. Fujie developed the use of robotic technology to study human joint kinematics [Fujie, 1996], in-situ force of ligaments [Fujie, 1993], and friction in animal joints [Fujie, 1995].

The present research project utilizes an industrial robot for the experimental stage, and its commercial controller has not been altered in any way. Therefore, we cannot modify the control law designed by the robotic manipulator manufacturer. In this research underlying this thesis, we designed a control law that works in conjunction with the manufacturer position control law and study the performance of the system, as will be explained in the next chapters.

CHAPTER 3

THEORETICAL BACKGROUND

This chapter defines two methods to control simultaneously the position and the force of an industrial robot. These methods are direct hybrid position/force control and indirect hybrid position/force control. It also deals with the dynamics of the theoretical system of study and identifies each of the factors affecting the performance of such a system.

3.1. HYBRID POSITION / FORCE CONTROL METHODS

The system of study is the constrained planar robot shown in Figure 3.1. The robot has two links, two revolute joints, and an end effector in contact with a surface. The vector X denotes the position of the end effector with respect to the base frame, and x and y are its world coordinates. Such a system has only two degrees of freedom, meaning that the robot can only move along the x and y axes in the world coordinate system. The general case would be represented by a robot with "N" degrees of freedom, but due to the configuration of the system available for the experiments, only two degrees of freedom will be considered.

Hybrid position/force control refers to the simultaneous control of the normal force F applied by the robot on a constraint and the position y along the unconstrained

axis. The position of joint i is denoted by q_i , and the actuator forces exerted by each joint are denoted by τ. In prismatic joints q is measured in degrees and in revolute joints q is measured in mm. To control the position, the joint encoders of the manipulator measure continuously the position q of the joints. To control the force, two methods are proposed and described below.

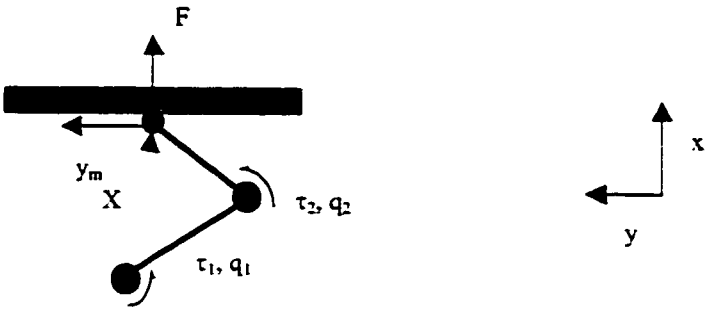


Figure 3.1. A constrained robot.

3.1.1. DIRECT HYBRID POSITION / FORCE CONTROL

Ideally, one would control directly the torques τ exerted by each joint actuator. This control method will be named "Direct Hybrid Position / Force Control" and is depicted in Figure 3.2.

From Figure 3.2, it can be seen that the force F_m measured by the force/torque sensor is compared to the desired force F_d to calculate the force error e_f. The joint encoders read the position q at each joint. Then, using forward kinematics, these are transformed into the measured position X_m, whose y_m coordinate is compared to the

desired position y_d to find the position error e_y. Then the errors e_f and e_y are used by the control law to determine the new values of the actuator forces τ required to make the errors tend to zero. The position "y_m" refers to the displacement of the manipulator along the unconstrained axis, and the force "F" is the force applied by the robot to a constraint along the x axis. This is the reason why there are no desired position x_d and measured position x_m considered along the constrained axis.

Theoretically, this ideal system should have a fast sampling rate, because the checking and correction of its position and force should be as fast as the checking and correction of only the position by industrial controllers.

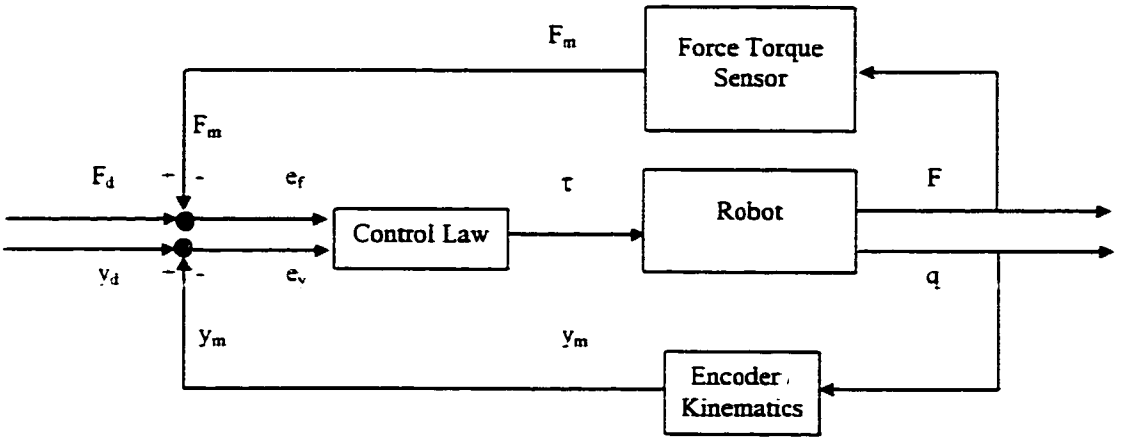


Figure 3.2 Direct Hybrid Position Force Control Block Diagram.

It is important to note that the data measured by the sensor and the encoders (F_m and y_m) and the real data resulting from the movement of the manipulator (F and q) are not the same due to the dynamics of the force/torque sensor, accuracy error of the

encoders, round-off errors during the homogeneous transformations of the vectors, noise, vibrations of the robot's arm, compliance of the flexible joints and inertia of the manipulator.

This hybrid position/force control method uses force and position to calculate the actuator forces τ. To achieve this, a special controller would have to be built. Because this customized controller is not out in the market place and because commercially available controllers control only position, a method to control the force by controlling the position is shown in Figure 3.3, and this control method is named "Indirect Hybrid Position/Force Control".

3.1.2. INDIRECT HYBRID POSITION / FORCE CONTROL

The dotted box of Figure 3.3 shows the control method used by a commercial controller. It compares the desired position increment x_c , and the position x_m and y_m obtained by applying kinematics to the readings q of the joint encoders. Then the position error e_x is introduced into the feedback law of the commercial controller in order to determine the new values of the actuator forces τ that would generate a position as close as possible to the desired one. In this inner loop, the desired position along the x axis is identified with the subscript "i" because, it is the position increment calculated by the computer program.

The commercial controller does not allow one to specify directly the actuator forces τ. Therefore, in order to control the force applied by the robot, it is necessary to create an "outer force control loop" that feeds back the measured force F_m from the force/torque sensor to a computer program. This input is then used to calculate the new desired position based on the force error and the stiffness coefficient of the system.

The computer program is the key element of this control method proposed. It determines the difference between the desired force F_d to apply to a constraint, and the force F_m measured by the force/torque sensor. Then, this force error is multiplied by a gain K in order to find the new calculated position increment x_i that would be used to reach the desired force. Once again, this measured force F_m is not identical to the real force F as explained in section 3.1.1, but it is very close – the difference can be appreciated in the 2nd and 3rd digits after the decimal place -. The input y_d is an absolute position introduced directly into the commercial feedback law to command the desired displacement of the manipulator along the unconstrained axis.

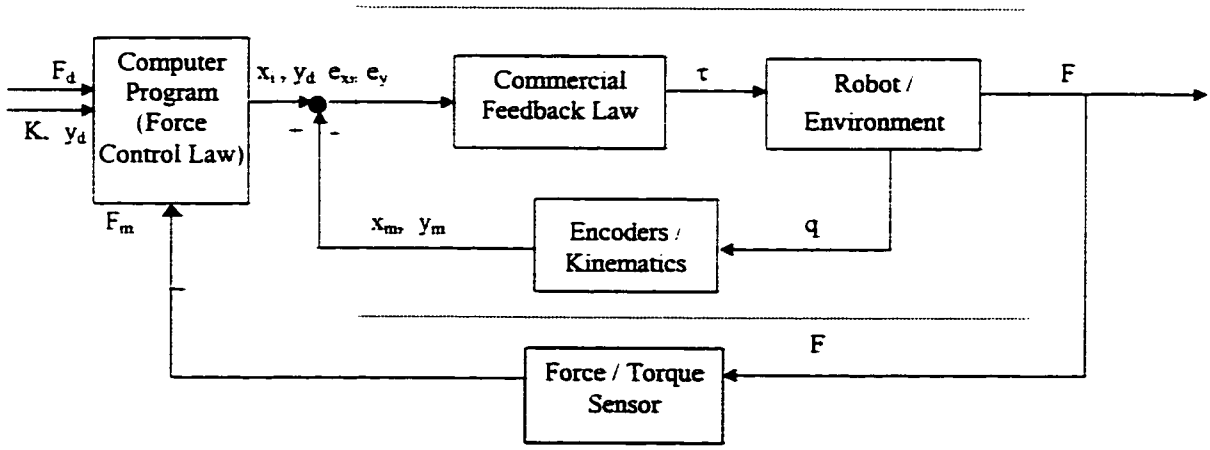


Figure 3.3 Indirect Hybrid Position Force Control Block Diagram.

The desired position y_d, the desired force F_d, and the gain K are the only input variables to the commercial feedback law. To determine the new position that the robot should reach theoretically in one step, the force error, F_d minus F_m , has to be multiplied by an adequate gain. The gain has been chosen to be the inverse of the stiffness coefficient of the constraint K_s, in order to optimize the number of iterations required to reach the desired force in one movement. The stiffness coefficient of the constraint K_s depends on the material and its geometry. The former is the control law applied in the computer program, and is quantified by:

$$x_i = (F_d - F_m) / K_s \ . \qquad (1)$$

It is important to mention that this control method is called Proportional Control. Neither PD Control nor PDI Control was used because the industrial controller already uses PDI Control in their control law –inner loop-.

Figure 3.4 depicts the deformation of the constraint when the robotic manipulator applies a force. The deformation of the surface is equal to the force read by the force sensor divided by the stiffness coefficient of the constraint. When the manipulator applies a force to the constraint, the position read by the joint encoders is given with respect to the base frame based on the tool's z axis (p^t). Therefore, if the real position of the constraint (p^c) has to be known, we have to add to the position of the tool, the radius (r) of the tool and subtract the deformation of the constraint (Figure 3.5).

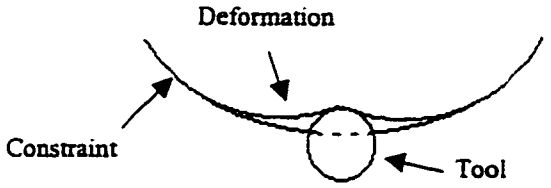


Figure 3.4 Deformation of the constraint

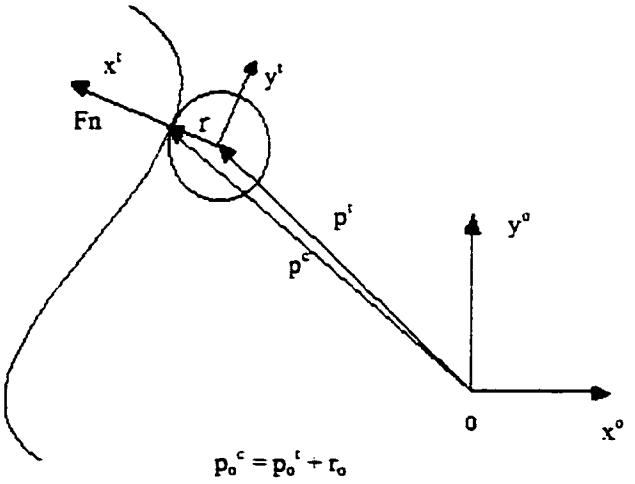


Figure 3.5 Position of the constraint.

This system involves two different sample frequencies, the sample frequency of the inner loop (i.e. the controllers position feedback loop) and the sample frequency of the outer loop (i.e. our force control program). The sample frequency of the inner loop is very high and it is provided by the manufacturer of the robot. In our case, it is 500 cycles per second. This means that every second, the joint encoders check 500 times the position of each joint to make corrections as necessary. On the other hand, the sample frequency

of the outer loop is very low compared to the former one. It is typically about 20 cycles per second and depends on the computer program. There are several reasons why this sample frequency is low; among them we have: delays in the computer program, length of the control program, architecture of the program, time required to move the manipulator, time required to read the forces by the force/torque sensor, etc. Therefore, it is very important to improve the efficiency of computer program as much as possible in order to reduce the response time of the system.

In order to explain the goal of reaching the force within a certain tolerance, let us assume that the manipulator is not in contact with any constraint and that it is not moving. If we ask the force sensor to read the forces applied by the robot on a constraint several times, every single reading will be different. This difference is shown in variations in the first and second decimal place. This is due to the dynamics and the noise of the force sensor [Adept, 1989], and it occurs whether or not the manipulator is in contact with a constraint. Therefore, if we command the manipulator to reach a desired force, we have to take into account that the readings from the force sensor are not as accurate as desired. Consequently, a force tolerance must be established to compensate for such force reading variations.

The force resolution of the system is another factor that affects the performance of the system, and this aspect will be explained through an example. When one multiplies

the stiffness coefficient (N/mm) by a position increment (mm) that deforms the constraint, the result is a force increment (N).

$$\Delta F = K_S * \Delta x \qquad (2)$$

Where
- ΔF is a force increment,
- K_s is the stiffness coefficient of the constraint, and
- Δx is the position increment.

If we multiply the stiffness coefficient K_s with the minimum positional increment of the robot, which is called the position resolution, we obtain the force resolution of the system. The position resolution of the robot is 0.013 mm (provided by the manufacturer) and the stiffness coefficient of the piece of wood as a constraint was found to be 50.364729 N/mm.

$$FR = K_s * x_r \qquad (3)$$

$$FR = 50.364729 \text{ N/mm} * 0.013 \text{ mm}$$

$$FR = 0.654741 \text{ N}$$

The minimum force increment that the robot can achieve when applying a force on such constraint is 0.654741 N. The reason why the force resolution affects the performance of the system is that a case might be given in which it is necessary to apply a force correction lower than the force resolution of the system to obtain the desired force. Therefore, since the robot is unable to reach such desired position, the system becomes slower because the robot will keep moving forward and backward along the normal axis until the force is reached, which will never be reached unless there is noise.

It is hypothesized that the compliance in the system is provided not only by the constraint, but also by the joints of the manipulator. If the robot arm is in contact with a very stiff surface and let us say that it is commanded to move along the constrained axis. the motor pulleys of the joints will be the elements supplying such compliance.

Another limitation of the system is that the noise of the noise filter parameter chosen in the computer program that sets up the force/torque sensor, will affect the sampling frequency of the system. The "Force Sensing Module - User's Guide" specifies that the higher the noise filter parameter, the slower the system becomes because it requires more time to detect contact [Adept, 1989]. Therefore, setting an adequate noise filter parameter in the computer program is an important task to reduce these effects.

3.2. SYSTEM DYNAMICS

Figure 3.6 depicts a vector X in the plane X-Y that goes from point 0 to point n.

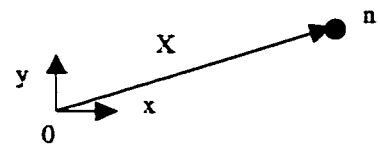


Figure 3.6 Vector position in X-Y plane.

The frame 0(x,y) represents the base frame of the planar robot, and the point n represents the location of the end effector, which is a function of the joint variables q and can be expressed in world coordinates as:

$$X = \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4}$$

If we take the first derivative of the position, we obtain the velocity, which is represented in figure 3.7, and quantified in the equation (5).

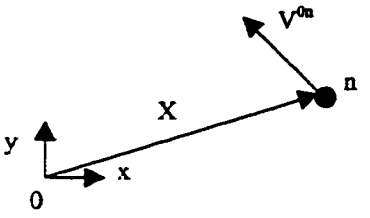


Figure 3.7 Vector velocity in X-Y plane

$$V^{0n} = \dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}. \tag{5}$$

Now,

$$\dot{X} = \frac{dX(q)}{dt} = \frac{\partial X}{\partial q}\frac{dq}{dt}, \tag{6}$$

or

$$\dot{X} = \frac{\partial X}{\partial q}\dot{q}, \tag{7}$$

which is equal to

$$\left.\begin{aligned} \dot{x} &= \left(\frac{\partial x}{\partial q_1} \quad \frac{\partial x}{\partial q_2}\right)\dot{q} \\ \dot{y} &= \left(\frac{\partial y}{\partial q_1} \quad \frac{\partial y}{\partial q_2}\right)\dot{q} \end{aligned}\right\}. \tag{8}$$

Let J(q), the manipulator Jacobian be defined as

$$J(q) = \frac{\partial X(q)}{\partial q}. \tag{9}$$

From (7) and (9) we obtain

$$\dot{X} = J(q)\dot{q} \ . \tag{10}$$

From (10) we have

$$\dot{q} = J(q)^{-1}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}. \tag{11}$$

Its second derivative is

$$\ddot{q} = \dot{J}(q)^{-1}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} + J(q)^{-1}\begin{pmatrix}\ddot{x}\\ \ddot{y}\end{pmatrix}. \tag{12}$$

The dynamic equation that defines the constrained system in joint coordinates is given by [31]:

$$\tau = D_q(q)\ddot{q} + h_q(q, \dot{q}) + J(q)^T F \ , \tag{13}$$

where

$D_q(q)$ is a 2 x 2 matrix of motor and link inertias,
$h_q(q, \dot{q})$ is a vector of centrifugal and Coriolis terms,
$J(q)^T$ is the transpose of the Jacobian, and
F is a matrix of forces applied by the robot's tool.

In this particular case, the robot is set to move at very low speed rate across the X-Y plane. Therefore, the vector of Coriolis and centrifugal forces can be neglected. The forces exerted along the Y axis can be ignored because the desired force is to be applied only along the X axis. The compliance at the contact point between the tool and the object is modeled by a spring in Figure 3.8 and it is quantified as:

$$F_x = K_s * x \qquad (14)$$

Where

F_x is the force exerted along the x axis
K_s is the spring coefficient, and
x is the distance.

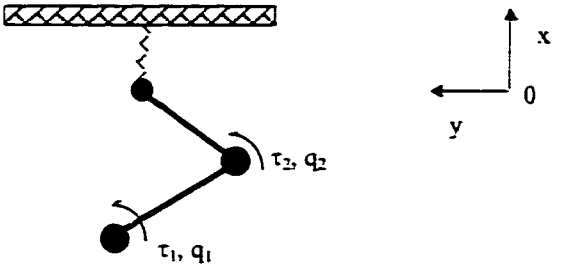


Fig. 3.8 Planar robot in contact with a compliant surface

Substituting (11) and (12) into (13) and then premultiplying by $J(q)^{-T}$, and neglecting the terms $h_q(q,\dot{q})$ and $J_{(q)}^{-T} D_q(q) \dot{J}_{(q)}^{-1} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$ yields:

$$J(q)^{-T} \tau = D(x,y) \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} + F , \qquad (15)$$

where

$$D(x,y) = J_{(q)}^{-T}\, D_q(q)\, J_{(q)}^{-1} . \qquad (16)$$

Here

$D(x,y)$ is a matrix of motor and link inertias expressed in world coordinates,

F is a vector of forces applied by the robot, and

$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix}$ is a vector of accelerations.

The force applied by the robot to the constraint is

$$F = \begin{pmatrix} F_x \\ F_y \end{pmatrix}. \qquad (17)$$

Let us define

$$u = J(q)^{-T}\, \tau . \qquad (18)$$

Substituting (17) and (18) into (15) yields:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} + \begin{pmatrix} F_x \\ F_y \end{pmatrix} . \qquad (19)$$

The vector u quantifies the forces applied by the robot to a constraint. This vector is an input that relates the value of the actuator forces τ and position q of the joints to the acceleration of the joints in world coordinates. When the acceleration equals zero (the static case) u will be equal to the forces applied along the x and y axes.

From equation (19) we obtain

$$F_x = u_1 - d_{11}\,\ddot{x} - d_{12}\,\ddot{y}\,. \qquad (20)$$

Equation (20) quantifies the force applied by the robot on a constraint along the x axis. It can be seen from this equation that although the force applied is along the x axis, the acceleration of the manipulator along the y axis affects the readings of the force sensor. From equation (19) the same analysis can be made for the forces along the y axis. Therefore, it is hypothesized that the acceleration along the y axis and the static friction will produce force readings along the y axis of the force sensor even when the manipulator is commanded to move only along the x axis. More precisely, the acceleration along the y axis will generate a slight displacement of the manipulator along the same axis and the static friction will keep the manipulator in that location after the acceleration becomes zero, thus producing a force along y.

3.3 ONE DEGREE OF FREEDOM FORCE CONTROL

In this case, the manipulator is commanded to move along a constrained axis. Once there is contact between the manipulator and the constraint, the force sensor reads the force applied by the robot. That measured force is compared to a desired force and a force error will then be defined. This force error is divided by the stiffness coefficient of the constraint, $\bar{K}_s$, defined previously in section 3.1.2, to determine the position increment correction that will generate a force as close as possible to the desired one. The pseudo code for the algorithm used to control the force is:

```
1 Define control parameters
2 Make contact
3 Control loop:
        DO
                Take force readings
                Find force errors
                Find position increment
                Correct trajectory along robot's normal axis
         UNTIL desired force reached
4 End
```

3.4 CONTOUR TRACKING FORCE CONTROL

The process of contour tracking a surface will be considered into two different cases: the contour tracking of a flat surface and the contour tracking of a curved surface.

The first case is very straight forward using indirect hybrid position/force control and the Alter mode instruction, which will be explained in Section 4.3.1.1. Once the desired force is reached along the normal axis using the one degree of freedom force control (see Section 3.3.), the manipulator is instructed to make constant displacements along the unconstrained axis. The pseudo code that explains this method is:

```
1 Define control parameters
2 Make contact
3 Control loop:
```

```
        DO
                Take force readings
                Find force errors
                Find position increment
                Correct trajectory along robot's normal axis
                If desired force reached then
                        Correct trajectory along robot's tangential axis
        UNTIL flat surface tracked
4 End
```

The speed of tracking a flat surface depends on how quickly the manipulator reaches the desired force to be applied on the constraint, because of the way we chose to implement the control law. The sooner such a force is reached, the sooner the manipulator moves to the next tangential location. Therefore, it is that displacement along the unconstrained axis that tracks the surface.

The second case is similar to the contour tracking of a flat surface, the first difference is that in this experiment the user enters control parameters, such as desired force to apply on constraint, force tolerance and speed of the manipulator. The second difference is that the trajectory of the manipulator is altered along the x, y and z axis of the robot tool's frame. The pseudo code that describes this process is:

```
1 Enter control parameters
        Contact force
        Contact force tolerance
        Speed of the manipulator
2 Make contact
3 Achieve normal force
4 Control loop:
```

```
DO
        Take force readings
        Find force errors
        Find position increment
        Correct trajectory along robot's normal axis
        If desired force reached then
                Correct trajectory along robot's tangential axis
                Rotate tool
        End
    UNTIL curved surface tracked
5 End.
```

The contour tracking of a curved surface requires the definition of task coordinates [Goldsmith, 1995], such that the force applied by the robot on the constraint is always normal to the constraint. See Figure 3.9.

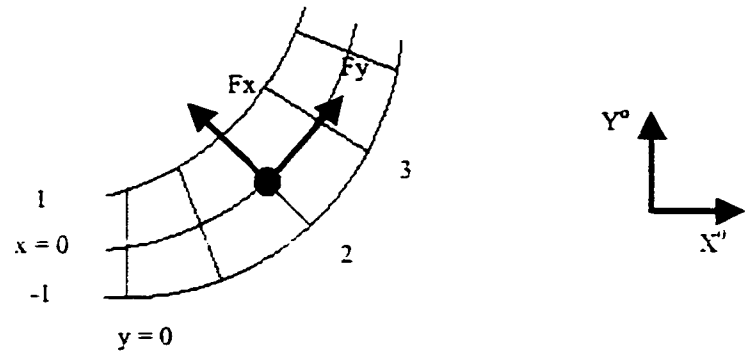


Figure 3.9 Task coordinates for a curved surface.

The tracking force applied by the manipulator on the constraint consists of the compression force normal to the surface and the friction force tangent to the surface. In Section 3.2, equation (20), the effects of the acceleration of the manipulator and the static

friction on the force sensor readings are stated. There are no dynamic friction forces because the applying of the normal force on the constraint is done when the manipulator is not tracking the surface. Therefore, the system deals only with compression and static friction forces.

At the first contact point, the force/torque sensor will read the forces applied on the constraint. The resultant force will have an x component and a y component in the tool coordinate frame. In the ideal case where there are no friction forces, we know that when the tangential force is equal to zero, the normal force is equal to the resultant force. To minimize the tangential force, it is necessary to align the resultant force with the normal component and α is the angle by the which the x component of the resultant force in the robot tool's frame must be rotated. See Figure 3.10. The angle α is approximately equal to:

$$\alpha = \tan^{-1} (fy / fx) \qquad (21)$$

It can be seen that α is the angle by which the end effector's tool has to be rotated about the z axis such that the compression force is always normal to the constraint. Once the manipulator reaches the desired force, it takes the force sensor readings, finds α, rotates the end effector's tool and moves the manipulator along the unconstrained axis using the Alter mode. The unconstraint axis will always be equal to the force sensor y axis. By repeating this procedure, the manipulator tracks a curved surface.

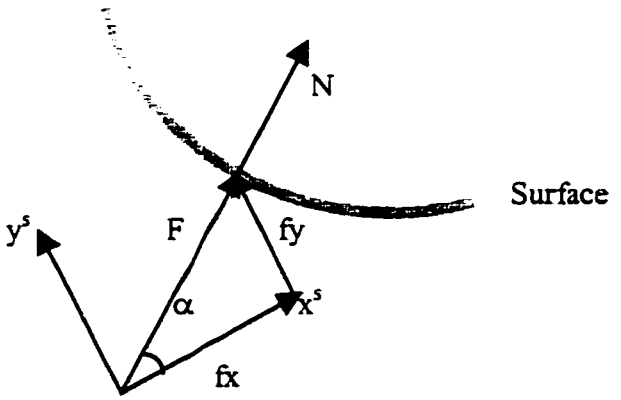


Figure 3.10 Forces at a contact point of a curved surface.

Once the manipulator reaches the desired force, it is commanded to move along the tool's tangential axis and then it is commanded to rotate about the same axis. One limitation of the system expected, is the loose of contact between the tool and the constraint when the tangential increments are big. The manipulator will require more iterations to reach the desired force, slowing down the system.

It is hypothesized that the manipulator will yield a good performance for the contour tracking of both the flat and curved surfaces using the control law written in the computer algorithms for the indirect hybrid position/force control method. The number of iterations that the force control loop will require to reach the desired force is:

a. require ZERO iterations (if the force is already within desired force range), or

b. require ONE iteration (if the force is outside the desired force range tolerance and the robot has sufficient force resolution), or

c. NEVER reach the desired force (if the force is outside the desired force range tolerance and the robot's force resolution is too small).

3.5 HYPOTHESES SUMMARY

To summarize, the key hypotheses stated in this chapter are:

1. Indirect hybrid position/force control is a feasible method to control the force through the position when using industrial manipulators. (page 2)

2. The compliance in the system is provided not only by the constraint, but also by the joints of the manipulator (page 21).

3. The acceleration along the y axis and the static friction will produce force readings along the y axis of the force sensor even when the manipulator is commanded to move only along the x axis (page 26).

4. The manipulator will yield a good performance for the contour tracking of both the flat and the curved surfaces using the control law designed in the computer algorithms for the indirect hybrid position/force control method (page 31).

CHAPTER 4

EXPERIMENTS

In order to study the feasibility of the indirect hybrid position/force control method, several experiments were conducted. Such experiments consist of the determination of the stiffness of the three materials to be used and the application of a constant force by the manipulator while tracking the contour of a flat and a curved surface.

The purpose of the experiments is to determine how well the system performs using the indirect hybrid position/force control law and to identify all the limitations affecting the performance of the system. Prior to executing the force control experiments, stiffness experiments are conducted in order to determine the stiffness coefficient used to optimize the force control algorithms.

4.1 EXPERIMENTAL SETUP

Figure 4.1 is a representation of the Adept robot [Adept, 1988], its controller, the terminal, the force/torque sensor, and the constraining object used during the experiments. The force/torque sensor and the joint encoders of the manipulator communicate with the controller, which displays the requested information in the terminal.

The manipulator has three revolute joints and a prismatic joint. Joints 3 and 4 will not be moved in order to restrict the degrees of freedom of the robot. Links 3 and 4 will be considered as an extension of link 2. Figure 4.2 depicts the two degree of freedom SCARA configuration obtained. SCARA stands for Selected Compliant Articulated Robot for Assembly [Adept, 1988].

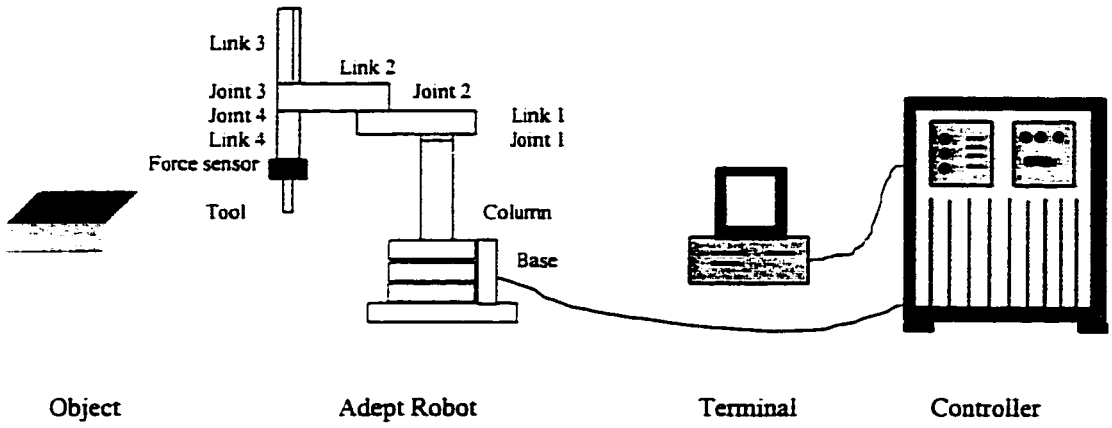


Fig. 4.1 System of study.

The Adept robot is a direct drive robot with a large work volume and 20-pound payload. It is designed to interface with the controller, which provides an environment that runs the V$^+$ language and controls the robot and external devices such as the force sensor [Adept, 1988].

The force sensor detects forces and torques exerted by the robot's tool. Strain gauges mounted on beams within the force sensor produce voltages proportional to the

bending that occurs when the robot applies a force. On-board electronics process the voltage differences and transmit the data to the controller [Adept, 1989].

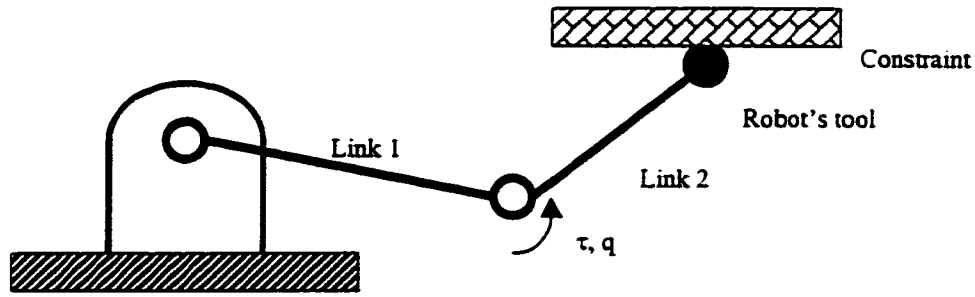


Figure 4.2 Two Degree of Freedom SCARA configuration.

The tool consists of a rod that has a height of 222.25 mm and a diameter of 30 mm. One end of this rod is attached to the force/torque sensor and the other end has a short cylinder of only 15 mm high coupled to it. The reason for having this second cylinder is to create a protuberance that will be the contact point between the tool and the constraint, and this protuberance is used to apply forces on the different constraints fixed to a table. See Figure 4.3.

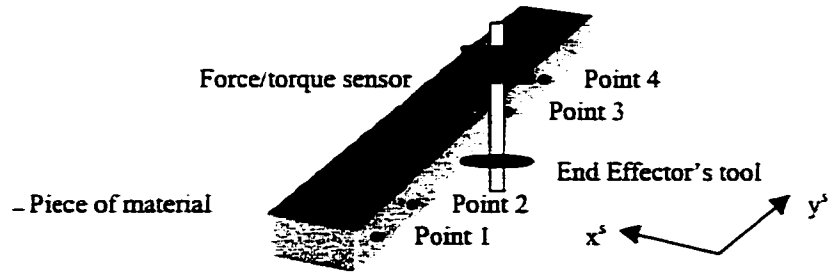


Figure 4.3 Stiffness experiment

4.2 STIFFNESS EXPERIMENT

The purpose of doing the stiffness experiment is to determine the stiffness coefficient for each one of the three materials chosen for the one degree of freedom force control and contour tracking experiments.

The computer program, written using Adept V$^+$ version 10.4 [Adept, 1988], is the key element in the system. V$^+$ is a programming language designed for use with Adept Technology Inc. industrial robots. This language is similar to computer languages such as Basic. The only difference is that it accepts motion commands. As this language, like any other industrial robot language, does not accept force instructions, a way has to be developed to control force using motion commands.

The robot executes a constrained motion task. We must determine the stiffness coefficient of the object to which the force is to be applied, so that we can correlate the performance of the system with the stiffness of the constraint. Therefore, three different materials are chosen to run the experiments: wood, wax and foam.

To find the stiffness of the materials, a robot program was written which determines the force and torque variations when the robot's arm experienced continuous displacements of .1 mm along the x axis of the world coordinate frame, which is normal to the surface of the object. The following is the program used for all the materials.

```
". PROGRAM stiffness()
 Distance_Y = 177
 Distance_Z = 724
 Rotation_X = 0
 Rotation_Y = 180
 Rotation_Z = 90
 MOVE non-contact_point
 DEVICE (2,0,,FS.SET.ZERO)
 TYPE /X4, "Xloop", /X3, "Xencoders", /X8, "FX", /X8, "FY", /X8, "TX", /X8, "TY"
 FOR X=593.6  TO 594.6 STEP 0.1
    MOVE TRANS(X, Distance_Y, Distance_Z, Rotation_X, Rotation_Y, Rotation_Z)
    DELAY 1
    DEVICE (2,0,,FS.GET.FORCE),F[]
    TYPE /F10.4, X, DX(HERE), F[0], F[1], F[3], F[4]
 END"
```

The above program moves initially the manipulator to a non-contact point. Then, a loop of 11 iterations follows. On each one of these iterations, the manipulator has to move along the X axis with increments of 0.1 mm, get the force readings from the force sensor, and display the position of the manipulator generated by the computer program, the position read by the joint encoders, and the forces and torques read by the force/torque sensor. The tool makes contact with the object on the first position increment commanded within the loop. See Figure 4.3.

The reason for starting at a non-contact point is to clear the force data latches to compensate for the end effector weight and temperature variations, set the force sensor readings to zero and ensure that there are no offsets. The force/torque sensor is a very sensitive device that is affected greatly by temperature variations [Adept, 1989].

This experiment was run nine times on each one of the four different points chosen for each material, in order to compensate for the dynamic effects of the force/torque sensor, mathematical noise, force/torque sensor noise, temperature variations, etc that modify the experiments. These points are depicted in Figure 4.3, page 35. In Table 4.1 the results for the Point 1, Run 1 of the piece of wood are shown. We summed these results together for each one of the nine runs and four points of the material and obtained an average in order to have a more homogeneous stiffness coefficient factor throughout the constraint and improve the force/torque readings to noise ratio.

TABLE 4.1 Forces and torques applied to the piece of wood
Point No 1, Run No 1

Loop	Xloop (mm)	Xencoder (mm)	Xerror (mm)	FX (N)	FY (N)	TX (Nmm)	TY (Nmm)	TY/h (N)
1	593.60	593.01	0.59	0.02	0.01	-1.06	2.03	0.01
2	593.70	593.61	0.09	4.17	0.53	-124.50	941.82	4.24
3	593.80	593.69	0.11	8.38	-0.06	7.51	1847.41	8.31
4	593.90	593.81	0.09	14.85	1.69	-397.01	3233.91	14.55
5	594.00	593.92	0.08	20.12	1.74	-412.47	4363.23	19.63
6	594.10	594.00	0.10	24.62	2.03	-479.49	5318.79	23.93
7	594.20	594.12	0.08	30.64	3.18	-748.90	6618.43	29.78
8	594.30	594.22	0.08	36.04	3.88	-910.90	7762.74	34.93
9	594.40	594.30	0.10	39.76	3.10	-738.05	8562.21	38.53
10	594.50	594.41	0.08	45.59	4.31	-1018.7	9835.51	44.25
11	594.60	594.52	0.08	50.43	3.84	-921.28	10868.7	48.90

For dynamic effects of the force/torque sensor we define high-frequency fluctuation and the sensitivity of such device. As a result of mechanical vibration or

electrical noise, the strain gauge readings of the force/torque sensor will normally have some high-frequency fluctuation. The manufacturer has determined that even though this noise level is well below the force level in which a user may be interested, it may be undesirable when the user wants to detect extremely low force level changes [Adept, 1989]. The force sensor allows the user to define a force filter parameter for noise level. The higher the parameter is, the more accurate the force reading will be, but also, the more time the force/torque sensor will require to execute the force and torque readings.

From Table 4.1 we can see that the position reached by the robot and read by the joint encoders, Xencoder (mm), is not equal to the position defined in the computer program, Xloop (mm). The normal forces FX (N) are positive and increasing. The tangential forces FY (N) are insignificant compared to the values of the forces along the X axis. The torques about the x axis are negative and decreasing. The torques about the y axis are very significant and increasing. The sign in the force/torque readings indicates the sense of the resulting force/torque.

The force sensor is separated from the contact point of the tool by a vertical distance h of 222.25 mm, and this is the amount by which the torques are divided by to obtain the forces. The previous results are comparable to those read by the force/torque sensor as can be seen from the columns "Fx(N)" and "Ty/h (N)" of Table 4.1. The difference between the forces read by the sensor and those calculated from the torque measurements are caused by the sensor calibration and noise.

Figures 4.4.1, 4.4.2 and 4.4.3 show respectively the mean of the forces applied on the three different materials while the manipulator executed displacements close to 0.1 mm. It can be seen the normal force FX, the tangential force FY and the force resulting from dividing the torques about the Y axis by the height of the tool Ty/h. In these figures we can observe that the forces are proportional to Xencoder. FX and TY/h have a behavior very similar and in some points they overlap.

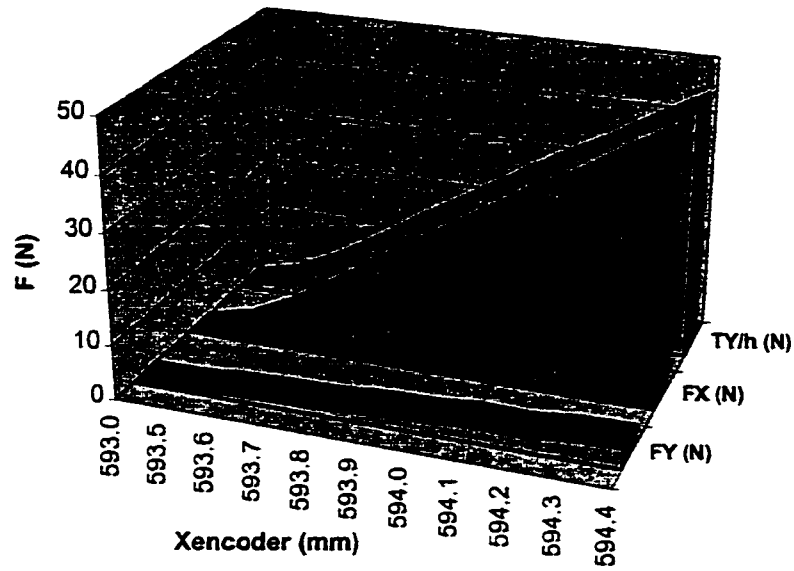


Figure 4.4.1 Mean normal forces (Piece of wood)

There are two main reasons why the forces applied by the robot along the tangential axis are not zero. Firstly, the constraint is not perfectly aligned with the force/torque sensor frame. Therefore, the force F_n, normal to the constraint, will have a component along the y axis of the force/torque sensor frame. Secondly, there is a friction force F_f opposed to the displacement of the manipulator (see Figure 4.5). Friction forces

will give an error in the direction of the normal force, which will give an error in the position of the constraint. The forces applied by the robot along the x axis vary when the manipulator is moved along that same axis, being affected by the x component of the static friction force.

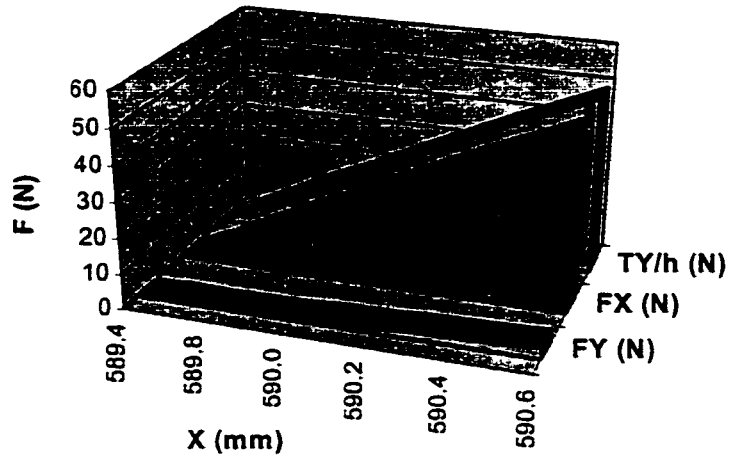


Figure 4.4.2 Mean normal forces (Piece of wax).

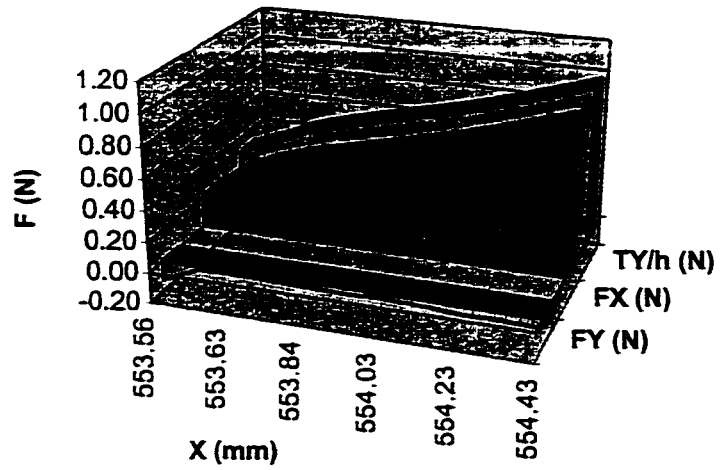


Figure 4.4.3 Mean normal forces (Piece of foam).

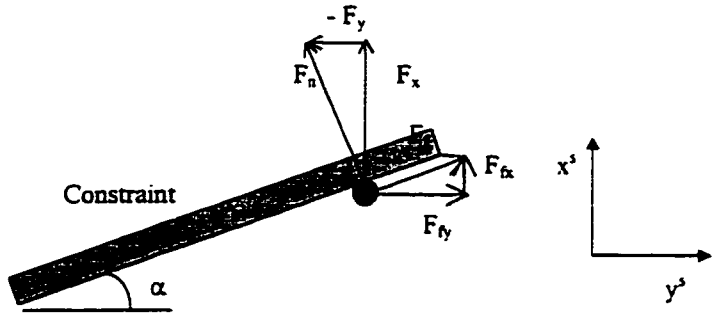


Figure 4.5 Upper view of the forces at the contact point.

One important finding is the difference between the real position of the manipulator read by the joint encoders, and the commanded position generated by the loop of the computer program. This difference will be called "Xerror" and is depicted in equation 22 and shown in Table 4.1.

$$\text{Xerror} = \text{Xloop} - \text{Xencoder} \qquad (22)$$

In theory, the Xerror should be zero, but in reality, it can be observed that it is never equal to zero. In Figures 4.4.1, 4.4.2 and 4.4.3 a consequence of this Xerror can be observed when the distance between the first and the second points is not proportional to the distance between the following points in the three materials which have linear stiffness. The manipulator tries to compensate for the difference between the Xloop and Xencoder at the first contact point generating a great position increment such that the new Xerror at the second contact point is close to zero. Moreover, for the very first displacement of the manipulator, the greatest error is recorded, as can be appreciated in Figures 4.6.1, 4.6.2, and 4.6.3

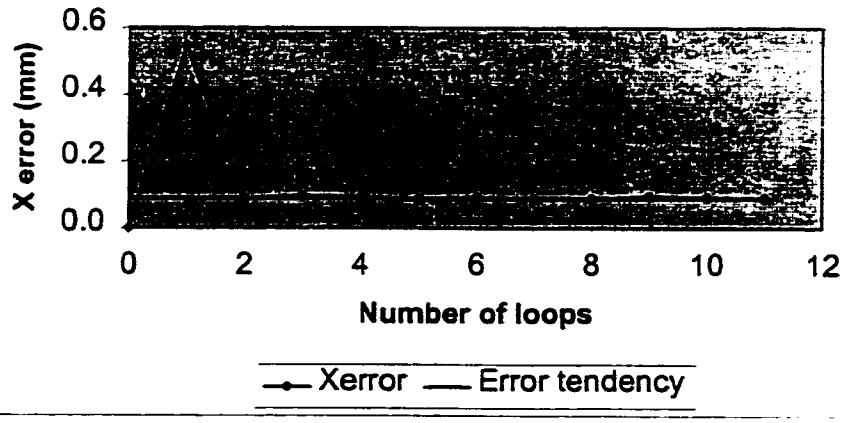


Figure 4.6.1 X position error for wood.

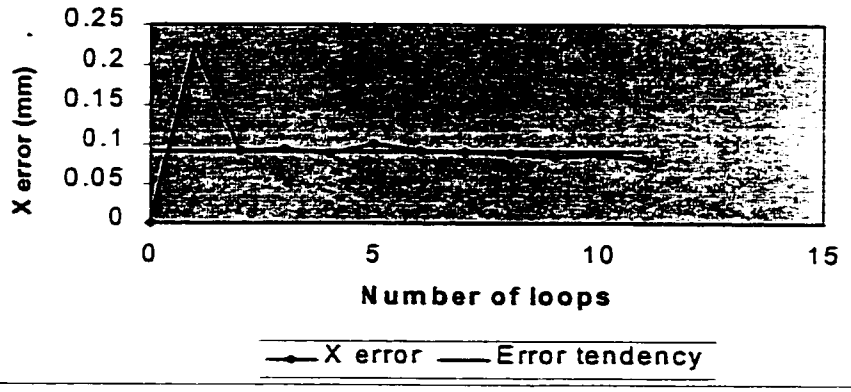


Figure 4.6.2 X position error for wax.

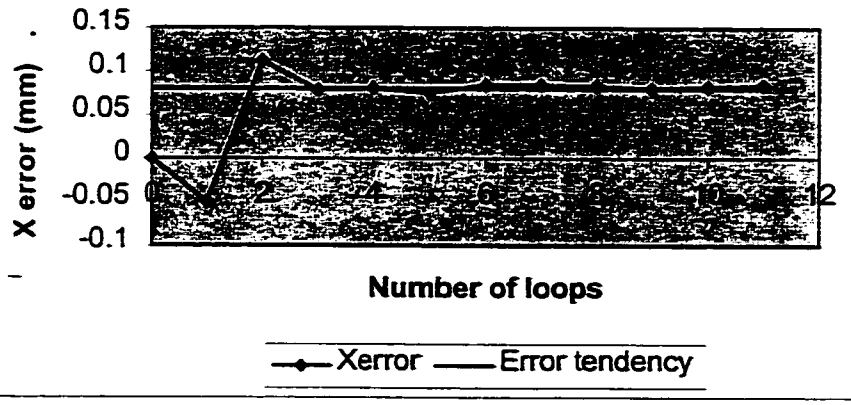


Figure 4.6.3 X position error for foam.

In the preceding figures we can see that the position (0,0) represents the non-contact point. In the curve "Xerror", the loop No 1 represents the first contact point, and in fact, the force on the constraint is applied here for the first time. Therefore, this first segment of the curve symbolizes the displacement of the manipulator from the non-contact point to the first contact point. In loop No 1 the controller detects the error originated by the difference between the Xloop and the Xencoder. So, the manipulator compensates for that Xerror and this is depicted in the second segment of the curve. For the consecutive segments, the manipulator tries to reduce the Xerror constantly. The curve "Error tendency" shows approximately the value at which the Xerror tends to converge.

This first displacement increment will not be taken into account when calculating the average position increment of the manipulator because it is a deviation shown only during the first movement, and it is related to the control law applied by the industrial controller. It is a reflection of the acceleration reached by the robot when it starts a motion instruction. When a single motion instruction is processed, the robot begins moving toward the location by accelerating smoothly to the commanded speed. Later, when the manipulator is close to the location, the robot decelerates smoothly to a stop at the specified location. This process is called "Continuos Path Control". It is not clear what originates the greatest Xerror for the first displacement of the manipulator, it might be the acceleration of the manipulator or even the control law which we do not have access to.

Figures 4.7, 4.8 and 4.9 show the increment in the force and position, as well as the mean of such values. On the piece of wood, for an average position increment of 0.0998 mm along the x axis, there was on average a force increment applied by the robot along the x axis of 5.0264 N. On the piece of wax, for an average position increment of 0.1010 mm along the x axis, there was an average a force increment applied by the robot along the x axis of 5.0290 N. And on the piece of foam, for an average position increment of 0.1031 mm along the x axis, there was on average a force increment applied by the robot along the x axis of 0.0658 N.

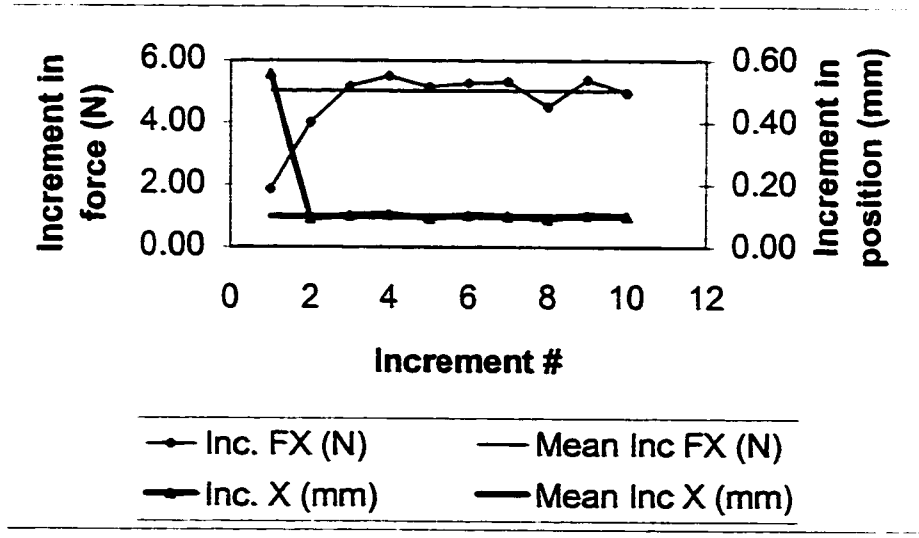


Figure 4.7 Normal force and position increments for wood.

It can be concluded that the stiffness of the piece of wood used in the experiments, is 50.364729N/mm. In other words, for each positional increment of 1mm, there is an increment in the force of 50.364729 N. The stiffness of the piece of wax used is 49.792079 N/mm, and the stiffness of the piece of foam used is 0.638215 N/mm.

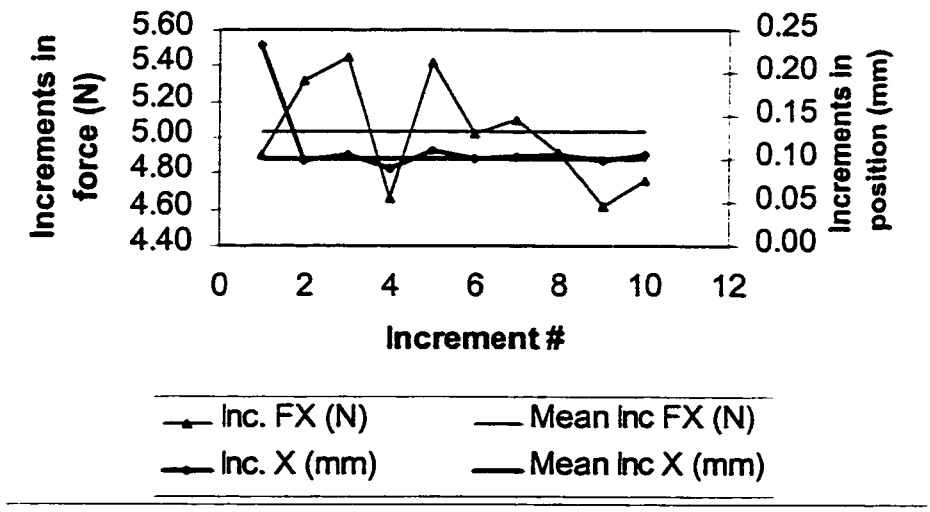


Figure 4.8 Normal force and position increments for wax.

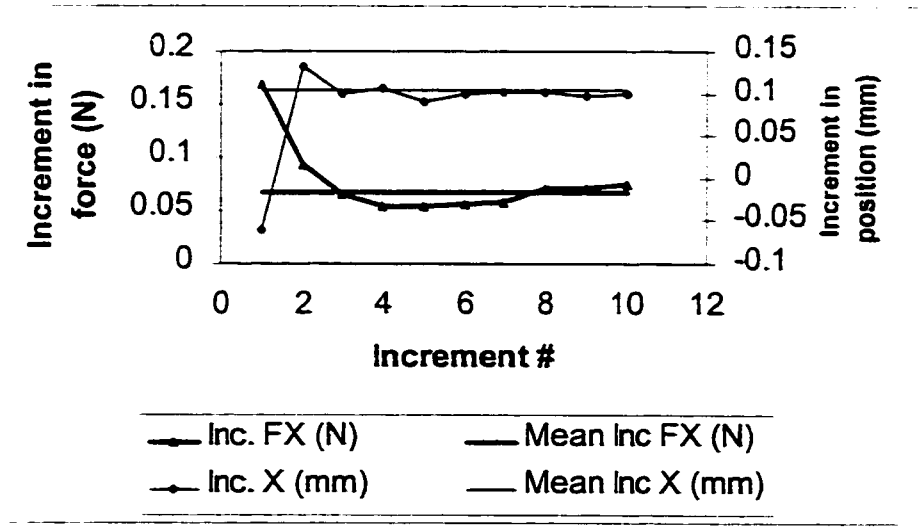


Figure 4.9 Normal force and position increments for foam.

Although the piece of wax has a stiffness coefficient almost identical to the stiffness coefficient of the piece of wood, its position increments are better for the first displacement of the manipulator than the one for the piece of wood. The reason for this is

the homogeneity of the wax grain. The wax has a more compact and homogeneous surface than the wood. A smooth surface will be easier to track than a coarse surface.

The similarity between the stiffness coefficient of the piece of wood and the piece of wax suggests that the compliance is in the robot, and not in the object. Therefore, we hypothesized that the stiffness coefficients found did not belong to the materials studied, but to the robotic manipulator.

A fourth material was selected to verify this hypothesis: steel. Since steel is the hardest material amongst wood, wax and steel, it was reasonably assumed that its stiffness coefficient would be significantly greater than the coefficients of the wood and wax. In Figure 4.10 we can see the normal forces applied to the constraint versus the positions read by the joint encoders. Once more, the behavior of the system using steel as a constraint was similarly to the behavior obtained with the other materials studied. It is observed that for an average position increment of 0.1005 mm along the x axis, there was on average a force increment applied by the robot along the x axis of 5.1208 N. The stiffness of the piece of steel used in the experiments, is 50.9532338 N/mm.

With the experiments carried out with the steel as a constraint, we conclude that the compliance is in the robotic manipulator. Therefore, the bending of the robot is generating the force resolution of the system which depends on the position resolution of the manipulator and which depends on the robot position. This is a limitation of the system.

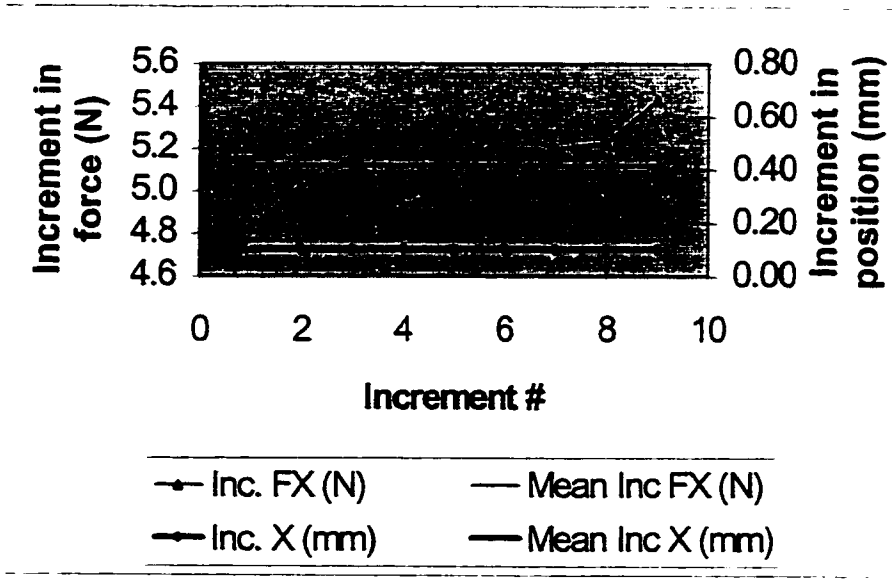


Figure 4.10 Normal force and position increments for steel.

To summarize, in this section was hypothesized that when using stiff materials the stiffness coefficients found did not belong to the materials, but to the robotic manipulator. Such hypothesis was verified and it is concluded that the compliance is in the robotic manipulator. Therefore, the stiffness coefficient of the piece of foam was 0.638215 N/mm and the stiffness coefficient of the robotic manipulator was approximately 50 N/mm.

4.3 FORCE CONTROL EXPERIMENTS

The purpose of doing the force control experiments is to determine how feasible the indirect hybrid position/force control method is. The force study will be divided into

two parts. The first one is the one degree of freedom force control, and the second one is force control for contour tracking. In the one degree of freedom force control section, computer algorithms for the three different objects were written. Also, the problem of low position resolution is tackled and improved versions of the computer algorithms are presented.

In order to guarantee a good performance of the force/torque sensor, a computer program was written to configure the sensor. This algorithm sets the parameters for the conversion factor for force in Newtons, for length in millimeters and the force filter parameter for noise level. It enables the protect and guard modes and sets the force and torque values that trigger the protection and guard modes of the sensor. This computer program can be read in Appendix A-1. The guarded mode sets a binary signal when a force trip occurs and halts the robot at the point of the trip. There is no recovery time required, hence the manipulator is ready for the next motion. The protect mode is activated when the strain gauge threshold is exceeded and acts like a panic stop disabling the power of the robot.

4.3.1 ONE DEGREE OF FREEDOM FORCE CONTROL

This part of the study is called "One Degree of Freedom Force Control" because the robot is moved only in one direction along the normal axis, in order to apply a desired force of 10 N. The same computer programs are used for the three materials, the only

difference is the control gain which varies depending on the material used. The basic approach used to control the robot is described in Chapter 3, section 3.3.

4.3.1.1 RESULTS

In Chapter 3, Section 3.3, the main pseudo code used to write the computer algorithms utilized in the one degree of freedom force control experiments was explained. The control program consists of two programs running in parallel, force.control and path.modification, and a subroutine, main.loop. See Appendixes A-2, A-3 and A-4 respectively. Following there is a more thorough description of the computer programs. Even though these programs are used for both the one degree of freedom force control and contour tracking of a flat surface experiments, the results are analyzed individually.

```
 1 PROGRAM "force.control()"
 2 RESET SOFTWARE SIGNALS
 3 SET SPEED OF THE MANIPULATOR
 4  MOVE TO NON-CONTACT LOCATION
 5 WAIT FOR MOVE INSTRUCTION COMPLETION
 6 ZERO THE FORCE SENSOR
 7 EXECUTE "path.modification()"; "force.control and path.modification"
                                    running in parallel
 8 ENABLE ALTER MODE
 9 DO
10      MOVE TO 1st CONTACT POINT
11      WAIT FOR MOVE INSTRUCTION COMPLETION
12      SEND SOFTWARE SIGNAL; to indicate to "path.modification()" that
        robot has  moved
13      WAIT FOR SIGNAL; "path.modification()" sends signal when force
                              readings and contour tracking are completed
```

```
14      RESET SOFTWARE SIGNAL
15 UNTIL CONDITION IS SATISFIED
16 DISABLE ALTER MODE
17 END.
```

The program force.control sets the speed of the manipulator to 2% of its normal speed, moves the manipulator to a non-contact point close to the first contact point of the main loop, sets to zero the force sensor, orders the parallel execution of the program path.modification, enables the alter mode, and initiates the movement of the robot's arm within a DO-UNTIL loop sending software signals that let the path.modification program know when is the right time to read the forces. The alter mode is a V$^+$ facility that allows the use of sensory feedback to alter the robot trajectory. The reason for having two programs running in parallel is that it is recommended by the manufacturer [Adept, 1989].

```
 1 PROGRAM "path.modification()"
 2 SET TIMER (1) = 0
 3 SET TIMER (2) = 0
 4 DO
 5      WAIT FOR SIGNAL; from "force.control" indicating that robot has
                                  moved
 6      RESET SOFTWARE SIGNAL
 7       x = x+1
 8      counter = counter+1
 9      GET FORCE READINGS
10      errorx = desired force – normal force read by force sensor
11      correctx = errorx *  stiffness coefficient; distance to reach desired force
12      IF ABS(normal force) > ABS(tangential force) THEN
13           CORRECT TRAJECTORY ALONG ROBOT'S NORMAL AXIS
14      END IF
15      IF (-force tolerance <= errorx <= force.tolerance) THEN
16           CALL SUBROUTINE "main.loop()"; to track curved surface
17      END IF
```

```
18      INSERT 0.1 SECOND DELAY; to make sure signals are read
19    SEND SOFTWARE SIGNAL; to "force.control" sensor read forces and
                                    alter mode executed
20 UNTIL CONDITION IS SATISFIED; from "main.loop"
21 TYPE "TOTAL TIME = ", TIMER(2), " SECONDS"
22 END.
```

The program path.modification defines all the variables and initial conditions, reads forces and torques, calculates the force error, determines the distance by which the trajectory of the manipulator must be altered, moves the manipulator along the normal axis executing the alter mode – based on the principle that the force along the normal axis is greater than the tangential force -, sets the force tolerance to +/- 0.7 N, calls the subroutine main.loop when the difference between the desired force and the force read by the force sensor lies within the force tolerance, and sets the software signal that lets the force.control program know that the force sensor read the forces and the alter mode was executed.

```
 1 SUBROUTINE "main.loop()"
 2 CORRECT TRAJECTORY ALONG ROBOT'S Y AXIS
 3 cycle = cycle+1
 4 IF TOTAL LENGTH TRACKED THEN
 5    SATISFY CONDITIONS "A" FOR DO-UNTIL LOOPS; used in programs
                  force.control() and path.modification()
 6 END IF
 7 PRINT FACTORS AFFECTING PERFORMANCE
 8 RESET TIMER(1)
 9 RETURN
10 END
```

The subroutine main.loop executes the contour tracking of a flat surface using the alter mode, determines the length to track, sets the variable "A" true which defines the UNTIL condition in the DO-UNTIL loop of the program path.modification, and provides a feedback of the efficiency and the time per loop factors affecting the performance of the system.

4.3.1.1.1. WOOD

Wood is the first material chosen as a constraint to run the one degree of freedom force control experiments. The tool of the robotic manipulator, while being in contact with a constraint, will be moved along the x axis of the world coordinate frame until the desired force of 10 N is reached.

The programs described in section 4.3.1.1. were executed using the piece of wood as the constraint. In order to understand the results, some concepts have to be defined. "Loop" is the number of iterations required to obtain the desired force and it describes the displacement of the manipulator along the normal axis. "Cycle" defines the position increment from one point along the tangential axis, to the next one along the same axis, 1 mm apart. "X" and "Y" represent the variation in the position along the x and y axis respectively. Likewise, "FX" and "FY" are the forces measured along the x and y axis.

It could be seen that the desired force was not reached in one step as it was stated theoretically in Chapter 3. In some cases, it took only one or two iterations to reach it, but

in the worst case, the manipulator required 34 iterations to reach the force of 10 N. In Chapter 3, section 3.1.2, this problem was identified with the force resolution of the system.

In Section 4.2, Figure 4.4.1, we can see that the increment in the force along the x axis is proportional to the increment in the position along the same axis. In Figure 4.11, it can be observed that in every cycle, although the manipulator was not commanded to move along the y axis, there were small displacements registered along the y axis. This is explained by equation (20) of Chapter 3. Although the force applied was along the x axis, there was a small acceleration of the manipulator along the y axis which was recorded in the form of small displacements along the y axis. It is the static friction what makes the manipulator stay in that new position along the y axis after the acceleration along that same axis becomes zero. This observation verifies the hypothesis stated in Chapter 3. Another reason for this displacement along the y axis is the alignment of the material.

In Figure 4.11, we can see that although there was an increment in the force applied along the x axis, there was not a similar behavior in the change of the position in x and y. We observe that for this particular cycle the manipulator required 6 loops to reach the desired force. For the three first loops, there is an increment in the position. Then, for the next 2 loops we observe that the position does not change, and finally, for the last loop, there is a reduction in the position. Examples like these can be observed throughout the results. The factors causing this are the noise of the force/torque sensor, as expected, and the position resolution of the manipulator, which will be explained in more

detail in section 4.3.1.2 . They make the system slower, requiring more computations to obtain the force.

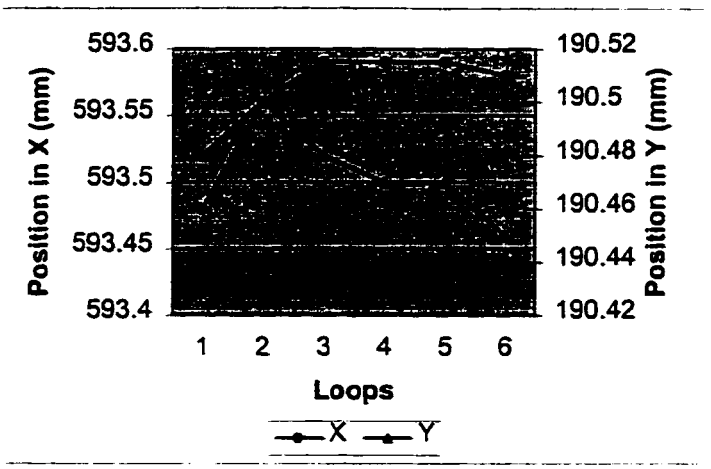


Figure 4.11 Deformation of the piece of wood.

Another important observation made from the previous figure is the deformation of the constraint. In this cycle, the total deformation along the normal axis is equal to 0.115 mm, and along the tangential axis there was a minimum compression of 0.05 mm. The wood is a stiff material, therefore, it does not undergo much deformation.

The parameters describing the system are explained next. The efficiency factor is defined as the number of loops per cycle required to reach the desired force. The sampling rate is the time required to obtain such force, and the loop rate is the time consumed at each loop. From this experiment one concludes that the loop rate is not constant, it varies per cycle. We can also say that every second the system made an

average of 2.0064 loops .The more loops per cycle required to reach the force, the less the time per loop consumed.

In order to study the efficiency of the system, we will consider only those cycles that required more than 2 loops per cycle to reach the force of 10 Newtons. Any cycle with one loop per cycle has to be considered as being within the force tolerance established in the computer algorithms. In other words, when the manipulator is moved 1 mm along the y axis (1 cycle), the force in the next cycle is very close to the desired force reached in the previous cycle and within the force tolerance, needing only one loop in the computer program to register the force readings. On the other hand, when the efficiency factor is 2 loops/mm it means that the manipulator required only 1 step to reach the desired force. The smaller the value of the efficiency factor, the better the system performs. In this experiment, there was a total of 12 cycles out of 50 that required more than 2 loops/cycle to reach the desired force, which means the 24% of the cycles. See Figure 4.12.

It was also observed that when the noise filter parameter was changed, the system performance changed as well. The greater the filter parameter is, the longer it will take the force system to sense contact, and this results in higher force and torque overshoots immediately after contact. Therefore, the settling time is greater and the system becomes slower. The best balance between noise reduction and fast response time was found with a filter parameter factor of 3, with 1 and 10 being the respective lower and upper limits.

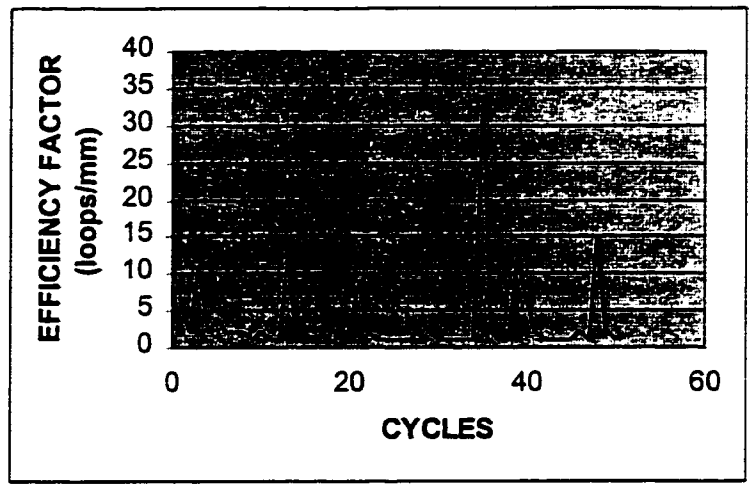


Figure 4.12 Efficiency of the system for wood with
a force tolerance = +/- 0.7 N

It is practically impossible to get at the same contact point two identical force readings. This is due to the dynamics of the force/torque sensor and variations of temperature. Therefore, a tolerance has to be set to improve the performance of the system. The results shown in Figure 4.12 were obtained with a force tolerance of +/- .7N, which means that any force reading between 9.3 N and 10.7 N is considered to be equal to the desired force of 10 N. Four new force tolerances were chosen: +/- 0.85 N, +/-1 N, +/-1.5 N, and +/- 2 N.

In Figure 4.13 we can observe the performance of the system for the different force tolerance variations. The worst performance was reported for a force tolerance of +/- 0.7 N. Out of 47 cycles, 11 required more than 2 loops per cycle to achieve the desired force (23.4 %). For a range of +/- 0.85 N, the performance of the system

improved showing that only 6 cycles required more than 2 loops per cycle (12.76 %), although 2 of those cycles had more than 9 loops per cycle. When the force tolerance was set to +/-1 N, the performance decreased compared to the results obtained with the previous force tolerance. 17.02 % of the cycles had more than 2 loops per cycle, but the maximum value of loops per cycle registered was 6. The performance of the system improved for a force tolerance of +/- 1.5 N where 5 cycles out of 47 (10.63 %) shown more than 2 loops per cycle. Finally, the best performance was obtained with a force tolerance of +/- 2 N. Only 8.51% of the cycles required more than 2 loops per cycle to reach the desired force of 10 N.

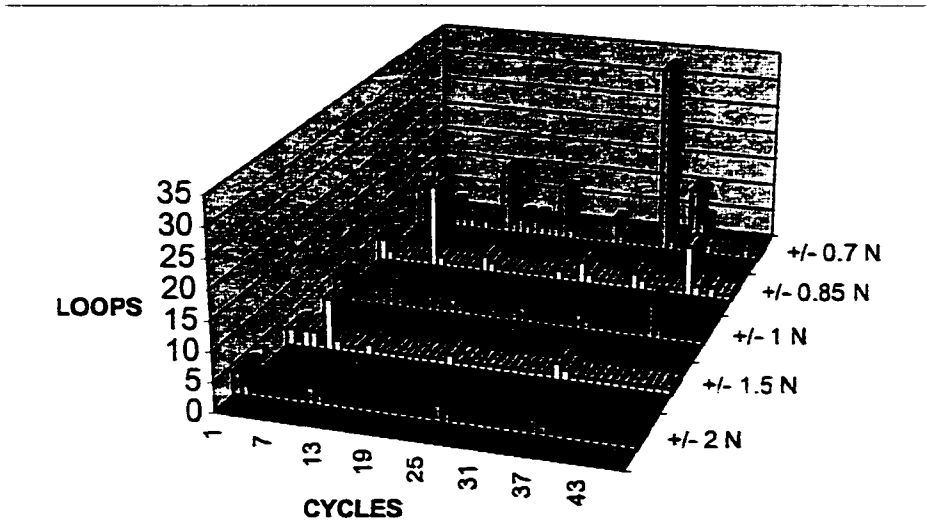


Figure 4.13. Force tolerance variations for wood.

As it can be seen from the results depicted in Figure 4.13, the bigger the force tolerance, the fewer loops per cycle the manipulator makes to reach the desired force.

Therefore, if the manipulator requires fewer loops per cycle to reach such force, the faster the system becomes. In other words, there is a trade-off between accuracy and speed.

From Figure 4.13, we can also note a periodicity on each one of the curves. For example, in the curve representing the force tolerance of +/- 0.7 N, there is a peak in average every 8 samples, which represents a harmonic distortion with a frequency equal to 8. In the force readings this reflects some noise interference originated in both the force/torque sensor and the joints of the manipulator.

4.3.1.1.2. WAX

Wax is the second material chosen as a constraint to run the one degree of freedom force control experiments. In the previous sections we discussed the results of the experiments using a piece of wood as the constraint. We can say that the same observations made for the piece of wood with respect to the behavior of the forces and displacements of the manipulator are applicable to this case.

In Figure 4.14, we can analyze the deformation of the piece of wax. In this cycle, the total deformation along the normal axis is equal to 0.11 mm, and the deformation along the tangential axis was 0.03 mm. The wax behaved very similar to the wood, not undergoing much deformation.

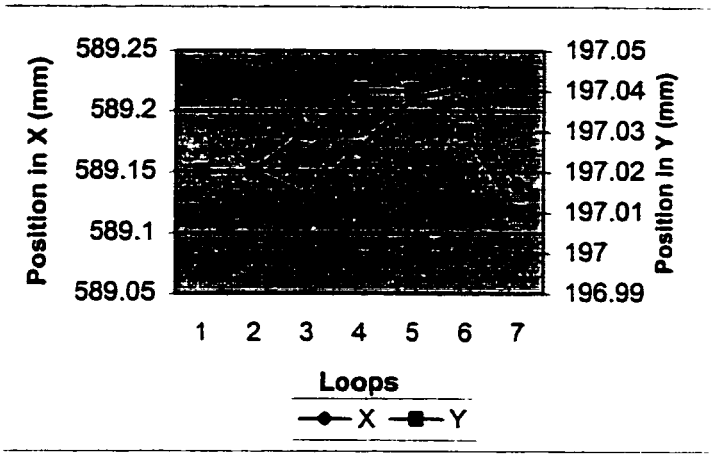


Figure 4.14 Deformation of the piece of wax.

In Figure 4.15, we can observe that for a force tolerance of +/- 0.7 N, the worst efficiency factor was 19 loops/mm. We can say that the system performed better with the piece of wax than with the piece of wood. The reason for stating this is that there was a total of 9 cycles out of 50 that required more than 2 loops per cycle to reach the desired force, representing the 18% of the cycles.

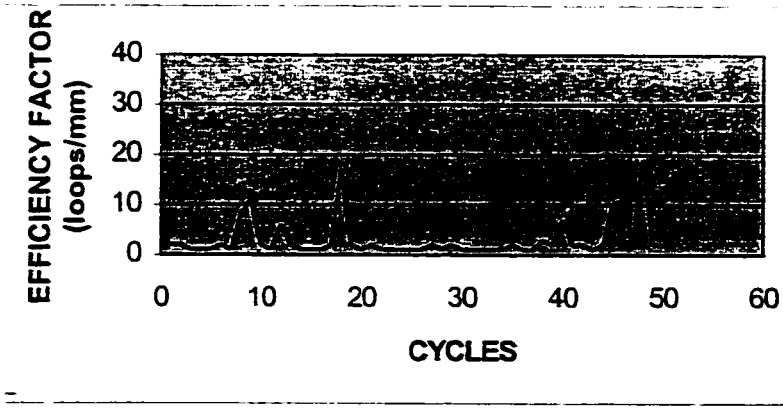


Figure 4.15 Efficiency of the system for wax with a force tolerance = +/- 0.7 N.

In average the manipulator covered 1.9528 loops per second. In conclusion, for a force tolerance of +/- 0.7N the system performed better than with the piece of wood. It was faster, it has a better efficiency factor and required less loops/s than the performance with the piece of wood.

Once again, four different force tolerances were chosen in order to study the performance of the system when the limit for the normal force was varied. These force tolerances are: +/- 0.85 N, +/- 1 N, +/- 1.5 N and +/- 2N.

In Figure 4.16, the number of loops per cycle for each one of the force tolerances selected are depicted. The worst performance of the system was registered for a force tolerance of +/- 0.7 N. A total of 9 out of 50 cycles (18 %) had more than 2 loops per cycle and 19 was the maximum value of loops per cycle. When the force tolerance was set to +/- 0.85 N the performance of the system improved, showing that only 7 cycles out of 51 (13.72 %) required more than 2 loops per cycle to reach the desired force and the maximum number of loops per cycles was 17. The performance of the system improved for a force tolerance of +/- 1 N where 6 cycles out of 51 (11.76 %) needed more than 2 loops per cycle and the maximum value of such loops was 9. For a force tolerance of +/- 1.5 N the performance improved when only 4 cycles out of 51 (7.84 %) required more than 2 loops per cycle and being 6 the maximum value of loops per cycle read. Finally, the best performance observed was for a force tolerance of +/- 2 N when just 3 cycles out of 51 required more than 2 loops per cycle and the maximum value of loops per cycle was 3.

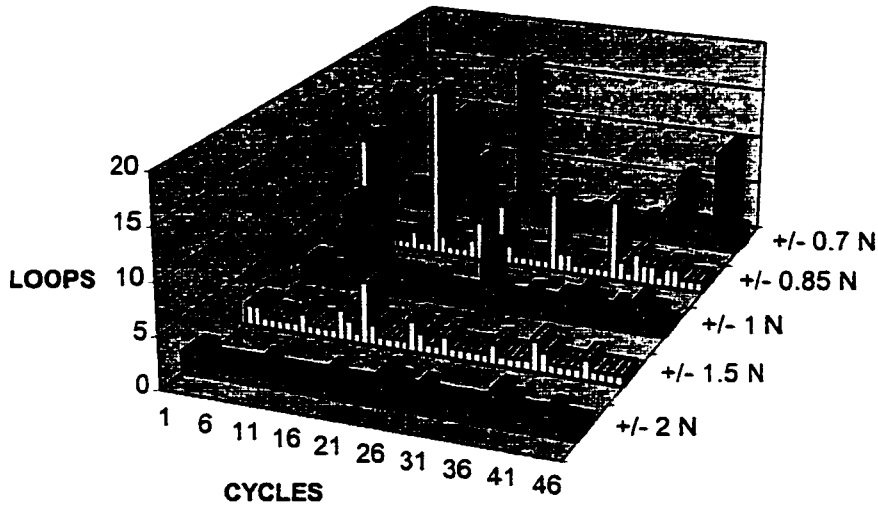


Figure 4.16 Force tolerance variations for wax.

One can conclude from the results explained in the previous paragraph, that the performance of the system is proportionally inverse to the force tolerance. In other words, the greater the value of the force tolerance, the less the number of loops per cycle required to reach the desired force and also, the less the maximum value of the loops per cycle read.

4.3.1.1.3. FOAM

The one degree of freedom force control experiments were run once again, using the piece of foam as the constraint. The same comments regarding the description of the experiments with the pieces of wood and wax apply to this case. Similarly, it was also

observed that the same comments related to the variation of the normal forces with respect to the variation of the position in x and y are applicable to this case.

In Figure 4.17, we can observe the deformation of the piece of foam. The worst cycle with a force tolerance of +/- 0.7 N was selected to explain the deformation of the material. In this particular cycle, the manipulator required 4 loops to reach the desired force. The deformation along the normal axis was equal to 13 mm, and along the tangential axis was equal to 0.08 mm. We can conclude that the more compliant the constraint, the more deformation the material shows.

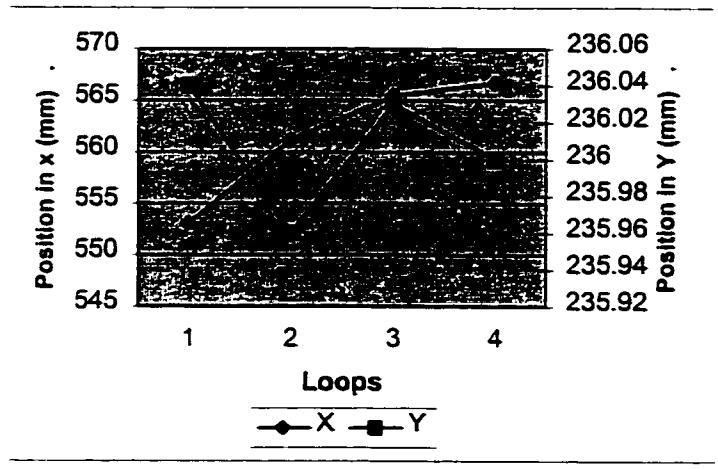


Figure 4.17 Deformation of the piece of foam.

In Figure 4.18 the results for the one degree of freedom experiment are shown. These results were obtained with a force tolerance of +/- 0.7 N and a desired force of 10 N. As it can be observed, the system had the best performance compared to the results obtained with the pieces of wood and wax. There were a total of 3 cycles out of 50 that

required more than 2 loops per cycle to reach the desired force, representing 6% of the cycles.

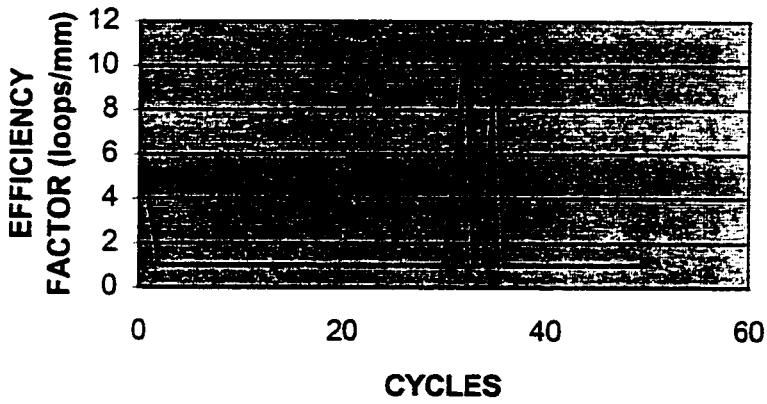


Figure 4.18 Efficiency of the system for foam
Force tolerance = +/- 0.7 N.

Experiments were run to determine the behavior of the system when the force tolerance was modified to +/- 0.85, +/- 1 N, +/- 1.5 N and +/- 2 N. As it can be seen in Figure 4.19, the system performed extremely well for all the force tolerances.

The worst performance was obtained for a force tolerance of +/- 0.7 N. For a range of +/- 0.85 N, the performance of the system improved showing that only 2 cycles required more than 2 loops per cycle. In other words, only 4.25 % of the cycles required more than 2 loops per cycle to reach the desired force of 5 N. When the force tolerance was set to +/-1 N, the performance decreased compared to the results obtained with the

previous force tolerance. 6.38 % of the cycles had more than 2 loops per cycle, but the maximum value of loops per cycle registered was 6. The best performance of the system was obtained for a force tolerance of +/- 1.5 N and +/- 2 N, where only 1 cycle out of 47 (2.13 %) shown more than 2 loops per cycle.

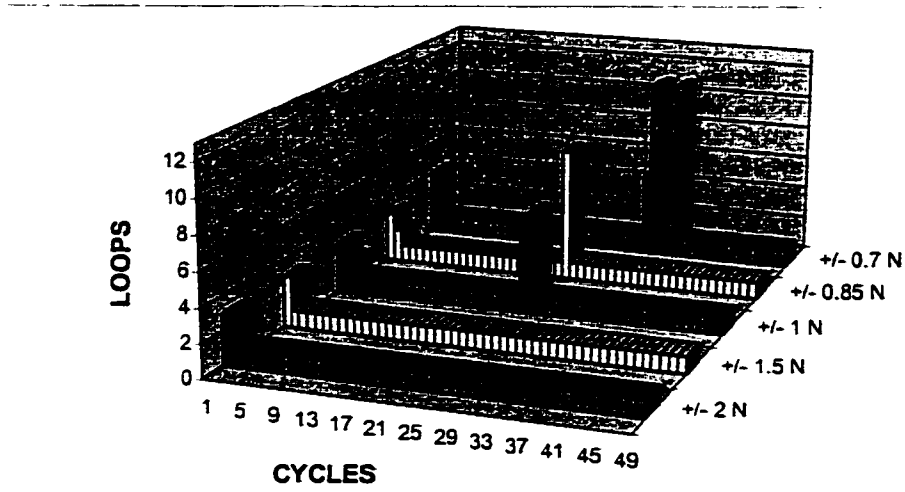


Figure 4.19 Force tolerance variations for foam

Taking into consideration only the best two performances of the system (force tolerance of +/- 1.5 N and +/- 2 N), the following indices were obtained:

- The average speed of the manipulator was 1.37 mm/s.
- Every second the manipulator required an average of 1.43 loops to reach the desired force of 10 N.

The system performed the best with the foam as a constraint. It was the fastest, and more efficient having the least amount of cycles with more than 2 loops per cycle, the least number of loops required in the 47 cycles, loops per millimeter and loops per second. Table 4.2 summarizes the results for the three constraints used in the experiments for a force tolerance of +/- 2 N, which yielded the best results in the three cases and for the same number of cycles.

TABLE 4.2 SYSTEM INDICES FOR THE THREE MATERIALS. (USING A FORCE TOLERANCE OF +/- 2 N AND CONSIDERING 47 CYCLES)

Material	Cycles with more than 2 loops/cycle	# Loops	Loops/mm	Loops/s
Wood	4	60	1.27	1.56
Wax	2	59	1.25	1.55
Foam	1	49	1.04	1.43

4.3.1.2. POSITION RESOLUTION

In this section the position resolution of the manipulator is analyzed. From the results obtained in the one degree of freedom force control experiments, it was observed that the robotic manipulator was not always able to reach the desired force within a reasonable number of loops per cycle. Since the force resolution of the system depends

on the position resolution of the manipulator, it is necessary to determine how this factor affects the performance of the system.

In chapter 3, section 3.1, the force resolution of the system was discussed via an example. The manufacturer provided the position resolution and the stiffness coefficient of the constraint was the one obtained with the wood stiffness experiment. As it was mentioned before, the minimum force increment that the robot can achieve theoretically, when applying a force on such constraint, is 0.654741 Newtons.

From the experiment we can see that the minimum force increments reached by the robot were higher than the theoretical force resolution of the system. This means that the position resolution of the manipulator was larger than the one provided by the manufacturer. It is recommended to mention that the position resolution is not the same in all points within the envelope of the manipulator, it depends on the value the Jacobian matrix takes which varies as the position of the manipulator changes.

An experiment was set up to determine the real value of the mentioned resolution. The robot program written provided a loop in which the robot moved from one point to another with a determined increment, then read the position from the joint encoders, and displayed both the position read by the encoders (forward kinematics position), and the position determined by the program (commanded position). See Appendix A-5.

Seventy seven loops were required to move the manipulator one millimeter, from the point 400 to the point 401 along the x axis, with an increment of 0.013 mm. In Figure 4.20 , the behavior of the commanded position versus the position read by the encoders is shown. We can observe that both curves look like staircases, but the size of the step is larger in the curve representing the position read by the joint encoders. This graph also shows the large X difference mentioned in section 4.2.

It is necessary to determine the position resolution at each point in order to observe and understand the variations at the least significant digit. Two other position increments were selected in order to compare and determine the real position resolution of the commercial manipulator.

In Figure 4.21 the theoretical position resolution can be seen as a straight line, being always constant and equal to 0.005 mm throughout 92 loops. But, this is not the case of the real position resolution. It varies from 0.0053 mm to 0.0612 mm and 0.0204 mm was the value with the most occurrences. In Figure 4.22, the commanded position increment was 0.013 mm -the provided by the manufacturer-, 0.0204 mm was the value with the most occurrences within the 77 loops and it varied between –0.0054 mm and 0.0613 mm. The last case shows in Figure 4.23, a commanded position increment of 0.020 mm within the 50 loops, being 0.0054 mm the lower limit, 0.0612 mm the upper limit and 0.0204 mm the one with the most occurrences.

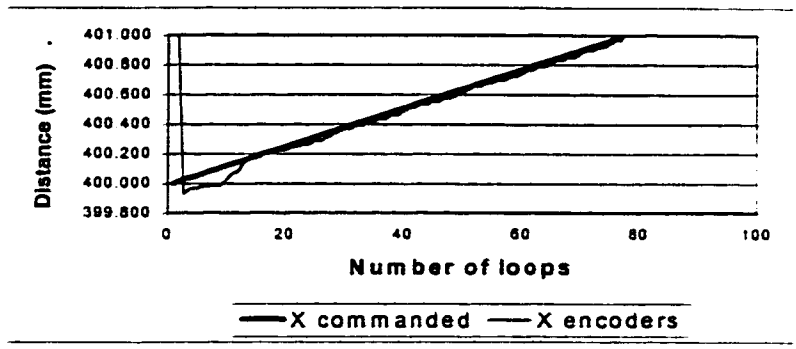


Figure 4.20 Commanded position vs. position read by the encoders
Position increment = 0.013 mm.

It can be seen from the previous experiments that the value with the most occurrences is the same in the three cases. Therefore, it can be concluded that the real position resolution of the manipulator is 0.0204 mm. This means that the real force resolution of the system is 1.0274 N instead of 0.654741 N. In other words, the minimum force increment that the system is capable of doing is 1.0274 N. It must also be said that the real position resolution is not constant and varies. Therefore, the force resolution of the system is not constant as well.

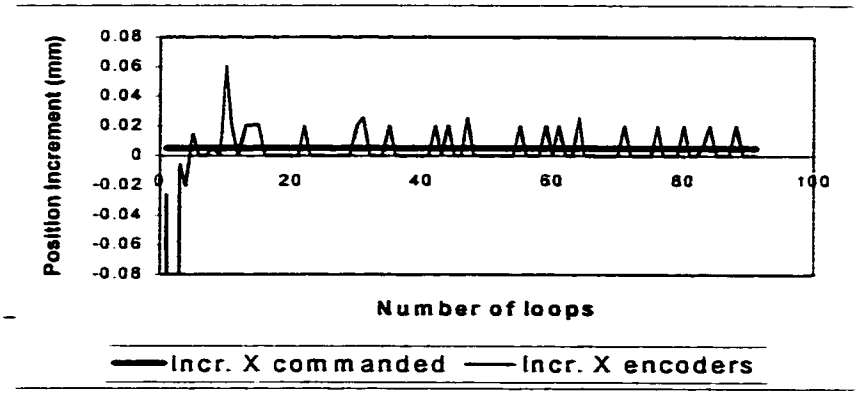


Figure 4.21 Position resolution
Increments of 0.005 mm.

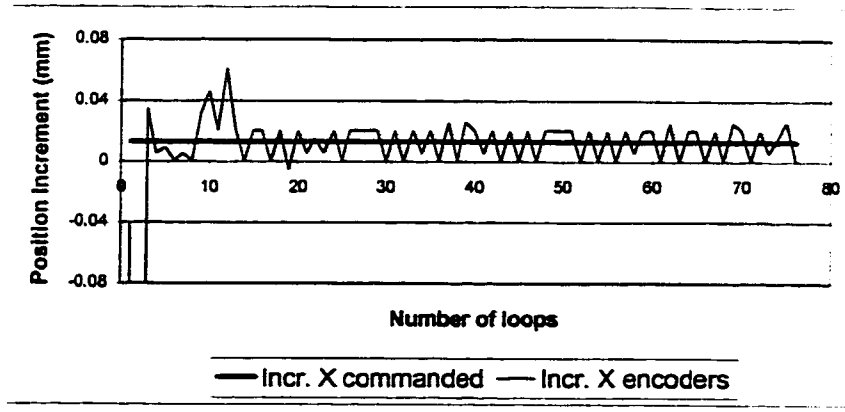


Figure 4.22 Position resolution
Increments of 0.013 mm.

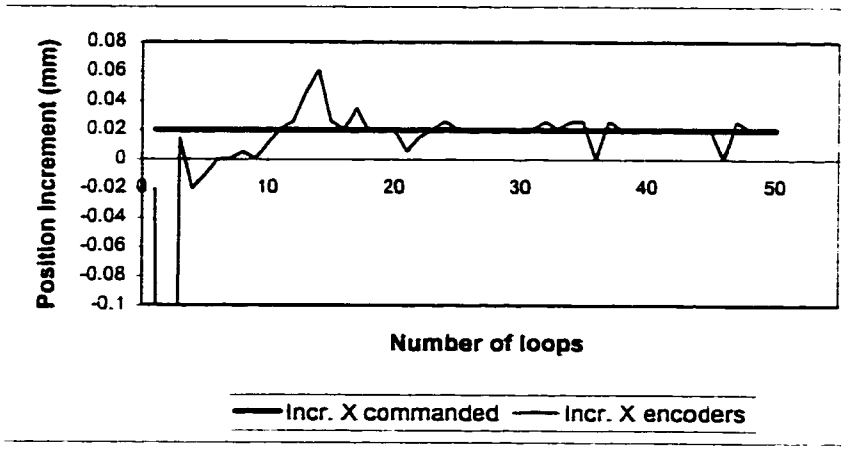


Figure 4.23 Position resolution
Increments of 0.020 mm.

4.3.1.3. IMPROVED COMPUTER ALGORITHM

In section 4.3.1.1 we described the computer algorithms used for the one degree of freedom force control experiments. They consisted of two robot programs running in

parallel and synchronized by the use of software signals. A new version was written in order to improve the performance of the system. See Appendix A-5. This new program called "Force.control" unifies the two original programs run in parallel.

The main distinction between them is that the improved version has a routine to reach the constraint, in which the manipulator starts its motion at a non-contact point and then moves gradually along the force sensor normal axis until it detects a force change, when the tool gets in contact with the constraint.

Once the constraint and the tool are in contact, the force/torque sensor reads the forces applied by the robot, which are assumed to be minimal due to the low speed of the manipulator (2% of its normal speed) and the small position increments (0.1 mm). Then, when the force is between 0.1 N and 5 N, that position is defined as the new start point for the one degree of freedom experiment. The reason for defining such force tolerance is only to locate the first contact point between the manipulator and the constraint.

A delay of 5 seconds was used to stabilize the system before the execution of the path modification routine. Then all the variables are defined, the path modification loop initiates moving the manipulator to the start point already set, gets the force readings, calculates the force error, determines the path correction factor used in the alter mode instruction, and finally, calls the same main_loop subroutine to displace the manipulator along the y axis when the force along the normal axis is within the force tolerance of +/- 0.7 N.

As it can be seen, it is a friendlier program, easier to read, understand and debug. Unfortunately, it did not improve the performance of the system. Instead of having software signals to synchronize and stabilize the system, now we have a delay instruction in line 39 of the program, to stabilize the force sensor due to the dynamics of the system.

Experiments were run using this program and the results obtained were very similar to the ones obtained with the original robot programs. Using the same force ranges, we conclude that the best performance of the system was reached for a variation of +/- 2 N. Foam was the constraint that yielded the best results. The number of loops, loops per millimeter, loops per second, speed (mm/s) and number of cycles with more than two loops per cycle remained almost identical to the ones obtained previously for the three constraints. Therefore, these results will not be shown in the appendices to avoid redundancy.

4.3.2 CONTOUR TRACKING FORCE CONTROL

In Section 3.3 of Chapter 3 the following of a flat and a curved surface was analyzed from a theoretical point of view. In the present Section, we will discuss the force control algorithms written, experiments conducted and results obtained when the manipulator is commanded to track a surface while applying a constant normal force on such constraint, using a position controller. The reason for including this section in the thesis is to study the feasibility of indirect hybrid position/force control.

4.3.2.1 CONTOUR TRACKING OF A FLAT SURFACE

The contour tracking of a flat surface is considered to be the simplest case of contour tracking of a constraint. It was decided to include this experiment in order to analyze the performance of the system and have a comparison base for the results obtained with the contour tracking of curved constraints.

When running the one degree of freedom force control experiments, we also run the contour tracking of a flat surface experiment. Once the measured force is close to the desired force within the force tolerance specified in the robot program, the manipulator proceeds to track the flat surface of the object, which consists of displacements of 1 mm along the tangential axis.

In Figures 4.13, 4.16, and 4.19 we can see that for the three materials, the best performance of the system was obtained for a force tolerance of +/- 2 N. Within this force tolerance, the robotic manipulator took 38.52 s to track a portion of 47 mm of the piece of wood, giving the average speed of 1.22 mm/s. To track the same distance in the piece of wax the manipulator required 37.9 s yielding an average speed of 1.24 mm/s. For the piece of foam, the manipulator took 34.3 s to track the same distance with an average speed of 1.37 mm/s.

Out of the three materials analyzed foam was the material that yielded the best contour tracking performance, as shown in the previous analysis as well as in Table 4.2.

Such speed depends almost exclusively on the number of loops required to reach the desired force. The less loops per cycle, the less time required to reach the desired force along the x axis. Therefore, it can be concluded that the performance of the following of a surface is affected by the control of the force and not by the control of the position, when using a commercial controller.

4.3.2.2 CONTOUR TRACKING OF A CURVED SURFACE

This is the last experiment conducted to analyze the performance of the system when using indirect hybrid position/force control. As explained in section 4.3.2.1 a piece of wax was chosen as the constraint with a circular shape. Amongst the three materials used for the one degree of freedom force control experiment, wax was selected because the system performed better than with the piece of wood and worse than with the piece of foam, representing the intermediate case.

In Appendixes A-7 – A-10 the robot programs used are presented. The pseudo codes used for each one of the programs are displayed next:

```
 1 PROGRAM "curve.control()"
 2  RESET SOFTWARE SIGNALS
 3 USER'S INPUT "SELECT INTERPOINT SPEED"; user sets interpoint speed
 4  MOVE TO NON-CONTACT LOCATION
 5 WAIT FOR MOVE INSTRUCTION COMPLETION
 6 ZERO THE FORCE SENSOR
 7 CALL SUBROUTINE "normal.force ()"; to find tangent force
 8 EXECUTE "curve.path()"; "curve.control and curve.path" running in parallel
 9 ENABLE ALTER MODE
10 DO
```

```
11      MOVE TO 1st CONTACT POINT; defined in subroutine "normal.force()"
12      WAIT FOR MOVE INSTRUCTION COMPLETION
13      SEND SOFTWARE SIGNAL; to indicate to "curve.path()" that robot has
                                                    moved
14      WAIT FOR SIGNAL; "curve.path()" sends signal when force readings
                                        and contour tracking are completed
15      RESET SOFTWARE SIGNAL
17 UNTIL CONDITION IS SATISFIED
18 DISABLE ALTER MODE
19 END
```

The program "Curve.control" is similar to the program "force.control" explained in Section 4.3.1.1. The main difference is that it requests the user to introduce the interpoint speed of the manipulator. The interpoint speed is an instruction used with the V+ language that allows the programmer to specify the speed between two points, as a percentage of the normal speed of the manipulator when commanded to move. It is not the distance moved divided by the time required to complete the total displacement. The latter will be defined as average speed of the manipulator. Another key difference in the programs is that the Alter mode is set with respect to the tool's coordinate frame.

```
 1 PROGRAM "curve.path()"
 2 USER'S INPUT "SELECT DESIRED FORCE TO APPLY: "
 3 USER'S INPUT "SELECT FORCE TOLERANCE: "
 4 SET TIMER (1) = 0
 5 SET TIMER (2) = 0
 6 DO
 7      WAIT FOR SIGNAL; to tell to "curve.path()" that robot has moved
 8      RESET SIGNAL
 9      x = x+1
10      counter = counter+1
11      GET FORCE READINGS
12      errorx = desired force – normal force read by force sensor
13      correctx = errorx *  stiffness coefficient; distance to reach desired force
14      IF ABS(normal force) > ABS(tangential force) THEN
```

```
15          CORRECT TRAJECTORY ALONG TOOL'S NORMAL AXIS
16      END IF
17      IF (-force tolerance <= errorx <= force.tolerance) THEN
18            CALL SUBROUTINE "curve.main()"; to track curved surface
19      END IF
20      INSERT 0.1 SECOND DELAY; to make sure signals are read
21      SEND SOFTWARE SIGNAL; sensor read forces and alter mode
                                executed
22 UNTIL CONDITION IS SATISFIED
23 TYPE "TOTAL TIME = ", TIMER(2), " SECONDS"
24 END
```

The program Curve.path is comparable to the program path.modification used in Section 4.3.1.1. The only difference is that it requests the user to introduce the desired force to apply on the constraint and the force tolerance to track the surface.

```
 1 SUBROUTINE "curve.main()"
 2 CORRECT TRAJECTORY ALONG TOOL'S X, Y AND Z AXIS
 3 cycle = cycle+1
 4 IF TOTAL ARC LENGTH TRACKED THEN
 5      SATISFY CONDITIONS FOR DO-UNTIL LOOPS; used in programs
                curve.control() and curve.path()
 6      TYPE "arc length = ", tangential increments*number of cycles
 7 END IF
 8 PRINT FACTORS AFFECTING PERFORMANCE
 9 RESET TIMER(1)
10 RETURN
11 END
```

The subroutine Curve.main uses the Alter mode to rotate the tool and move the manipulator along the tool tangential and normal axis. It displays the total number of cycles, loops per cycle, the x and y coordinates that describe the position of the end effector, the force in x and y applied by the robot, and the total time used per cycle. It

also displays the arc length of the constraint. It is by doing the rotation of the tool and the displacement along the tool's tangential axis that the manipulator tracks the contour of a curved surface.

```
1 SUBROUTINE Normal.force()
2 ENABLE ALTER MODE
3 DO
4       MOVE TO NON-CONTACT POINT
5       MODIFY TRAJECTORY ALONG WORLD'S NORMAL AXIS
6       GET FORCE READINGS
7       IF THERE IS CONTACT THEN
8               DEFINE CONTACT POINT
9       END IF
10 UNTIL THERE IS CONTACT
11 DO
12      GET FORCE READINGS
13      IF tangential force > 10% of normal force THEN
14          alfa = ARCTANGENT(tangent force/normal force); angle for rotation
                                                    of the tool
15          ROTATE TOOL BY alfa DEGREES
16          delay .1
17      END IF
18 GET FORCE READINGS
19 IF tangential force <= 10% of normal force THEN
20              SATISFY CONDITION
21 END IF
22 UNTIL CONDITION SATISFIED
23 END
```

The subroutine "Normal.force" is used to find the angle of the normal force applied by the robot when it is in contact with the constraint. It is an easy program that determines the angle alfa by the which the robot's tool has to be rotated. In Chapter 3, Section 3.4 the reader will find a more detailed explanation regarding this angle. The reason for making the tool rotate and using therefore task coordinates, is that the case

may be given in which a special design of the tool may be required with only one side of it in contact with the constraint. In this program in line 4, the manipulator is commanded to move to a non-contact point already defined in the program curve.control. The reason for using again this location is to be able to use the Alter mode. It is a V+ language requirement.

During the first experiment executions, it was noticed that the robot spent too much time trying to find the normal force after each displacement along the tool's tangential axis. This problem was solved by finding a new rotation angle using the known dimensions of the tool (see Figure 4.24). The radius "r" of the tool is equal to 25.1464 mm, the height "h" represents the displacement of the manipulator along the tangential axis. The angle "α" is equal to the arctangent of the height over the radius.

The height of the triangle represents the tangential increments. Different tangential increments were chosen to study the performance of the system. In Appendix B the results are summarized. Four force tolerances were selected: +/- 2N, +/- 3 N, +/- 5 N and +/- 7 N with tangential increments of 0.5 mm and 1 mm, and interpoint speeds varying from 2 % to 400 %.

Only one quarter of the circle was measured in order to reduce the amount of data to analyze. It was noticed that the manipulator tracked well the constraint for all the interpoint speeds, force tolerances and tangential increments selected. Even though the manipulator was commanded to move from 2% to 400% of its normal speed, the

difference in time between the slowest and the fastest speed was only 9.228 seconds for a force tolerance of +/- 2 N and tangential increments of 0.5 mm, which represents the worst case. Between 20% and 400% of the manipulator interpoint speed, there was practically no difference in the total time required to track an arc length of 100 mm. Due to the small size of the position increments along the tool tangential axis, the manipulator did not have time to accelerate enough and show an increment of the average speed and a reduction of time. The total number of loops required to track the constraint are proportional to the total time and to the average speed of the manipulator. A similar analysis can be made for the other force tolerances.

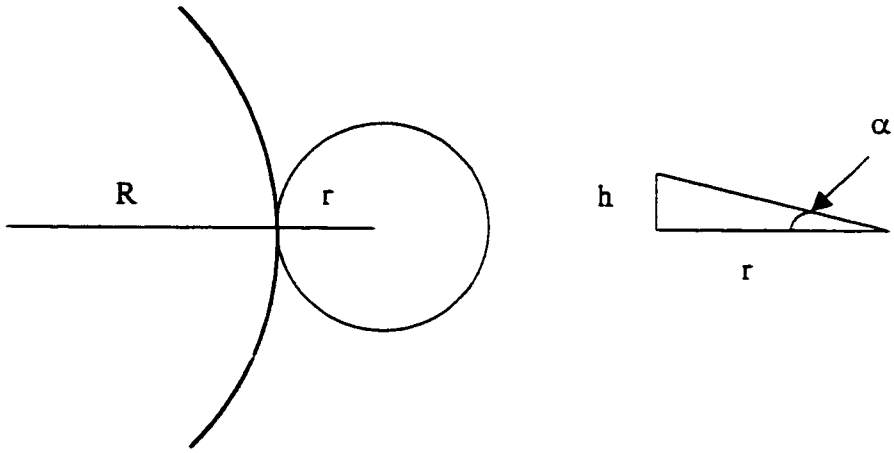
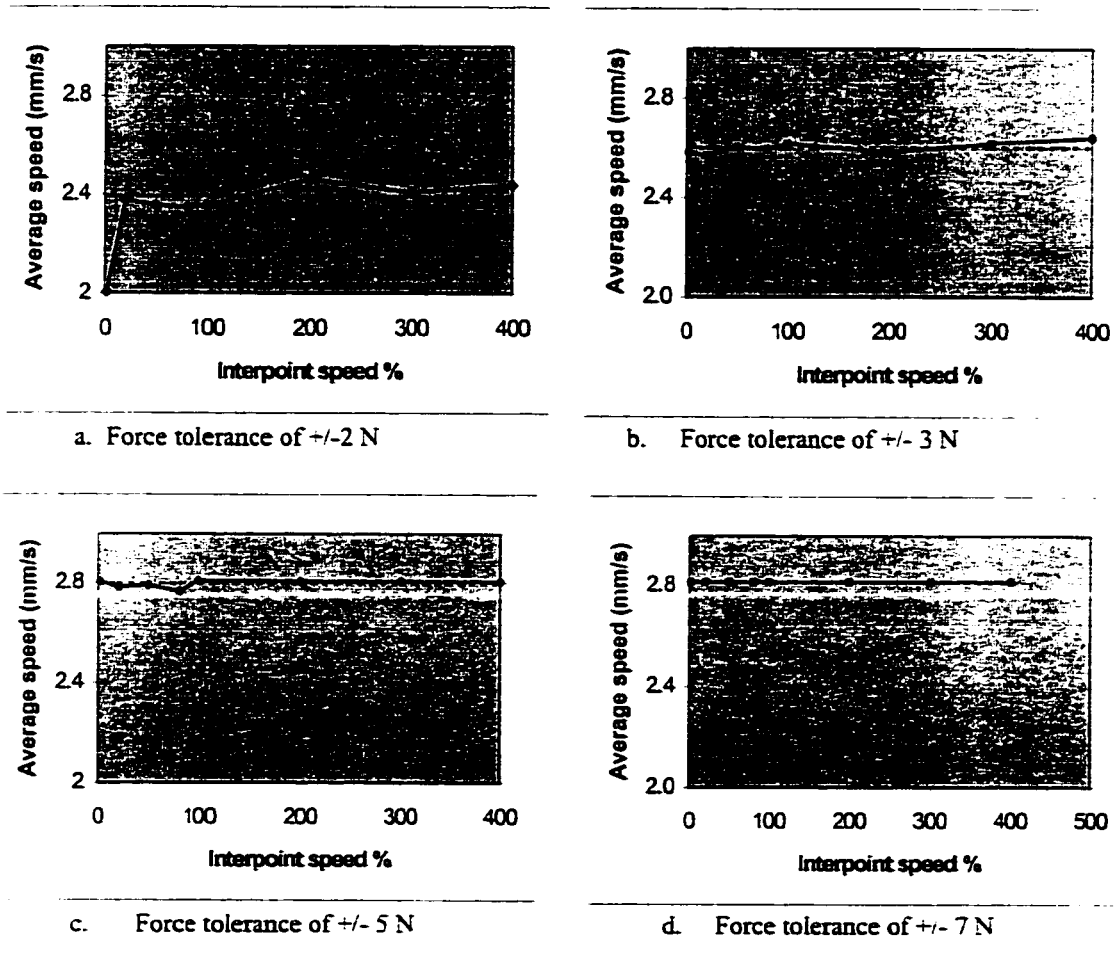


Figure 4.24 New rotation angle.

Figure 4.25 depicts the average speed of the manipulator versus the interpoint speed, for the four force tolerances selected and tangential increments of 0.5 mm. It can be observed that even though the ideal speed was 4.545 mm/s, the average speed of the manipulator was far below that value. For the first case (Figure 4.25.a) the average speed tends to 2.39 mm/s, shown in dashed lines, and only for an interpoint speed of 2% we

observe a lower average speed. In Figure 4.25.b the average speed is more constant, staying closer to the average speed of 2.6 mm/s. Figures 4.25.c and 4.25.d show that the average speed stays almost constant. This performance is obtained because when having a great force tolerance, the manipulator requires less loops per cycle to reach the desired force and therefore tracks the surface faster.



a. Force tolerance of +/-2 N

b. Force tolerance of +/- 3 N

c. Force tolerance of +/- 5 N

d. Force tolerance of +/- 7 N

Figure 4.25 Average speed vs Interpoint speed with tangential increments of 0.5 mm.

The reason for running experiments with force tolerances of +/- 5 N and +/- 7 N is that we were interested in studying the performance of the system when using the robot programs to measure the constraint without scratching or deforming it. It was assumed and proved that the manipulator moved faster with greater force tolerances. For both cases the system performed almost identically.

In Figure 4.26, the average speed of the manipulator versus the tangential increments are shown for an interpoint speed of 100%. It can be seen that for each force tolerance at each tangential increment the average speed increases. The average speed also increases with respect to the force tolerance. The manipulator moves faster with higher force tolerances and tangential increments.

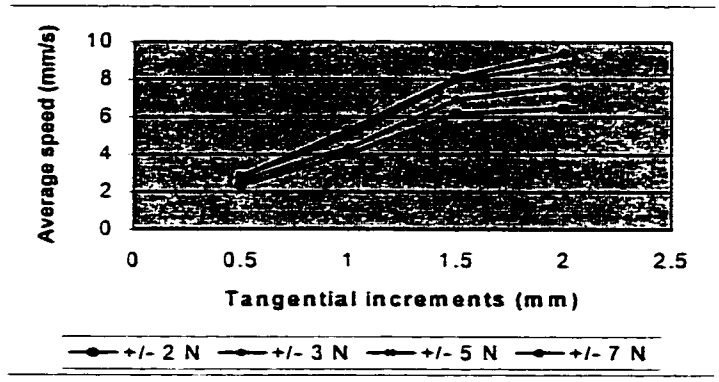


Figure 4.26 Average speed versus tangential increments for an interpoint speed of 100 %.

Shown Figure 4.27 are the performance of the normal forces at various interpoint speeds for a force tolerance of +/- 2 N and tangential increments of 0.5 mm. The forces displayed are only the ones within the force tolerance per cycle.

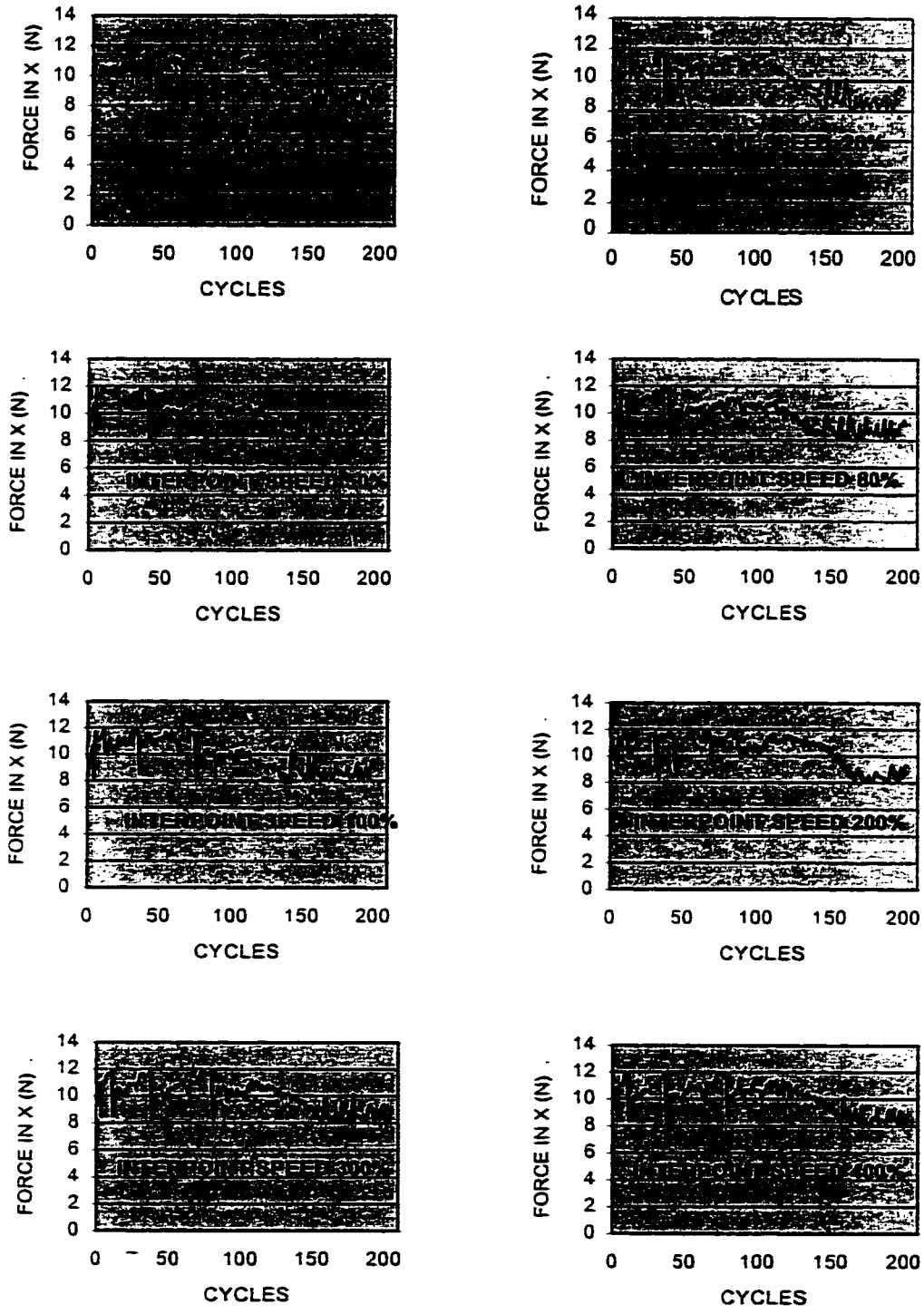
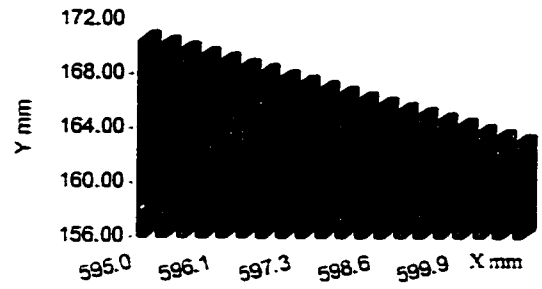
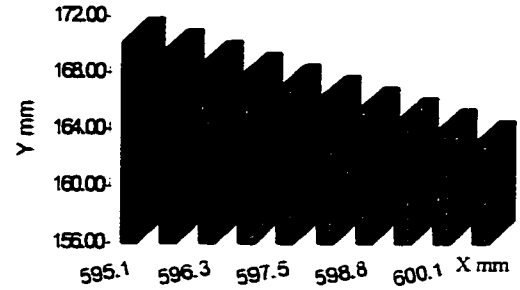


Figure 4.27 Normal forces with various interpoint speeds and a force tolerance of +/- 2 N.

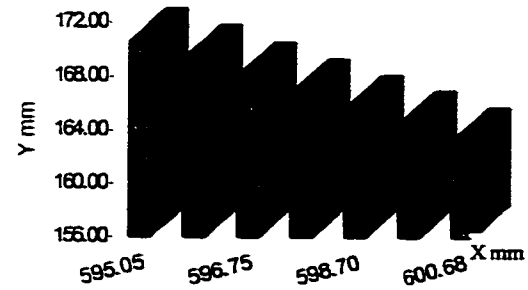
The last analysis made is related to the tracking accuracy. In Figure 4.28, tracking resolution can be analyzed regarding the average speed of the manipulator. Each one of the bars represents the displacement of the manipulator in world coordinates. The top of the bars shows the contact point between the manipulator and the constraint, and the sum of the tops together represents the arc length tracked. We can see in Figure 4.28.a that to track an arc length of 10 mm, the manipulator required 20 cycles with a tangential increment of 0.5 mm. To track the same arc length, the manipulator required 10 cycles with a tangential increment of 1 mm (Figure 4.28.b), 7 cycles with a tangential increment of 1.5 mm (Figure 4.28.c), and 5 cycles with a tangential increment of 2 mm (Figure 4.28.d).



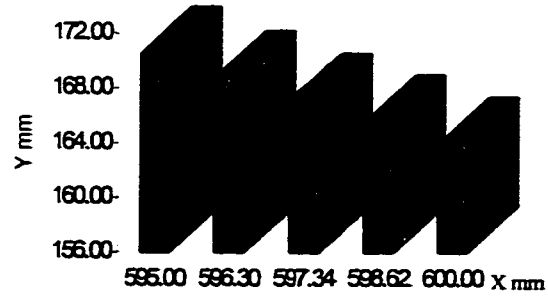
a. Tangential increments of 0.5 mm



b. Tangential increments of 1 mm



c. Tangential increments of 1.5 mm



d. Tangential increments of 2 mm

Figure 4.28 Tracking resolution.

In order to have a more accurate tracking of a surface, it is recommended that smaller tangential increments be used, which is the same as more cycles per arc length. Choosing a greater force tolerance means having a faster system, but it also means having a coarser tracked surface. The latter would be recommended for surface measurement, but not for a grinding task, where force accuracy is needed for a good finishing quality. We can see that there is a tradeoff between accuracy and speed.

Figure 4.29 depicts the surface tracked with a force tolerance of +/- 2 N and tangential increments of 0.5 mm. We can see a smooth tracked surface, where there are no big jumps between points. The graph does not show a coarse surface, it shows a good tracking accuracy.

Due to the difference of dimensions between the tool and the constraint, one limitation detected is the fact that if the radius of the surface is smaller than the radius of the tool, and the surface is concave, then the tool can not maintain single-point contact with the surface.

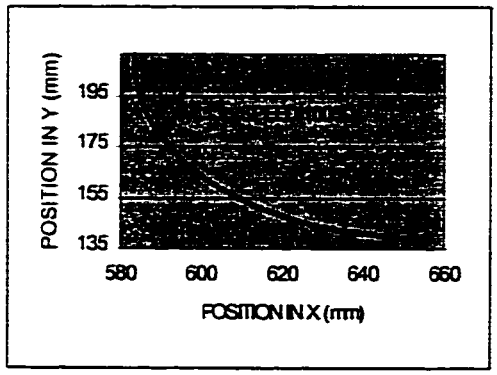


Figure 4.29 Curved surface tracked.

To conclude the section related to the contour tracking of a constraint, it must be said that the system performed well for both the contour tracking of flat and curved surfaces using the control law designed for the indirect hybrid position/force control method.

The three hypotheses stated in Chapter 3 were verified in this chapter. In page 48 was established that the compliance in the system is provided not only by the constraint, but also by the manipulator. In page 54 was confirmed that the acceleration along the y axis and the static friction will produce force readings along the y axis of the force sensor even when the manipulator is commanded to move only along the x axis. Finally, in page 85 was corroborated that the performance of the system was good for the contour tracking of flat and curved surfaces using the control law proposed, making the indirect hybrid position force/control method feasible.

CHAPTER 5

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

In this thesis, we formally characterized the difference between hybrid position/force control, as formulated in the robotics literature, and its indirect implementation using industrial (i.e. position-controlled) robotic manipulators, which we called "indirect hybrid position/force control". We then focused on the design and performance of the latter. In particular, we designed a generic algorithm for controlling the force applied by the robot to a constraint surface and another for tracking the contour of an unknown surface by maintaining a constant contact force. Both algorithms were successfully implemented on a planar industrial robot in contact with materials of differing stiffness. The contour following algorithm records the shape of the contour and allows the user to easily modify the tracking speed of the robot and the allowed force tolerance.

The main conclusion reached from this study is that the indirect hybrid control approach is a feasible method for simultaneously controlling position and force. Regarding the specific hypotheses stated in Chapter 3, the experiments support the first three hypotheses and contradict the last. Our conclusions with respect to these hypotheses are as follows:

1. The achievable force resolution of the system is proportional to the position resolution of the robot and inversely proportional to the stiffness of the

robot/constraint system.

2. For stiff materials such wood and wax, the compliance of the robot/constraint system is dominated by that of the robot itself (presumably due to the compliance of the motor pulleys).

3. Even during pure (1-dof) force control, positional errors are introduced in the tangential direction. This supports our prediction that such errors would arise due to a combination of inertial and stiction effects.

4. Contrary to hypothesis 4, it was observed that the force control loop sometimes required more than one loop to reach the desired force, but did not loop infinitely. This is because random noise in the force measurements would sometimes bring the force measurements into the desired force range, even though the robot's force resolution was insufficient.

Based on the results of this study, we recommend that a subroutine be used in the computer program that will determine the stiffness coefficient of the constraint based on the average of several stiffness experiments executed on various contact points of the same constraint. It is also recommended that smaller tangential increments be used in order to have a more accurate tracking of a surface. Finally, the last recommendation for future work is that a tool be designed to provide the compliance required in the system, similar to the Remote Center Compliance Device [Spong, 1987]. Such a tool should also

measure the deflection of the constraint to compensate for the readings of the joint encoders that provide the position of the end effector with the deformation of the constraint.

BIBLIOGRAPHY

- Adept Technology, Inc., "Force sensing module, user's guide", Version 1.8, San Jose California, July 1989.

- Adept Technology, Inc., "V$^+$ Reference guide", Version 8.0, San Jose California, July 1988.

- Anderson R., Spong M., "Hybrid impedance control of robotic manipualtors", IEEE Journal of Robotics and Automation, Vol 4, No 5, October 1988, pp 549-556.

- Bausch J., Gold F., "Design and adaptation of a brush-based active robotic deburring system", Robotics Today, Vol 9 No 2, Second quarter 1996, pp 1-7.

- Bone G., Elbestami M., "Active end effector control of a low precision robot in deburring", Robotics and Computer-Integrated Manufacturing, Vol 8, No 2, 1991, pp 87-96.

- Bone G., Elbestami M., Lingarkar R., Liu L., "Force Control For Robotic Deburring", in Transaccions of the ASME, Vol 113, September 1991, pp 395-400.

- Chiaverini S., Siciliano B., Villani L., "Force/position regulation for compliant manipulators", IEEE Transactions on Automatic Control, Vol 39, No 3, March 1994, pp 647-651.

- Fujie H., Mabuchi K., "Use of robotic technology to measure friction in animal joints", Clinical Biomechanics, Vol 11, No 3, 1996, pp 121-125.

- Fujie H., Mabuchi K., Woo S., Livesay G., Arai S., Sukamoto Y., "The use of robotic technology to study human joint kinematics: a new methodology", Journal of Biomechanical Engineering, Vol 115, August 1993, pp 211-217.

- Fujie H., Livesay G., Woo S., Kashimaguchi S., Blomstron G., " The use of a universal force-moment sensor to determine in-situ forces in ligaments: a new methodology", Journal of Biomechanical Engineering. Vol 117, February 1995, pp1-7.

- Goldenberg A., Liu G., "Robust hybrid impedance control of manipulators", IEEE International Conference on Robotics and Automation, Sacramento CA, April 1991, pp 1-6.

- Goldsmith P., " Stability of Hybrid Position/Force Control applied to robots with flexible joints" University of Toronto, 1995, pp 14-22.

- Harris N., Wang D., "Feedback stabilization and tracking of constrained robots", IEEE Transactions on Automatic Control, Vol 33, No 5, May 1988, pp 419-426.

- Hogan N., "Impedance control of a robotic manipulator. Part I: design" ASME Journal of Dynamic Systems, Measurements and Controls, Washington DC, Vol 107, March 1985, pp 1-7.

- Hogan N., "Impedance control of a robotic manipulator. Part II: implementation" ASME Journal of Dynamic Systems, Measurements and Controls, Washington DC, Vol 107, March 1985, pp 8-16.

- Hogan N., "Impedance control of a robotic manipulator. Part III: applications" ASME Journal of Dynamic Systems, Measurements and Controls, Washington DC, Vol 107, March 1985, pp 17-24.

- Hu Y., Davison E., "Position tracking and force control of robotic manipulators with a description system model", 33rd Conference on Description and Control", Lake Buena Vista FL, December 1994, pp 2388-2393.

- Kazerooni H., "On the robot compliant motion control", in Transactions of the ASME, Vol 111, September 1989, pp416-425.

- Kazerooni H., Bansch J., Kramer B., "An approach to automated deburring by robot manipulators", in Transactions of the ASME, Vol 108, December 1986, pp-354-359.

- Kazerooni H, Her M., "Automated robotic deburring of parts using compliance control", in Transactions of the ASME, Vol 113, March 1991, pp 60-66.

- Mason M., "Compliance and force control for computer controlled manipulators". In IEEE Transactions on Systems, Man, and Cybernetics. Vol SMC-II, No 6, June 1981, pp 418-432.

- Matsuno F., Kim E., Sakawa Y., "Dynamic hybrid position/force control of a flexible manipulator which has two degrees of freedom and flexible second link", Procedures of the 1991 International Conference on Industrial Electronics, Control and Instrumentation- IECON 1991, Vol 2, Los Alamos, California, 1991, pp 1031-1036.

- Ohto M., Mayeda H., " A hybrid position/force control for robot manipulators with position controllers", Proceedings for the 1991 International Conference on Industrial Electronics, Control and Instrumentation, 1991, pp 1037-1042.

- Paul R., Shimano B., "Compliance and control", in procedures of Joint Automatic Control Conference, July 1976, pp 694-699.

- Qu Z., "Globally stable I/O robust control of flexible joint robots", IEEE Journal of Robotics and Automation, 1993, pp 1004-1010.

- Raibert M., Craig J., "Hybrid position/force control of manipulators", ASME Journal of Dynamic Systems. Measurements and Controls. Vol 102. June 1981, pp 126-133.

- Salisbury J., "Active stiffness control of a manipulator in Cartesian coordinates", 19th IEEE Conference on Decision and Control, Albuquerque NM, December 1980, pp 95-100.

- Spong M.W., Vidyasagar M., "Robot Dynamics and Control", John Wiley & Sons, New York, 1989.

- Whitney D., "Force feedback control of manipulators fine motions", ASME Journal of Dynamic Systems, Measurements and Controls, June 1977, pp 91-97.

- Whitney D., Edsall A., "Development and control of an automated robotic weld bead grinding system", in Transactions of the ASME, Vol 112, June 1990, pp 166-176.

- Yao B., Tomizuka M., "Adaptive robust motion and force tracking control of robot manipulators in contact with compliant surfaces with unknown stiffness", ASME Journal of Dynamic Systems, Measurements and Controls, Vol 120, June 1998, pp 232-240.

- Yoshikawa T., "Dynamic Hybrid position/force control of robot manipulators- Description of hand constraints and calculation of joint driving force", in IEEE Journal of Robotics and Automation, Vol RA-3, No 5, October 1981, pp 386-392.

- Yoshikawa T., Sugie T., "Dynamic hybrid position/force control of robot manipulators – Controller design and experiment". IEEE Journal of Robotics and Automation, Vol 4, No 6, December 1988, pp 699-705.

APPENDIX A-1
COMPUTER PROGRAM: CONFIGURE_PARAMETERS

```
 1 .PROGRAM conf_par()
 2       LOCAL dim[], ft.flags, lower[], status, trip, upper[], vec[,],$error
 3 ; **** SET PARAMETERS ****
 4       out[0] = 4.44 ;CONVERSION FACTOR FOR FORCE IN NEWTONS
 5       out[1] = 25.4 ;CONVERSION FACTOR FOR LENGTH IN MM
 6       out[2] = 5 ; FORCE FILTER PARAMETER FOR NOISE LEVEL
 7       DEVICE (2, 0, status, fs.set.parm, 3, 1) out[], , NULL
 8       IF status < 0 GOTO 90
 9 ; **** ENABLE PROTECT MODE ****
10       out[0] = 7000
11       DEVICE (2, 0, status, fs.ena.protect, 1) out[]
12       IF status < 0 GOTO 90
13 ; **** ENABLE GUARD MODE ****
14       guarded = TRUE ; ENABLE GUARDED MODE
15       ft.flags = ^B10 ;TRIP CONDITION 1 FOR  FORCE AND 2 FOR TORQUE
16       dim[0] = 2 ; TRIP CONDITION 1 IS PLANAR (FX-FY)
17       dim[1] = 2 ; TRIP CONDITION 2 IS PLANAR (TX-TY)
18       dim[2] = 0 ; TRIP CONDITION 3 DISABLED
19       dim[3] = 0 ; TRIP CONDITION 4 DISABLED
20       vec[0,0] = 0 ; X COMPONENT OF TRIP CONDITION 1
21       vec[0,1] = 0 ; Y COMPONENT OF TRIP CONDITION 1
22       vec[0,2] = 1 ; Z COMPONENT OF TRIP CONDITION 1
23       vec[1,0] = 0 ; X COMPONENT OF TRIP CONDITION 2
24       vec[1,1] = 0 ; Y COMPONENT OF TRIP CONDITION 2
25       vec[1,2] = 1 ; Z COMPONENT OF TRIP CONDITION 2
26       lower[0] = -20 ; MINIMUN FORCE FOR TRIP CONDITION 1
27       upper[0] = 20 ; MAXIMUN FORCE FOR TRIP CONDITION 1
28       lower[1] = -80 ; MINIMUN TORQUE FOR TRIP CONDITION 2
29       upper[1] = 80 ; MAXIMUN TORQUE FOR TRIP CONDITION 2
30       CALL fs.config(guarded, ft.flags, dim[], vec[,], lower[], upper[], stat)
31       IF status < 0 GOTO 90
32 ; **** SET ZERO FORCE ****
33       DEVICE (2, 0, status, fs.set.zero)
34 ; **** ERROR SUBROUTINE ****
35    90  IF status < 0 THEN
36          CALL fs.error(stat, $error)
37          TYPE "ERROR DURING OPERATION: ", $error
38          PAUSE
39        END
40 .END
```

APPENDIX A-2
ONE DEGREE OF FREEDOM FORCE CONTROL COMPUTER ALGORITHM: FORCE.CONTROL

```
1 .PROGRAM force.control()
2       LOCAL point
3 ; APPLY A FORCE OF 10 NEWTONS IN X DIRECTION
4       SIGNAL -2020, -2021
5       SPEED 2 ALWAYS
6       MOVE TRANS(590.9,222.94,728.27,0,180,90)
7       BREAK
8 ; **** SET ZERO THE SENSOR ****
9       DEVICE (2, 0, , fs.set.zero)
10 ; **** ROUTINE TO MODIFY THE PATH ****
11      EXECUTE 1 path.modif()
12 ; **** STARTING ALTER MODE ****
13      SET point = TRANS(591.31,222.94,728.27,0,180,90)
14      ALTON () ^B11
15      DO
16        MOVES point
17        BREAK
18        SIGNAL 2021 ; ROBOT HAS MOVED
19        WAIT SIG(2020) ; FOR MOVING ROBOT'S ARM
20        SIGNAL -2021
21      UNTIL force.reach
22      ALTOFF
23 ; **** PRINTING LAST FORCES APPLIED ON SURFACE ****
24      TYPE "FORCE APPLIED CLOSE TO 10 NEWTONS AND CONTOUR
              TRACKING COMPLETED"
25      TYPE " FORCE IN X = ", f[0]
26      TYPE " FORCE IN Y = ", f[1]
27 .END
```

APPENDIX A-3
ONE DEGREE OF FREEDOM FORCE CONTROL
COMPUTER ALGORITHM: PATH.MODIFICATION

```
 1 .PROGRAM path.modif()
 2     LOCAL cycle, value, loop, errorx, correctx, a, force.reach, des.force
 3      a = FALSE
 4      des.force = 10 ;force of 10 Newtons to be applied on surface
 5      force.reach = FALSE
 6      cycle = 0, value = 0, loop = 0, x = 0
 7      TYPE "LOOP", /X4, "CYCLE", /X4, "DX", /X8, "DY", /X8, "FX", /X8, "FY"
 8      TIMER (2) = 0
 9 ; **** PATH MODIFICATION LOOP ****
10       DO
11          WAIT SIG(2021) ; for force reading execution
12          SIGNAL -2020
13          x = x+1
14          loop = loop+1
15 ;        **** GET FORCE INPUT ****
16          DEVICE (2, 0, , fs.get.force) , f[]
17          TYPE loop, /X7, cycle, /F10.2, DX(HERE), DY(HERE), f[0], f[1]
18 ;        **** CALCULATE ERROR ****
19          errorx = des.force-f[0]
20 ;        **** MODIFY DEVIATION ****
21          correctx = errorx*0.0998/5.0264 ;  stiffness coefficient of the wood
22 ; **** ALTER MODE TO APPLY FORCE ON SURFACE ****
23          IF ABS(f[0]) > ABS(f[1]) THEN
24             ALTER () correctx ;alter mode to apply force in x direction
25          END
26          fx = f[0]
27          fy = f[1]
28 ;   **** ROUTINE FOR DISPLACEMENT ALONG Y AXIS ****
29          IF (errorx <= 0.7) AND (errorx >= -0.7) THEN
30             CALL main.loop(fx, fy, cycle, value, point, a, x)
31          END
32          TIMER (1) = 0
33          WAIT TIMER(1) > 0.2
34          SIGNAL 2020 ;sensor read forces and alter mode executed
35       UNTIL a
36       force.reach = TRUE
37 .END
```

APPENDIX A-4
ONE DEGREE OF FREEDOM FORCE CONTROL
COMPUTER ALGORITHM: MAIN.LOOP

```
1 .PROGRAM main.loop(fx, fy, cycle, value, point, a, x)
2 ; **** ALTER MODE FOR CONTOUR TRACKING ****
3       IF (ABS(fx) > ABS(fy)) THEN
4          ALTER () , -1
5       END
6 ;   **** ROUTINE FOR DETERMINIG LENGTH TO TRACK ****
7       cycle = cycle+1
8       value = DISTANCE(HERE,point)
9       IF value > 230 THEN
10         a = TRUE
11      END
12 ;   **** PRINTING FACTORS AFFECTING PERFORMANCE ****
13      TYPE /X20, "EFICIENCY FACTOR = ", x, " loops/mm"
14      TYPE /X20, "TIME PER LOOP = ", /F5.2, TIMER(2), " seconds/mm"
15      TIMER (2) = 0
16      x = 0
17      TIMER (1) = 0
18      WAIT TIMER(1) > 0.4
19      RETURN
20 .END
```

APPENDIX A-5

COMPUTER ALGORITHM: POSITION RESOLUTION

```
1 .PROGRAM position.resolution
2       C = 0
3       INCREMENT = 0.013
4       MOVE  TRANS(400,128,853,0,180,90)
5       TYPE /X5, "COMMANDED POSITION",/X14, "FWRDKN POSITION"
6       DELAY 1
7       FOR X= 400 TO 401 STEP INCREMENT
8             C = C + 1
9             MOVE TRANS(X, 128, 853, 0, 180, 90)
10            BREAK
11            TYPE /G10.6, X, DX(HERE)
12      END
13      TYPE "NUMBER OF LOOPS = ",C
14      TYPE "        INCREMENTS = ",INCREMENT
15      END
```

APPENDIX A-6
ONE DEGREE OF FREEDOM FORCE CONTROL
COMPUTER ALGORITHM: FORCE.CONTROL
IMPROVED VERSION

```
1 .PROGRAM force.control()
2 ; PROGRAM TO APPLY A CONSTANT FORCE OF
3 ;10 NEWTONS ALONG THE X AXIS OF THE CONSTRAINT
4 ;     *****SET ZERO THE SENSOR*****
5       DEVICE (2, 0, , fs.set.zero)
6       SET initial.point = TRANS(580,239.5,724.25,0,180,90)
7       SPEED 2 ALWAYS
8 ;     *****STARTING ALTER MODE*****
9       ALTON () ^B11
10 ;    *****ROUTINE TO REACH THE CONSTRAINT*****
11      DO
12        i = 1
13        MOVES initial.point
14        ALTER () 0.1
15        DEVICE (2, 0, , fs.get.force) , f[]
16        fx = f[0]
17        IF (fx >= 0.1) AND (fx < 5) THEN
18          SET point = TRANS(DX(HERE),DY(HERE),DZ(HERE),0,180,90)
19        END
20      UNTIL fx >= 0.1
21      BREAK
22      DELAY 5
23 ;    **** DEFINING VARIABLES ****
24      des.force = 10
25      cycle = 0
26      value = 0
27      loop = 0
28      x = 0
29      TIMER (2) = 0
30      TYPE "LOOP", /X4, "CYCLE", /X4, "DX", /X8, "DY", /X8, "FX", /X8, "FY"
31 ;    **** PATH MODIFICATION LOOP ****
32      DO
33        MOVES point
34        a = FALSE
35        x = x+1
```

```
36          loop = loop+1
37 ;        **** GET FORCE INPUT ****
38        DEVICE (2, 0, , fs.get.force) , f[]
39        DELAY 0.5
40          TYPE loop, /X7, cycle, /F10.2, DX(HERE), DY(HERE), f[0], f[1]
41 ;        **** CALCULATE ERROR ****
42          errorx = des.force-f[0]
43 ;        **** MODIFY DEVIATION ****
44          correctx = errorx*0.0998/5.0264
45 ;        **** APPLY FORCE ALONG X DIRECTION ****
46          IF ABS(f[0]) > ABS(f[1]) THEN
47            ALTER () correctx
48          END
49          fx = f[0]
50          fy = f[1]
51 ;        **** DISPLACEMENT ALONG Y AXIS ****
52          IF (errorx <= 0.7) AND (errorx >= -0.7) THEN
53            CALL main.loop(fx, fy, cycle, value, point, a, x)
54          END
55        UNTIL a
56        ALTOFF
57 .END
```

APPENDIX A-7
CURVED SURFACE CONTOUR TRACKING ALGORITHM
COMPUTER ALGORITHM: CURVE.CONTROL

```
1 .PROGRAM curve.control()
2 ; THIS PROGRAM APPLIES A DESIRED FORCE ON A CONSTRAINT
3 ;WITHIN A SPECIFIED FORCE TOLERANCE AND CHOSEN SPEED
4 ; WRITTEN BY YAJAIRA HERRERA
5 ;
6       TYPE /C2
7       SIGNAL -2020, -2021
8       PROMPT "Select speed (%) 2-400: ", speed.value
9       SPEED speed.value ALWAYS
10      MOVE nocontact
11      BREAK
12 ; **** SET ZERO THE SENSOR ****
13      DEVICE (2, 0, , fs.set.zero)
14 ; **** ROUTINE TO MODIFY THE PATH ****
15      CALL normal.force ()
16      EXECUTE 1 curve.path()
17 ; **** STARTING ALTER MODE ****
18      ALTON () ^B1
19      DO
20        MOVES point
21        BREAK
22        SIGNAL 2021 ; robot has moved
23        WAIT SIG(2020) ; for force readings and contour tracking
24        SIGNAL -2021
25      UNTIL force.reach == TRUE
26      ALTOFF
27      TYPE "CONTOUR TRACKING COMPLETED"
28 .END
```

APPENDIX A-8
CURVED SURFACE CONTOUR TRACKING ALGORITHM
COMPUTER ALGORITHM: CURVE.PATH

```
1 .PROGRAM curve.path()
2       LOCAL cycle, value, loop, errorx, correctx, a, force.reach, force.des, force.tol
3       a = FALSE; force.reach = FALSE
4       cycle = 0, counter = 0, x = 0
5       PROMPT "Select desired force to apply: ", force.des
6      PROMPT "select force tolerance: ", force.tol
7       TYPE /C2
8       TIMER (2) = 0
9       TIMER (3) = 0
10 ; **** PATH MODIFICATION LOOP ****
11       DO
12          WAIT SIG(2021) ; for force reading execution
13          SIGNAL -2020
14          x = x+1
15          counter = counter+1
16 ;**** GET FORCE INPUT ****
17          DEVICE (2, 0, , fs.get.force) , f[]
18 ; ***** CALCULATE ERROR *****
19          errorx = force.des-f[0]
20 ; ***** MODIFY DEVIATION *****
21          correctx = errorx*0.101/5.029
22 ; **** ALTER MODE TO APPLY FORCE ON SURFACE ****
23          IF ABS(f[0]) > ABS(f[1]) THEN
24             ALTER () , correctx
25          END
26          fx = f[0]
27          fy = f[1]
28 ; ***** ROUTINE FOR DISPLACEMENT ALONG TOOL'S Y AXIS *****
29          IF (errorx <= force.tol) AND (errorx >= -force.tol) THEN
30             CALL curve.main(cycle, a, force.reach, x)
31          END
32          TIMER (1) = 0
33          WAIT TIMER(1) > 0.1
34          SIGNAL 2020; sensor read forces and alter mode executed
35       UNTIL a
36       TYPE "TOTAL TIME = ", TIMER(3), " SECONDS"
37 .END
```

APPENDIX A-9
CURVED SURFACE CONTOUR TRACKING ALGORITHM
COMPUTER ALGORITHM: CURVE.MAIN

```
1 .PROGRAM curve.main(cycle, a, force.reach, x)
2 ; **** ALTER MODE FOR CONTOUR TRACKING ****
3       ALTER () -1, 0.05, , , , -0.82
4 ;   **** ROUTINE FOR DETERMINIG LENGTH TO TRACK ****
5       cycle = cycle+1
6       IF cycle > 100 THEN
7          a = TRUE
8          force.reach = TRUE
9          TYPE "arc length = ", 1*cycle, "mm"
10      END
11 ;   **** PRINTING FACTORS AFFECTING PERFORMANCE ****
12      TYPE cycle, /X7, x, /F10.2, DX(HERE), DY(HERE), fx, fy, TIMER(2)
13      TIMER (2) = 0
14      x = 0
15      RETURN
16 .END
```

APPENDIX A-10
CURVED SURFACE CONTOUR TRACKING ALGORITHM
COMPUTER ALGORITHM: NORMAL.FORCE

```
 1 .PROGRAM Normal.force()
 2       a = FALSE
 3       ALTON () ^B1
 4        DO
 5          MOVES nocontact
 6          ALTER () 0.1
 7          DEVICE (2, 0, , fs.get.force) , f[]
 8          fx = f[0]
 9          IF (fx >= 0.1) AND (fx < 5) THEN
10             SET point = TRANS(DX(HERE), DY(HERE), DZ(HERE), RX(HERE),
                                 RY(HERE), RZ(HERE))
11          END
12       UNTIL fx >= 0.1
13       DO
14          DEVICE (2, 0, , fs.get.force) , f[]
15          fxo = f[0]
16          fyo = f[1]
17          IF ABS(fyo) > ABS(fxo*10/100) THEN
18             alfa = ATAN2(fyo,fxo)
19             DRIVE 4, alfa, 10
20             delay .1
21          END
22          DEVICE (2, 0, , fs.get.force) , f[]
23          fxf = f[0]
24          fyf = f[1]
25          IF ABS(fyf) <= ABS(fxf*10/100) THEN
26             a = TRUE
27             alfa = 0
28          END
29          TYPE "FXo= ", fxo, /X4, "FYo= ", fyo, /X4, "ALFA= ", alfa
30          TYPE "FXf= ", fxf, /X4, "FYf= ", fyf
31          TYPE /C1
32       UNTIL a
33 .END
```

APPENDIX B-1
CONTOUR TRACKING OF A CURVED SURFACE RESULTS
FORCE TOLERANCE +/- 2 N

Tangential increments of 0.5 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	49.854	337	6.760	2.006
20	42.094	264	6.272	2.376
50	42.452	268	6.313	2.356
80	42.596	269	6.315	2.348
100	42.484	268	6.308	2.354
200	40.626	250	6.154	2.461
300	41.534	259	6.236	2.408
400	41.008	254	6.194	2.439
Averages	42.831	271	6.319	2.335
Ideal	22	200	9.091	4.545
Ratios	1.947	1.356	0.695	0.514

Tangential increments of 1 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	27.584	195	7.069	3.625
20	25.814	178	6.895	3.874
50	27.898	198	7.097	3.584
80	26.398	184	6.970	3.788
100	24.884	169	6.792	4.019
200	25.402	174	6.850	3.937
300	25.070	171	6.821	3.989
400	25.492	175	6.865	3.923
Averages	26.06775	181	6.920	3.836
Ideal	11	100	9.091	9.091
Ratios	2.370	1.805	0.761	0.422

APPENDIX B-2
CONTOUR TRACKING OF A CURVED SURFACE RESULTS
FORCE TOLERANCE +/- 3 N

Tangential increments of 0.5 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	38.740	233	6.014	2.581
20	38.930	235	6.036	2.569
50	38.428	230	5.985	2.602
80	38.528	231	5.996	2.596
100	38.230	228	5.964	2.616
200	38.724	233	6.017	2.582
300	38.214	228	5.966	2.617
400	37.892	225	5.938	2.639
Averages	38.461	230	5.990	2.600
Ideal	22	200	9.091	4.545
Ratios	1.748	1.152	0.659	0.572

Tangential increments of 1 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	23.414	155	6.620	4.271
20	22.894	150	6.552	4.368
50	22.784	149	6.540	4.389
80	23.310	154	6.607	4.290
100	23.102	152	6.580	4.329
200	23.106	152	6.578	4.328
300	22.810	149	6.532	4.384
400	23.098	152	6.581	4.329
Averages	23.065	152	6.574	4.336
Ideal	11	100	9.091	9.091
Ratios	2.097	1.516	0.723	0.477

APPENDIX B-3
CONTOUR TRACKING OF A CURVED SURFACE RESULTS
FORCE TOLERANCE +/- 5 N

Tangential increments of 0.5 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	35.63	203	5.697	2.807
20	35.944	206	5.731	2.782
50	35.828	205	5.722	2.791
80	36.152	208	5.753	2.766
100	35.608	203	5.701	2.808
200	35.620	203	5.699	2.807
300	35.626	203	5.698	2.807
400	35.630	203	5.697	2.807
Averages	35.755	204	5.712	2.797
Ideal	22	200	9.091	4.545
Ratios	1.625	1.021	0.628	0.615

Tangential increments of 1 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	20.142	123	6.107	4.965
20	20.202	124	6.138	4.950
50	20.102	123	6.119	4.975
80	20.016	122	6.095	4.996
100	20.106	123	6.118	4.974
200	19.800	120	6.061	5.051
300	20.004	122	6.099	4.999
400	19.586	118	6.025	5.106
Averages	19.995	122	6.095	5.001
Ideal	11	100	9.091	9.091
Ratios	1.818	1.219	0.670	0.550

APPENDIX B-4
CONTOUR TRACKING OF A CURVED SURFACE RESULTS
FORCE TOLERANCE +/- 7 N

Tangential increments of 0.5 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	35.52	202	5.687	2.815
20	35.520	202	5.687	2.815
50	35.524	202	5.686	2.815
80	35.520	202	5.687	2.815
100	35.524	202	5.686	2.815
200	35.518	202	5.687	2.815
300	35.524	202	5.686	2.815
400	35.508	202	5.689	2.816
Averages	35.520	202	5.687	2.815
Ideal	22	200	9.091	4.545
Ratios	1.615	1.010	0.626	0.619

Tangential increments of 1 mm

SPEED (%)	TIME (s)	LOOPS	LOOPS/S	VELOCITY (mm/s)
2	18.770	110	5.860	5.328
20	19.092	113	5.919	5.238
50	18.544	108	5.824	5.393
80	18.970	112	5.904	5.271
100	18.984	112	5.900	5.268
200	18.760	110	5.864	5.330
300	18.868	111	5.883	5.300
400	18.864	111	5.884	5.301
Averages	18.857	111	5.880	5.303
Ideal	11	100	9.091	9.091
Ratios	1.714	1.109	0.647	0.583