Delicate Manipulation of Irregularly-Shaped Rigid Objects in a Stiff, Fragile Environment

by

Joseph Andrew Calzaretta

Bachelor of Science in Mechanical Engineering
Massachusetts Institute of Technology (1996)

Submitted to the
Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June, 1998

Signature of Author ______________________________
Department of Mechanical Engineering
May 8, 1998

Certified By ______________________________
Steven Dubowsky
Thesis Supervisor

Accepted By ______________________________
Ain A. Sonin
Chairman, Departmental Graduate Committee

Delicate Manipulation of Irregularly-Shaped Rigid Objects in a Stiff, Fragile Environment

Submitted to the Department of Mechanical Engineering
on May 8, 1998, in partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering

by

Joseph Andrew Calzaretta

Abstract

The automation of the crucible-charging portion of the CZ semiconductor production process with a robot assisted crucible-charging system (RACS) requires the investigation of certain control issues. This thesis addresses the problem of regulating contact forces and endpoint positions in a system in which there is little or no passive compliance present in the manipulator or the environment.

In order to address the challenge of regulating delicate contact forces while in contact with a stiff, fragile and known environment, a hybrid position/force control algorithm was developed and implemented. This implementation was executed in simulation, on a test manipulator system, and finally on an AdeptOne-based laboratory demonstration system. Simulation and experimental results are presented.

In order to resolve the additional difficulty of an environment of unknown orientation, a surface orientation observer algorithm has been developed which uses current and previous wrist force sensor data to produce its estimate. This observer is used in conjunction with the control algorithm to obtain force and position control in an unknown environment. Expected behavior of this algorithm is discussed, and simulation results are presented.

Integration of this work with the laboratory demonstration system is discussed, and suggestions for future research in this area are proposed.

Thesis Supervisor: Dr. Steven Dubowsky
Professor of Mechanical Engineering

Acknowledgements

I would like to thank Professor Steven Dubowsky for his guidance and assistance during the past two years. I wish to thank Anthony Leier, Vivek Sujan, and Melissa Tata, for their work on this project. Thanks also to my other colleagues at the Field and Space Robotics Laboratory, especially Karl Iagnemma, for general guidance and various contributions to my research.

This thesis describes research performed at the MIT Field and Space Robotics Laboratory under the sponsorship of Shin Etsu Handotai, Co., Ltd. Thanks to SEH for providing the financial support for this project.

I'd also like to thank my parents, sister, friends at MIT and around the country and world, and everyone I've ever met in my entire life.

Table of Contents

C Material Properties Testing

List of Figures

Chapter 1

Introduction

1.1 Background and Project Description

Shin Etsu Handotai, Inc. (SEH) produces silicon wafers via the Czochralski (CZ) process (Wolf and Tauber, 1986). One stage of this process involves the loading of irregularly shaped poly-crystalline silicon nuggets into fused silica crucibles. A charged crucible is shown in Figure 1-1, (Dubowsky, 1997).

A bed layer of small nuggets is formed at the bottom of the crucible. Then, a layer of large nuggets is carefully built touching the side wall, as the middle of the crucible is filled with bulk (small and large nuggets). Finally, a crown of large nuggets is built above the top of the crucible. Important constraints in this process include protection of the crucible from damage, minimization of silicon contamination, maintaining the required charge density, and achieving an appropriate side wall nugget contact orientation.

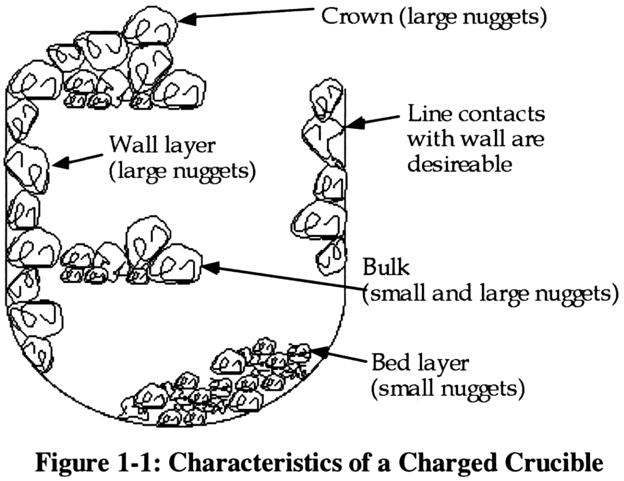

Crown (large nuggets)
Line contacts with wall are desireable
Wall layer (large nuggets)
Bulk (small and large nuggets)
Bed layer (small nuggets)

Figure 1-1: Characteristics of a Charged Crucible

Currently, the crucible charging process is performed manually. The ultimate goal of this project is to develop an automated process for packing the nuggets, known as the Robot Assisted Crucible Charging System (RACS). This shift to an automated system should facilitate larger charge densities and the use of larger crucibles in the manufacturing process.

1.1.1 The Laboratory Demonstration System

In order to achieve this goal, a laboratory demonstration system has been developed, depicted in Figure 1-2 (Dubowsky, 1997). This system is composed of three subsystems. The control system consists primarily of the AdeptOne manipulator, controller unit, control electronics, and computer running the control software. The vision/packing system contains the nugget scanning station and the overhead crucible scanner, as well as the computer which implements the packing algorithm (not shown). The grasping system consists of a custom end-effector with a three-joint wrist and a gripper. The purpose of this system is to address the important technical challenges and illustrate the feasibility of a RACS.

The automation of this process with a robot manipulator-based system presents several technical challenges. First, the RACS must be able to acquire information about the shape of a nugget to be placed and the landscape of the nuggets already in the crucible. From these data, the vision/packing system must determine the optimal location in the crucible to place the nugget so that charge density is maximized. This challenge is addressed by (Sujan, 1998). Secondly, the system end-effector must be able to grasp irregular nugget shapes and place nuggets in the crucible without contacting the crucible wall. This challenge is addressed by (Leier, 1998). Finally, the control system must be able to delicately manipulate and place the nuggets without damaging the crucible wall or disturbing the pre-existing layer of nuggets. It is this third challenge which concerns this thesis.

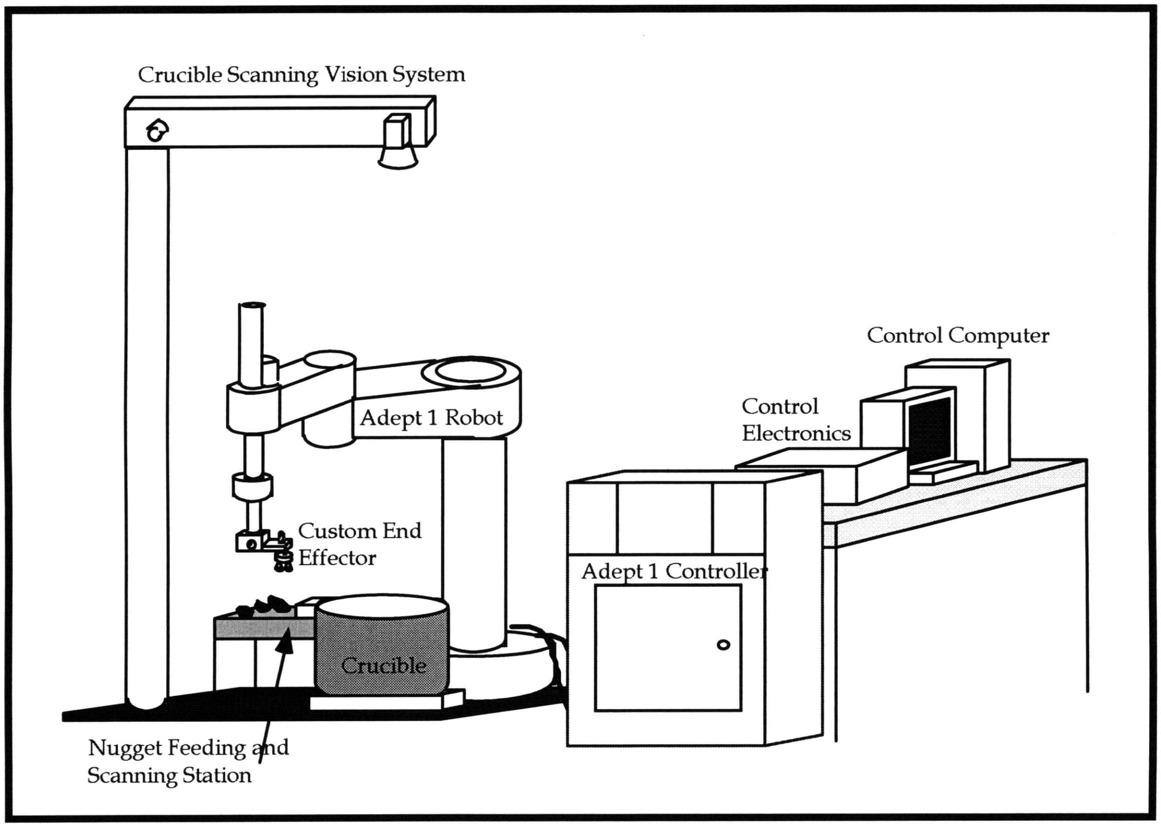


Figure 1-2: Laboratory Demonstration System

1.1.2 Control System Performance Specifications

Crucible charging involves five distinct subtasks for the control system, shown in Figure 1-3. These subtasks are nugget acquisition, nugget scanning, slew motion, wall and crown building, and bulk filling. Certain performance specifications for the control system were determined by the requirements for proper operation of the vision and packing system, (Sujan, 1998), and the gripper mechanism (Leier, 1998). Performance specifications regarding the minimization of crucible damage were determined by the material properties of silicon and glass, (see Appendix B) and through experimental testing, (see Appendix C).

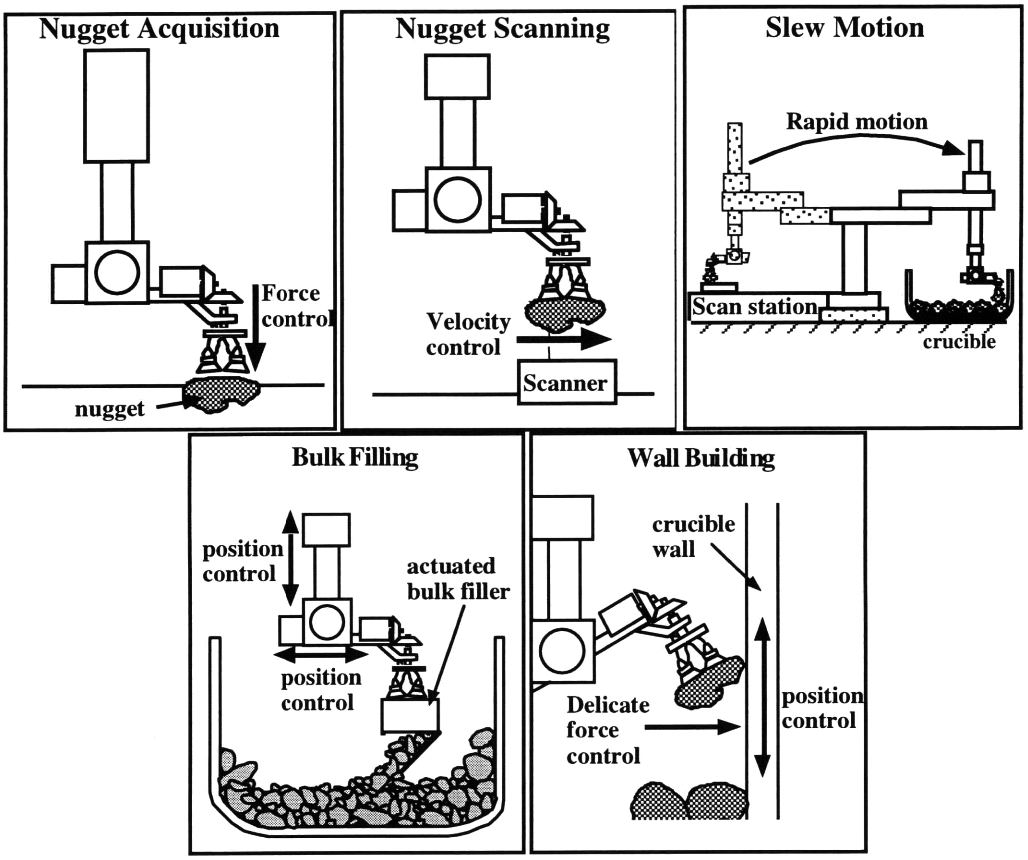


Figure 1-3: RACS Control Modes

In the nugget acquisition mode, the robot end-effector must grasp a nugget with the gripper. The end-effector must initially move downward at 15 cm/sec. Once contact is made and detected, the nugget is gripped. The manipulator must maintain a downward force of less than 20 N. The end-effector then moves upward at 15 cm/sec.

In the nugget scanning mode, the manipulator must pass a nugget over the nugget scanner. The end-effector is required to move horizontally at a constant speed of 3 cm/sec. The position must be regulated to within 0.5 mm.

During slew motion, the manipulator must rapidly move the nugget from the scanner to the interior of the crucible without contacting the crucible. This motion is to take place in approximately 2.5 seconds and the position must be regulated to within 1 cm for the entire motion.

In the wall building mode, the nugget must be brought against the crucible wall and lowered onto the existing layer of nuggets. The nugget is moved toward the crucible wall at no more than 10 cm/sec. Contact is made and detected. The manipulator must maintain a contact force of less than 2 N without losing contact as the nugget slides down the wall. The nugget makes contact with the existing nugget layer and must be positioned in a designated location without disturbing the existing nuggets. The manipulator releases the nugget and leaves the crucible interior. The crown building mode is similar, but it does not involve contact with the crucible wall.

In the bulk-filling mode, the manipulator must bring a bulk filler device to the interior of the crucible and release fill nuggets. The position must be regulated to within 1 cm.

1.1.3 Control System Key Technical Challenges

The wall and crown building mode is the most critical and challenging of the five control modes. The other four modes involve robotic manipulation under either pure position control or pure force control, the feasibility of which have been established previously (Craig, 1989; Whitney, 1987). The wall building process, however, requires the simultaneous positioning of a nugget while maintaining delicate contact forces. This requires another control algorithm.

When controlling contact force, it is helpful to know the surface orientation. When the contacted surface is the crucible wall, the contact surface orientation is known beforehand. An additional challenge is introduced when the nugget comes into contact with previously placed nuggets. The surface orientation of these nuggets is not necessarily known, and an algorithm must be used which can regulate force and position without *a priori* information.

1.2 Literature Review

Control of both contact force and endpoint position with a robot manipulator is a challenging task which has been studied by many researchers (Whitney, 1987). The two major control schemes in use today are hybrid position/force control and impedance control.

Hybrid position/force control splits the environment into force and position control domains, and then performs conventional control in each domain (Raibert and Craig, 1981). This can be performed with or without a dynamic model of the manipulator and, in theory, perfect position tracking in the position domain can be obtained without the generation of excessive contact forces in the force domain. The advantage of this scheme is its applicability to stiff environments. One disadvantage is that switching position and force domains during contact can cause instability. Another potential drawback of hybrid control is that the contact environment must be well known to achieve acceptable results.

Impedance control employs a dynamic model to create specified equations of motion of a manipulator (Hogan, 1984). Thus, the manipulator behaves as if it possesses a chosen inertia, damping, and stiffness. If a dynamic model is not available, impedance control reduces to stiffness control. When contact force feedback is incorporated in the model, force control can be achieved (Hogan, 1987). The advantage of impedance control is its stability and applicability during the entire contact control task; no switching is required when contact is made. The major disadvantage is that it may result in substantial position errors. The introduction of an integrator in a "position domain" would alleviate this problem, but such a scheme is essentially a variant of the hybrid controller.

Much work has been performed in the area of achieving stable contact behavior with a manipulator. Implicit contact control can be performed if significant errors in the force response are acceptable (Mills, 1996). Various schemes have been devised which incorporate sensor information such as a wrist force/torque sensor (Seraji *et al.,* 1996), and an optical proximity sensor (Li, 1996).

Performing accurate force control in an unknown stiff environment is also a topic of research interest (Whitney, 1987). A scheme for compliant manipulators in which the system estimates the location of the surface as the contact point moves has been devised (Cutkosky, 1985). This scheme assumes a rather compliant end-effector so that the contact forces stay within a window determined by this compliance. For higher stiffness end-effectors, a simple strategy has been developed which minimizes contact forces by following the path of least resistance (Niemeyer and Slotine, 1997). However, this scheme assumes that the interaction is constrained for both positive and negative forces. Manipulators pushing against a surface cannot apply negative contact forces, and thus the path of least resistance is away from the surface. A method which uses both sensed force and velocity data to estimate the contact point location and surface normal direction has also been devised (Muto and Shimokura, 1993).

Systems in which a robotic manipulator is used to pack objects into a container have been studied previously. A similar system to the RACS in development today is a frozen fish packaging system containing an AdeptOne manipulator and an overhead vision system (Neal *et al.,* 1997). Another such system is a manipulator-based system which retrieves irregularly-shaped parts from a bin using a 3-D sensor and a special gripper (Bach *et al.,* 1985). These systems face some of the technical challenges as the RACS, although the focus of these studies have been vision and grasping issues, and the performance specifications for these systems are much less stringent.

1.3 Purpose of this Thesis

This thesis addresses the problem of regulating contact forces and endpoint positions in a system in which there is little or no passive compliance present in the manipulator, the end-effector, or the environment. When the RACS is implemented, the end-effector may in fact exhibit significant compliance (Leier, 1998), which simplifies the control task. The more intellectually challenging task of achieving the desired performance with a high-stiffness system is explored.

In order to meet the challenge of regulating delicate contact forces while in contact with a stiff, fragile and known environment, a hybrid position/force control algorithm was developed and implemented. This implementation was executed in simulation, on a test manipulator system, and finally on the laboratory demonstration system depicted in Figure 1-2. The experimental results are promising and meet the process requirements, yet indicate that further research should be performed to ensure that the performance specifications will always be met.

In order to resolve the additional difficulty with an environment of unknown orientation, a surface orientation observer algorithm has been developed. This algorithm uses current and previous force sensor data to produce an estimate of the surface orientation. This observer is used in conjunction with the control algorithm to obtain force and position control in an unknown environment. This system is implemented in simulation. The results are encouraging and show potential for experimental verification.

1.4 Outline of this Thesis

This thesis is divided into six chapters. The first chapter serves as an introduction and summary of the thesis. The second chapter presents the hybrid position/force control algorithms selected for the wall-building process. Simulation results are presented for a planar manipulator in contact with a stiff environment.

Chapter 3 presents experimental investigations of the hybrid position/force control algorithms when the end-effector is in contact with a stiff environment of known surface orientation. These tests are performed on two robotic manipulator systems. The first system consists of a Puma 250 manipulator with force data provided by a uniaxial load cell. The second experimental system consists of an AdeptOne manipulator with force data provided by a six-axis wrist force/torque sensor. The latter system is the core of the laboratory demonstration system for the RACS.

Chapter 4 presents the algorithm whereby the control system determines the unknown orientation of the contact surface. An instantaneous estimate of the surface normal and its associated uncertainty can be obtained from force sensor data. This information is then incorporated by a constrained Kalman filter routine, which provides the system's best estimate of the normal vector as a function of current and previous data. The expected behavior of the algorithm is discussed.

Chapter 5 explores the behavior of the surface estimation algorithm in simulation. The algorithm is first tested in an open-loop sense, where it merely observes the system behavior and does not contribute to the manipulator motion. The algorithm is then used in combination with the hybrid control algorithms of Chapter 2; the surface estimate defines the position and force domains used in the control algorithm.

The final chapter summarizes the conclusions regarding the RACS control system, and discusses integration of this work with the laboratory demonstration system. It also offers suggestions for future research work in this area.

The appendices to this thesis provide information about particular topics which were necessary in the completion of the work presented. Appendix A lists the kinematic properties of the two experimental manipulator systems. Appendix B provides relevant material properties of silicon and glass, and Appendix C presents the results of empirical testing to determine the conditions which result in crucible damage. Appendix D shows the circuit diagrams which were designed to interface the control system with the gripper/wrist subsystem.

Chapter 2

Hybrid Position/Force Control Algorithms

2.1 Hybrid Position/Force Control

The hybrid position/force control algorithms selected for the RACS wall-building mode are derived and presented in this section.

2.1.1 Division of Environment into Subspaces

In order to achieve both force and position control of a manipulator, it is first necessary to characterize the environmental constraints. It is beneficial to formulate these constraints in a Cartesian task-space coordinate system, which describes the manipulator endpoint position and orientation with respect to a fixed global reference frame. It has been proposed that controllers based on this Cartesian formulation produce more successful results than do purely joint-based methods (Maples and Becker, 1986).

Hybrid force-position control divides task-space into two orthogonal subspaces (Raibert and Craig, 1981). One subspace corresponds to those directions in which endpoint motion is admissible and contact forces are determined by the environment. This subspace is associated with position control. The complementary subspace corresponds to those directions in which contact forces can be applied but the endpoint position is determined by the environment. This subspace is associated with force control.

This division into subspaces assumes that the environment offers no resistance in one set of directions, and perfectly rigid in the complementary set. In the case of the wall-building process, the manipulator, nuggets, and crucible exhibit very high mechanical stiffnesses. The interaction between the nugget and crucible exhibits relatively low sliding friction (Leier, 1998-2). It is also assumed that the interaction between two nuggets exhibits low sliding friction (Sujan, 1998-2); this assumption does not include the effects of interlocking. The environment is indeed nearly rigid in the force direction, and compliant in the position directions; thus the environment lends itself to this characterization.

2.1.2 Manipulator Dynamics and Simplifications

For a manipulator with a vector θ of joint positions, the dynamic equations can be represented by

$$\tau = \mathbf{H}\ddot{\theta} + \mathbf{C}\dot{\theta} + f + g + \tau_e, \qquad (2 - 1)$$

where τ is the vector of torques applied to each joint, **H** is the configuration-dependent inertia tensor, **C** is a matrix representing centripetal and Coriolis terms, f is a frictional torque, g is a gravitational torque, and the final term τ_e is a torque transmitted to the joints via endpoint contact with the environment (Craig, 1989). If the parameters in this model are known very well, a robust hybrid controller can be implemented (Liu and Goldenberg, 1991).

As mentioned previously, the motions required of the manipulator during the wall-building process are very slow, so that the velocities $\dot{\theta}$ and accelerations $\ddot{\theta}$ of the joints are nearly zero. Since dynamic torques are small compared to the other torques in the system, the inertial, centripetal, and Coriolis terms can be neglected. The gravitational term varies with the joint configuration, and therefore changes slowly. This term can be estimated as a constant plus a disturbance torque. The friction term, which is typically difficult to model, can either be regarded as a disturbance or compensated for via a friction estimation method such as BaST control (Morel and Dubowsky, 1996). Therefore, the reduced model consists of just an endpoint and disturbance torque:

$$\tau = \tau_e + \tau_d . \tag{2 - 2}$$

2.1.3 Jacobian Transpose and Jacobian Inverse Control

One position control concept which is well-suited to the task-space formulation is Jacobian Transpose Control. This scheme simulates a position controller which applies a force in task-space at the manipulator endpoint in response to the task-space position error. In the static or quasi-static case, it can be shown that the torque τ_e at the motors required to produce a force F at the end-effector is given by

$$\tau_e = \mathbf{J}^{\mathrm{T}} F , \tag{2 - 3}$$

where **J** is the manipulator Jacobian, defined by the kinematic relationship between the joint velocities and endpoint velocities (Craig, 1989). Thus, a position control algorithm is implemented by computing the desired control force and simply transforming it to joint torques via Equation (2 - 3). This scheme can be used for both position and endpoint force control, because it directly relates endpoint forces to joint torques.

A similar position-control algorithm is Jacobian Inverse Control. This scheme converts the task-space error into joint-space errors by assuming that these errors are small enough such that the relationship

$$\Delta x \approx \mathbf{J}\ \Delta\theta \tag{2 - 4}$$

applies. The controller then applies a torque in response to this joint-space error.

Jacobian Inverse and Jacobian Transpose control behave in a very similar manner when the manipulator is far from its singular configurations. They are both very simple methods which require no knowledge of the manipulator inertial parameters. Although neither scheme can guarantee trajectory tracking, both can be shown to be stable under PD position control (Craig, 1989). In addition, both can, in theory, achieve arbitrarily small steady-state errors to a constant reference position under PD position control.

2.1.4 The Implemented Control Algorithms

Two hybrid position/force control schemes have been implemented. One scheme performs Jacobian Transpose control in the position domain; the other performs Jacobian Inverse control in the position domain. Both schemes perform Jacobian Transpose control in the force domain. The presented schemes are based on the algorithms presented in (Raibert and Craig, 1981).

Figure 2-1 shows the implemented Jacobian Transpose hybrid position/force control algorithm. The position domain is represented by the projection matrix **P** and the force domain is represented by the complementary projection matrix **F**. Note that these two matrices split the vector space into orthogonal subspaces. Thus for any vector v, $\mathbf{P}v$ is the position component, $\mathbf{F}v$ is the force component, and

$$\mathbf{P}v+\mathbf{F}v=v. \qquad (2-5)$$

The joint locations q are measured and converted via the kinematic equations (Kin) to endpoint position x. The error in position is then projected into the position domain, and controlled via PID control. Note that the gains $\mathbf{K}_{px}$, $\mathbf{K}_{dx}$ and $\mathbf{K}_{ix}$ represent stiffness, damping, and integration gain in the Cartesian task-space.

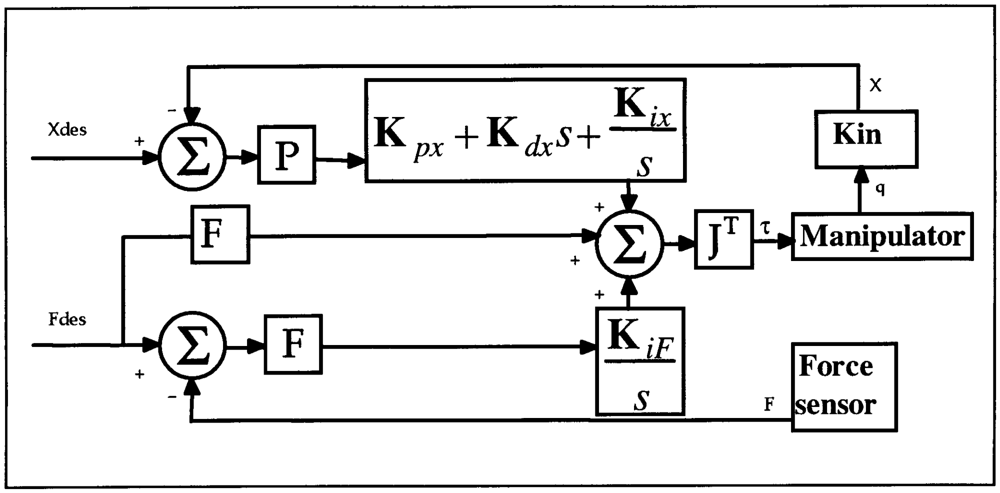
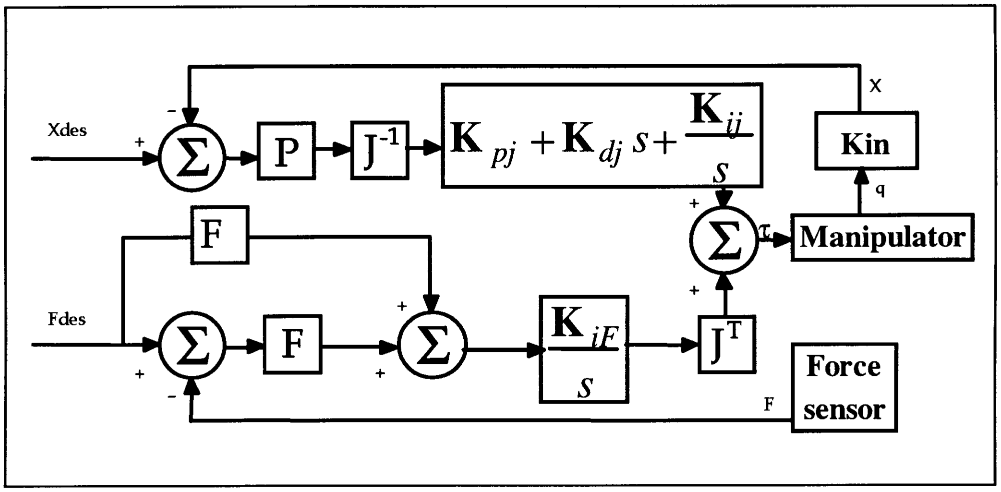

Xdes
Fdes
+
-
P
F
$\mathbf{K}_{px} + \mathbf{K}_{dx}s + \frac{\mathbf{K}_{ix}}{s}$
$\frac{\mathbf{K}_{iF}}{s}$
J^T
τ
x
Kin
q
Manipulator
Force sensor
F

Figure 2-1: Hybrid P/F Control, J Transpose

The desired endpoint force *Fdes* is projected into the force domain and fed-forward to the control force. This provides the control effort which nominally results in the desired endpoint force. Explicit force control is also implemented by measuring the endpoint force with a wrist force/torque sensor. The error in force is projected into the force domain and controlled via an integrator with gain $\mathbf{K}_{iF}$. Note that PD force control, which has been suggested (Eppinger and Seering, 1987), cannot realistically be implemented. The incorporation of a force sensor signal directly in the control loop (P control) violates causality; the signal being measured is of the same order as the system itself, and proportional control would therefore introduce an algebraic loop. Algebraic loops indicate that unmodelled dynamics can no longer be neglected. For instance, the time delay involved in digital sampling would become significant. Consequently, the system does not behave as expected. Differentiating the force sensor signal (D control) is even less reasonable; causality is violated further, and the force sensor signal is likely to be noisy. Integration of the force signal has been recommended since it filters the signal which both decreases the signal order by one and tends to reject high-frequency disturbances (Volpe and Khosla, 1992).

The position domain and force domain control forces are summed and then multiplied by the Jacobian transpose matrix to yield the joint control torque. It is important to note that the P and F selection matrices are likely to change during a contact control task. For instance, before contact, P is the identity matrix (full rank) and F is the zero matrix (no rank). After point contact without friction, the P matrix loses one rank and the F matrix gains one rank. In order to alleviate some problems associated with a controller that switches between two modes, the force and position control integrators are located past the P and F selection matrices. Because the integrators are located as shown, a discontinuous change in P and F does not translate to a large discontinuity in control input.



Xdes
Fdes
P
F
J^{-1}
$\mathbf{K}_{pj} + \mathbf{K}_{dj} s + \frac{\mathbf{K}_{ij}}{s}$
$\frac{\mathbf{K}_{iF}}{s}$
J^T
Kin
Manipulator
Force sensor
x
q
τ
F
+
-

Figure 2-2: Hybrid P/F Control, J Inverse

Figure 2-2 depicts the control algorithm for Jacobian Inverse position control. This scheme is similar to that shown in Figure 2-1, but now $\mathbf{K}_{pj}$, $\mathbf{K}_{ij}$, and $\mathbf{K}_{dj}$ represent *joint-space* stiffness, integration, and damping gains. The PID position controller operates on joint error and produces a control torque. The force domain is unchanged from Figure 2-1.

For both of these schemes, it is important that the contact transition be handled carefully. As was mentioned, the location of the integrators in the block diagram helps to minimize control discontinuity during contact detection. Another way to deal with the problem of control effort discontinuity is to apply an offset torque which is constant except during mode switching. Whenever the mode switches, this offset torque changes so that the control action required after the transition is the same as that before the transition. This offset torque can be seen as re-initializing the position and force integrators during mode switching.

In addition to guaranteeing stable switching, it is important to minimize the approach speed so that there is no impact damage to the crucible (Youcef-Toumi and Gutz, 1989, also see Appendix C). It is possible that the manipulator loses contact, and this condition must be handled gracefully by prohibiting the manipulator to achieve high speeds. The integral force controller does not handle this situation well. The contact force remains zero and the manipulator rapidly re-establishes contact, causing a contact force impulse, which tends to send the manipulator retreating from the surface faster than before. This clearly unstable behavior is undesirable even for relatively durable surfaces. Therefore, if the manipulator breaks contact, (which is detected by a dropping of the contact force to nearly zero,) the integral force controller shown in Figure 2-1 and Figure 2-2 is replaced by a velocity damping term and an integral positioning term which brings the manipulator back into contact with the surface slowly.

2.2 Simulation of Control Algorithms

In order to demonstrate the effectiveness of these algorithms, two-dimensional simulations were performed via Simulink. The system model, which resembles the laboratory demonstration system, consists of a SCARA manipulator arm in contact with a stiff cylindrical crucible. This system is shown in Figure 2-3(a). The manipulator consists of two rotational planar links in the *x-y* plane, followed by a vertical prismatic joint which can also rotate about its axis.

The dynamics of the prismatic joint are decoupled from that of the rotational links, so a top-view simulation captures the important dynamic behavior. It is assumed that the prismatic link could be lifted to clear the crucible if necessary. The manipulator can therefore be viewed as a two-link planar robot arm, shown in Figure 2-3(b).

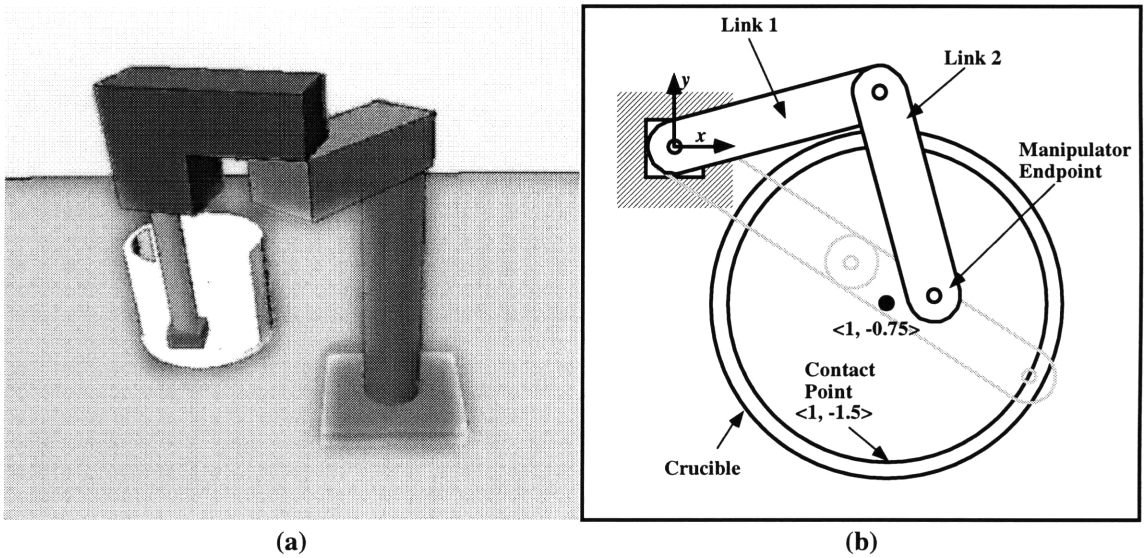


Figure 2-3: Simulation Model

In the simulation, the first joint is fixed at the origin of the *x-y* plane and each link is 1 unit in length. The crucible is represented by a circle centered at the point <1, -0.75> with a radius of 0.75 units. Note that in this system it is possible for the manipulator endpoint to the entire interior of the crucible; the farthest point in the crucible from the origin is <1.6, -1.2>, which is exactly 2 units away. In the simulations, the nominal contact point is at <1, -1.5> as shown; this point is sufficiently far from a singular configuration to perform the presented control algorithms.

A general block diagram of these simulations is shown in Figure 2-4. The manipulator dynamics are that of a two-link planar robot arm including some viscous and breakaway joint friction. The manipulator dynamics block produces endpoint coordinates $<x,y>$ as a function of input control torque and endpoint contact force, F. The contact model is that of a stiff circular surface which produces a normal contact force toward the center of the crucible. The contact forces and endpoint position are sensed and fed to the hybrid controller algorithm shown in Figure 2-2. This controller generates a control torque intended to effect tracking of the desired position and force trajectories, x_{des} and F_{des}. The hybrid controller assumes that the position and force domain directions are known throughout the entire trial, and that switching of the projection matrices occurs as soon as contact is detected (when the contact force is greater than zero).

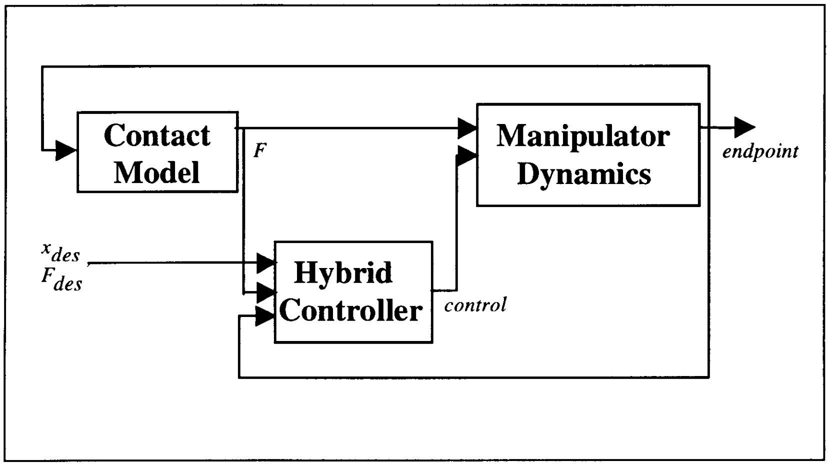


Figure 2-4: Block Diagram for Hybrid Control Simulation

In the simulated experiments, the end-effector starts at position <1,0> and moves under pure position control (**P=I**, **F**=0) with a speed of 0.33 units per second in the $-y$ direction until contact is made with the crucible. Once contact is detected, the system switches selection matrices so that there is position control in the *x* direction and force control in the *y* direction.

Figure 2-5 shows the results of an experiment in which the end-effector is held fixed after contact and the contact force is commanded to be regulated at 15 Newtons. Contact is detected at 4.5 seconds and results in a large initial force spike and oscillation. There is also a momentary positioning error. The contact force is then quickly regulated to the desired force of 15 Newtons.

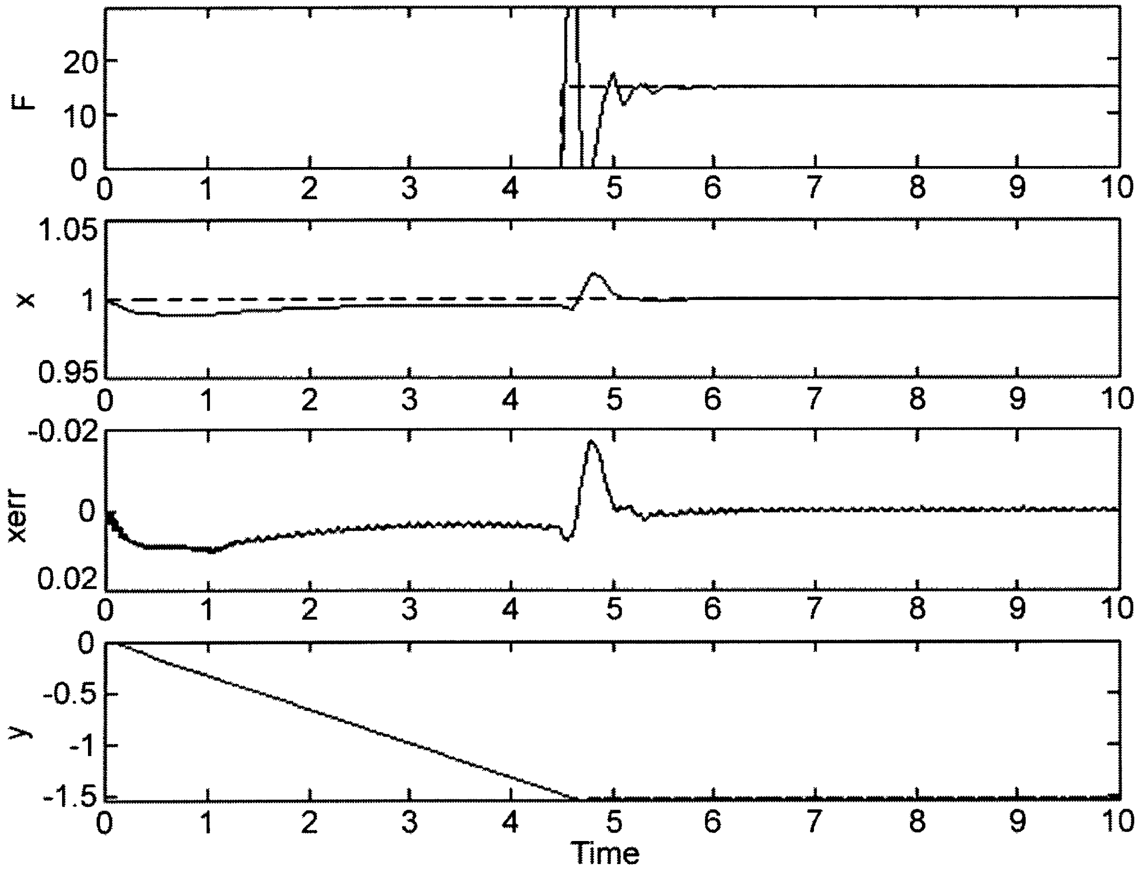


Figure 2-5: Simulation Results, Endpoint Held Fixed After Contact

Figure 2-6 shows the results of an experiment in which the end-effector is commanded to slide along the crucible surface in the *x* direction after contact. The desired position is a sinusoid of amplitude 0.02 units and period 6.28 seconds. Note that the amplitude is small enough that the *y* position of the trajectory is essentially constant. Again, the contact force is to be regulated at 15 Newtons. Contact is detected at 4.5 seconds, and again results in a large initial force spike and momentary positioning error. The contact force is then regulated relatively quickly to nearly 15 Newtons, although in this case there are small deviations of approximately 0.5 units in amplitude. There is a lag in the position tracking, which creates a periodic error of about 0.003 units in amplitude.

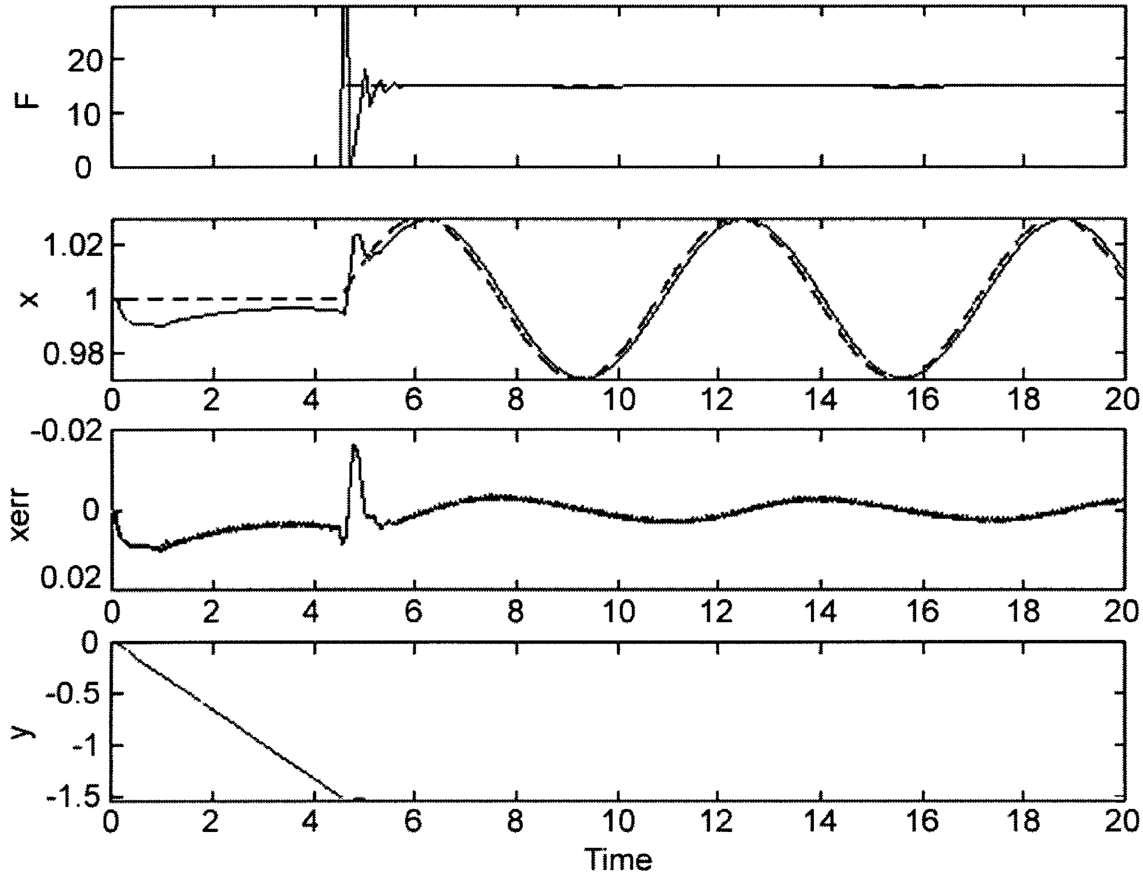


Figure 2-6: Simulation Results, Sinusoidal Motion

2.3 Summary

The hybrid position/force control algorithm presented in this chapter is the kernel of the control system for the RACS laboratory demonstration system. The simulation of this simple algorithm shows acceptable results; the force errors are less than 5 percent, and therefore would not cause damage to the crucible while the contact point slides. It is hypothesized that these algorithms can be used to safely place nuggets in the crucible.

Chapter 3

Hybrid Position/Force Control Experiments

3.1 Experiments on a Puma 250 Manipulator

The Jacobian Transpose algorithm presented in Chapter 2 was implemented and tested on a Puma 250 articulated robot manipulator. These tests were performed to verify the algorithm's suitability for use with the RACS system.

3.1.1 Apparatus

Figure 3-1 depicts the Puma 250 experimental setup, located at the MIT Field and Space Robotics Laboratory. This system was developed by past members of the lab, and is documented in (Idris, 1992). The control code software is written in C and compiled on a Sun 3/80 workstation. This code runs on a Heurikon HK68 embedded processor, which communicates to the Sun workstation via a piggy-backed V30XE extended ethernet card. The control code issues torque commands via VME Bus to the Programmable Multi-Axis Controller (PMAC). The PMAC can either accept torque commands from the Heurikon processor, or generate its own torque commands (the PMAC itself can implement PID position control). The PMAC then sends current signals to the Unimation power box, which powers the motors of the Puma 250 manipulator. Encoders in the joints of the manipulator send joint position information back through the power box to the PMAC. Force information from the load cell is passed to an A/D conversion board made by Data Translation, Inc. The A/D board and PMAC convey the force and position information to the control code running on the Heurikon processor via the VME Bus. Experimental data is collected and saved to the hard drive on the Sun 3/80.

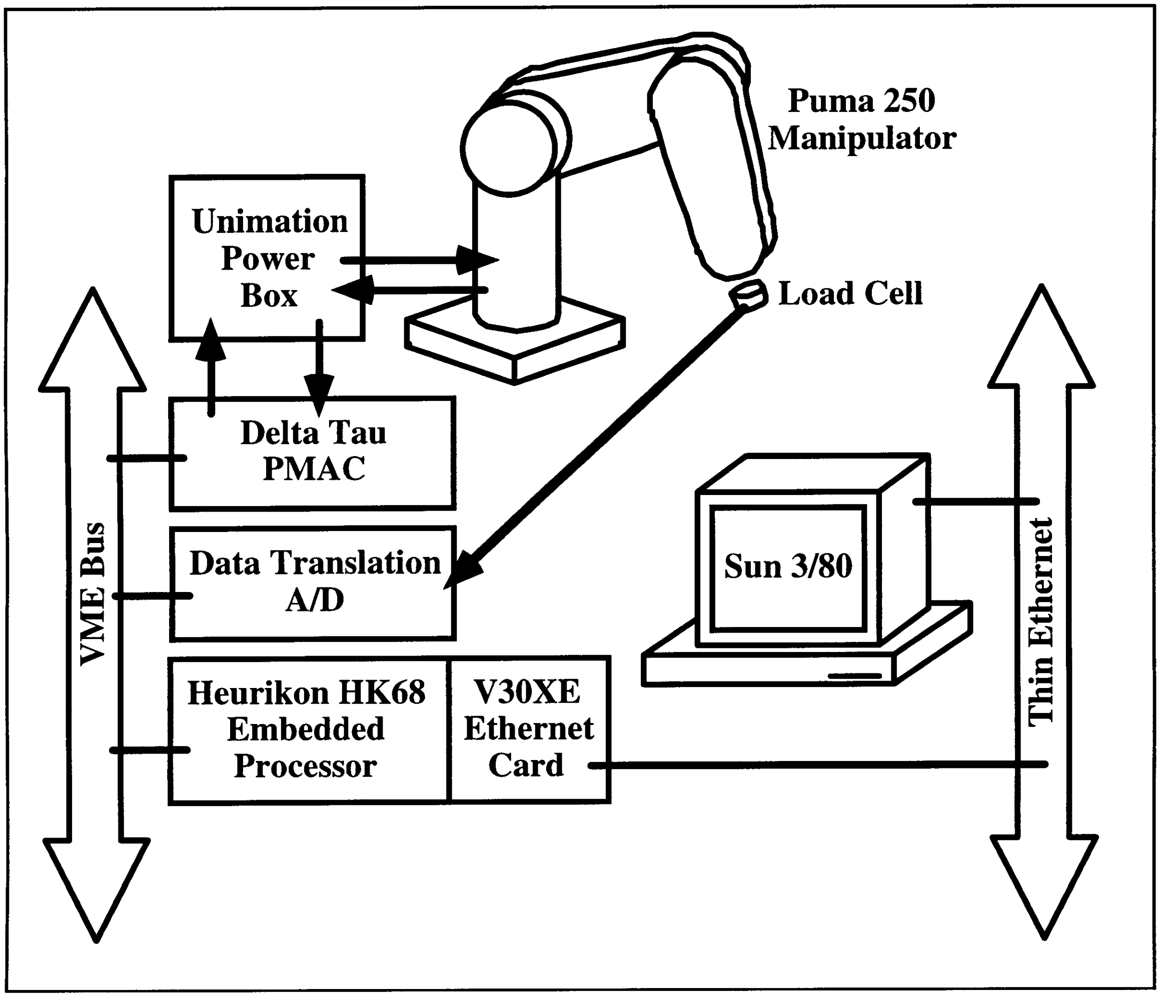


Figure 3-1: Experimental Apparatus for Puma 250

Figure 3-2 shows the structure of the Puma 250 manipulator. The Puma 250 is nominally a six degree-of-freedom vertically articulated robot arm, although for the following experiments the last three (wrist) joints are fixed. The load cell is mounted to the endpoint of the robot; this sensor can measure forces applied along the axis of the last link. The manipulator kinematics are described in Appendix A.

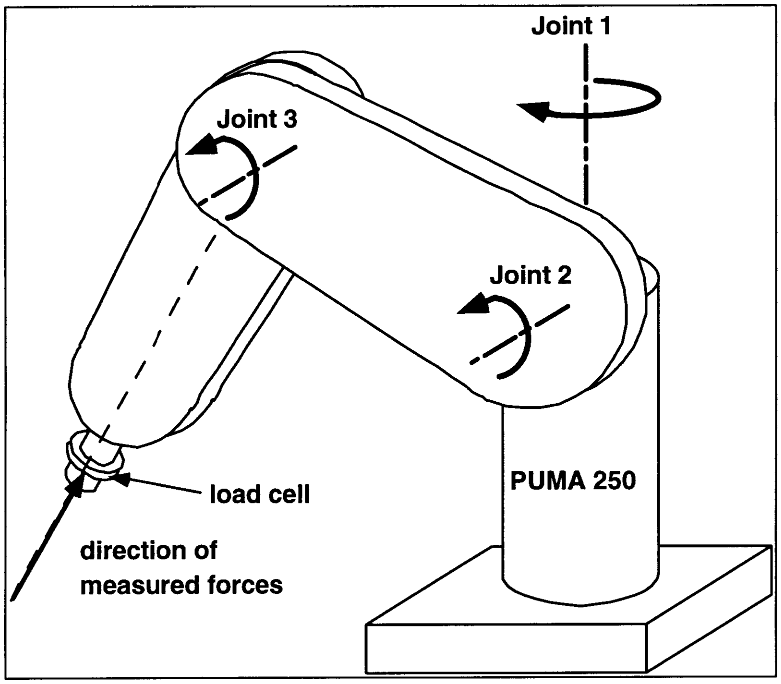


Figure 3-2: The Puma 250 Manipulator

3.1.2 Experimental Procedure – Roller and Nugget Tests

The experiments performed on the Puma 250 are divided into two categories: roller tests and nugget tests.

The roller test, illustrated in Figure 3-3, involves the regulation of contact force in the *z* direction while rolling along a wooden horizontal surface (the *x*-*y* plane). The end-effector for these experiments is a wooden cylindrical roller mounted onto the load cell. At the start of each trial, the roller is held several centimeters above the horizontal surface. Since there is no contact with the environment, every task-space direction lies in the position control domain (**P**=**I**), and there is no force control domain (**F**=0). The system is commanded to slowly move the roller vertically downward toward the table until contact is made. The system detects contact when the load cell measures a contact force above an empirically-determined threshold. Once contact is detected, the position control domain is restricted to the *x* and *y* directions, and the force control domain expands to include the *z* direction. The roller is commanded to either be held fixed, or to roll back and forth along the surface while maintaining a desired vertical force.

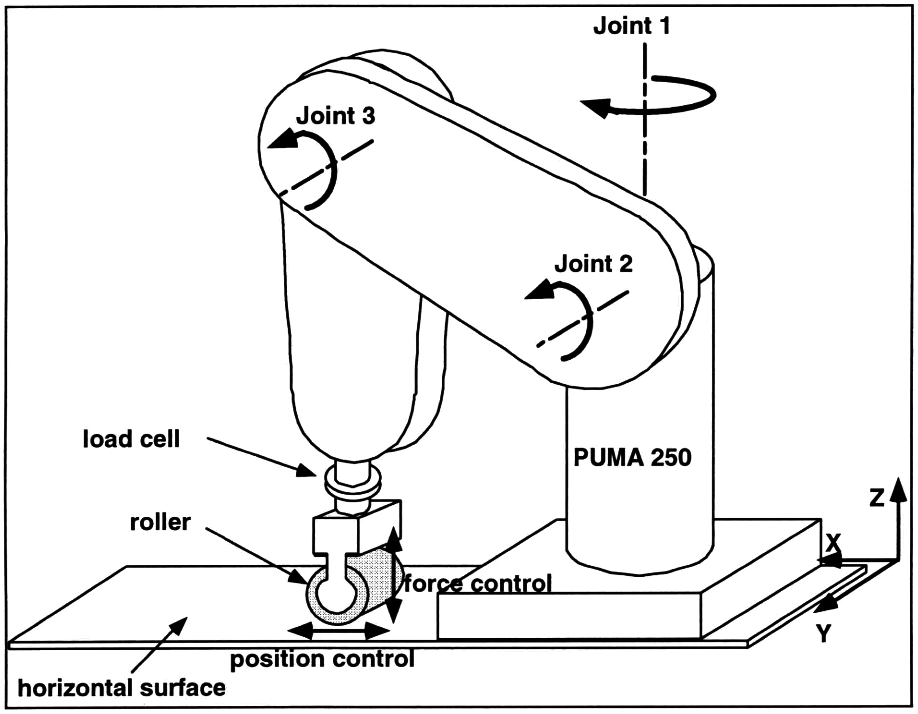


Figure 3-3: Puma Roller Tests

The nugget test, depicted in Figure 3-4, is similar to the roller test but simulates the nature of the wall-building process. In this case the end-effector firmly grips a silicon nugget and the contact surface is a glass wall. The nugget tests involve the regulation of contact force in the *x* direction while sliding along a vertical surface (the *y*-*z* plane). At the start of each trial, the nugget is held several centimeters in front of the vertical glass surface. The system is commanded to slowly move the nugget toward the wall until contact is made. Again, the nugget is commanded to either be held fixed, or to slide back and forth along the surface while maintaining a desired force.

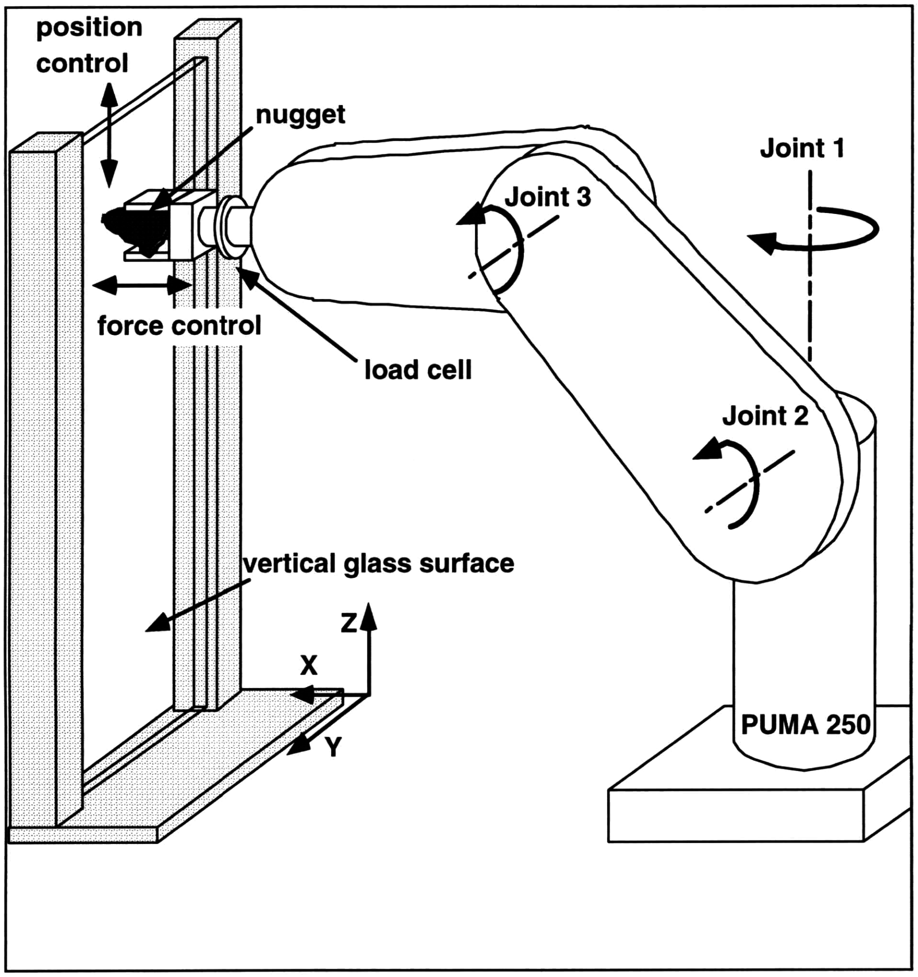


Figure 3-4: Puma Nugget Tests

3.1.3 Experimental Results

Figure 3-5 shows the typical results of a roller experiment in which the end-effector is held fixed. A desired contact force of 5 Newtons is commanded once contact is made. The endpoint begins approximately five centimeters above the surface and moves downward. Contact is detected at approximately 1.5 seconds. It can be seen that the contact force quickly converges to the desired force.

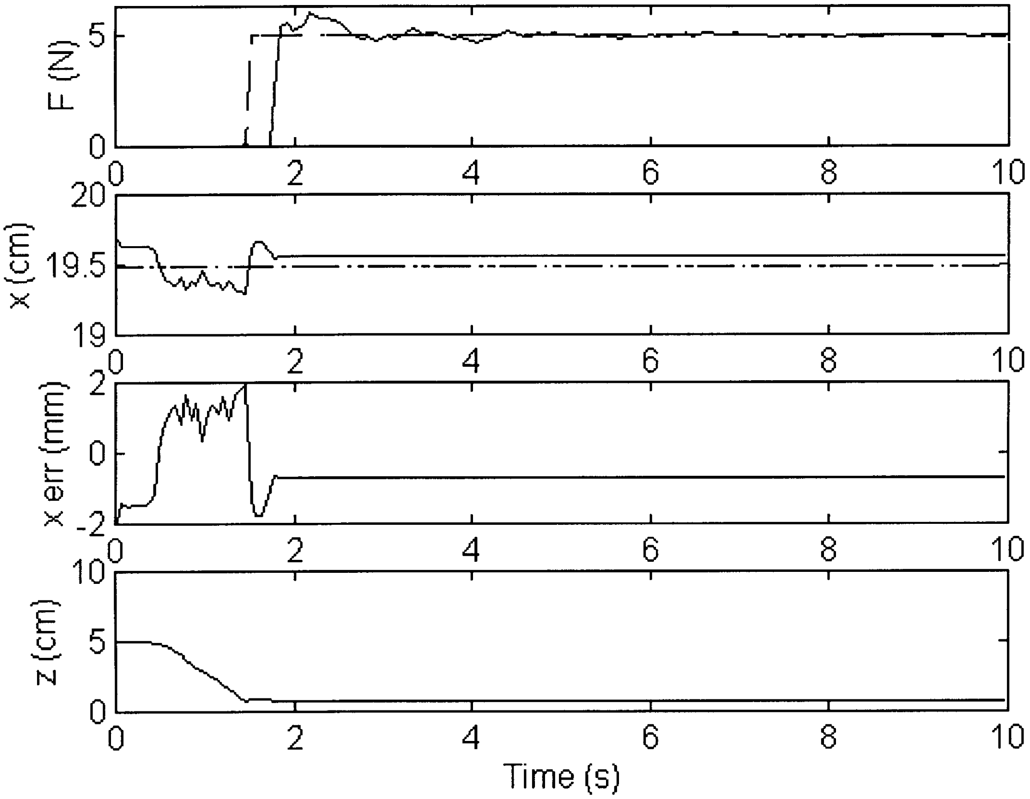


Figure 3-5: Puma Roller Test Results, Roller Held Fixed

Figure 3-6 shows the typical results to a roller experiment in which the end-effector is commanded to roll along the surface once contact is made. The desired position trajectory is a one-dimensional sinusoid with a 4-cm amplitude and a 0.25-Hz frequency. The endpoint again begins approximately five centimeters above the surface and moves downward. Contact is detected at approximately 1.5 seconds. The desired contact force is 5 Newtons. The system maintains contact with the surface; the measured force is always nonzero after initial contact. The position controller shows rather poor performance; there is significant lag, especially when the roller direction is commanded to change. The plateaus in the graph indicate that the roller sticks at the extrema of the path. Thus there is a position error of several millimeters. In this trial, the force regulation is moderately successful; the average (zero-frequency) force is maintained at 5 Newtons. However, there is also a disturbance which causes a deviation of nearly 5 Newtons from the desired force.

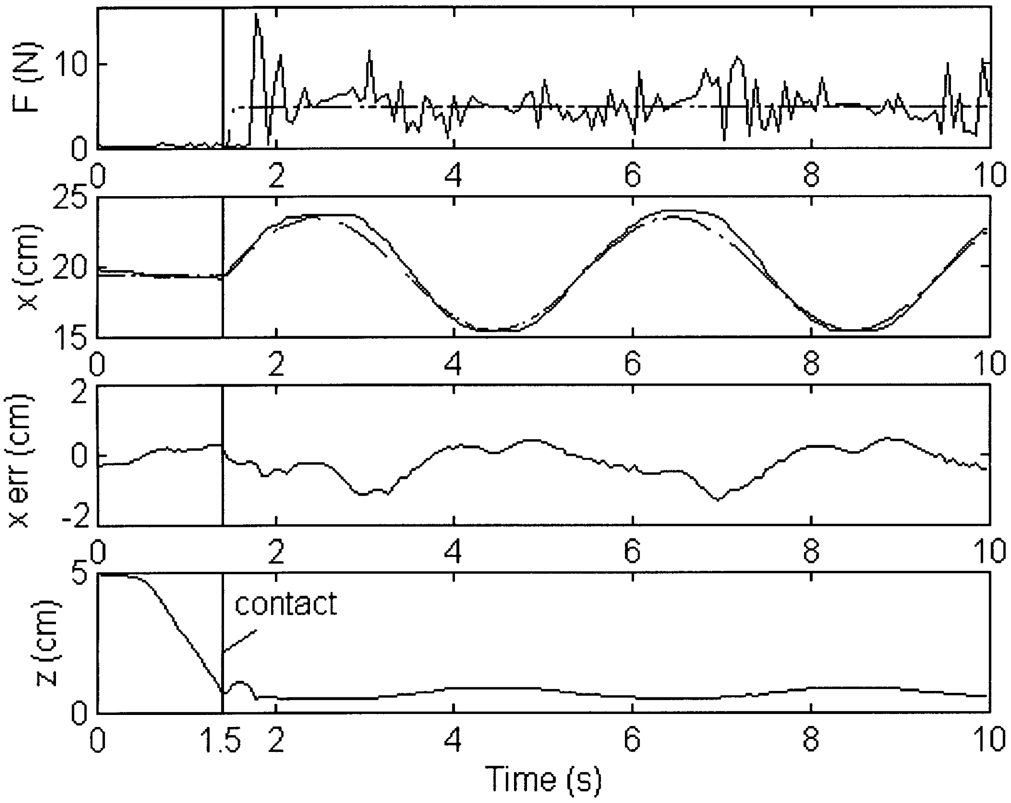


Figure 3-6: Puma Roller Test Results, Roller in Motion

Figure 3-7 shows the typical results of a nugget experiment in which the end-effector is held fixed once contact is made. The endpoint begins approximately three centimeters in front of the surface and moves forward. Contact is made and detected at approximately 0.5 seconds. The desired contact force is again 5 Newtons. It can be seen that, as in the roller case, the contact force converges on the desired force after an initial force spike.

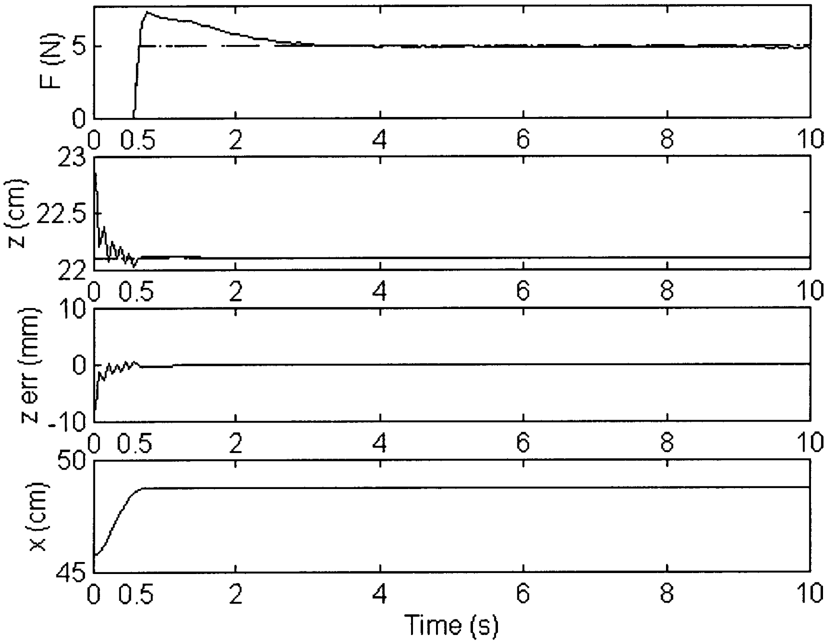


Figure 3-7: Puma Nugget Test Results, Nugget Held Fixed

Figure 3-8 shows the typical results of a nugget experiment in which the end-effector is commanded to slide the nugget along the glass surface with a desired contact force of 5 Newtons once contact is made. The desired position trajectory is a one-dimensional sinusoid with a 4-cm amplitude and a slow, 0.1-Hz frequency. The endpoint again begins approximately three centimeters from the wall and moves toward it.

Contact is detected at approximately 1 second. The behavior is somewhat degraded from that of the roller test. The system does maintain contact with the surface, although the contact force does momentarily drop to nearly zero about 7 seconds into the trial. Again, the position controller shows rather poor performance; there is significant lag, and plateaus in the graph indicate sticking. Thus there is position error of up to two centimeters. In this case, the force regulation is somewhat successful; the average force is indeed maintained at 5 Newtons. However, there is also an occasional disturbance force which causes a deviation anywhere from +7 Newtons to –5 Newtons. This negative deviation is nearly enough to cause contact to be lost.

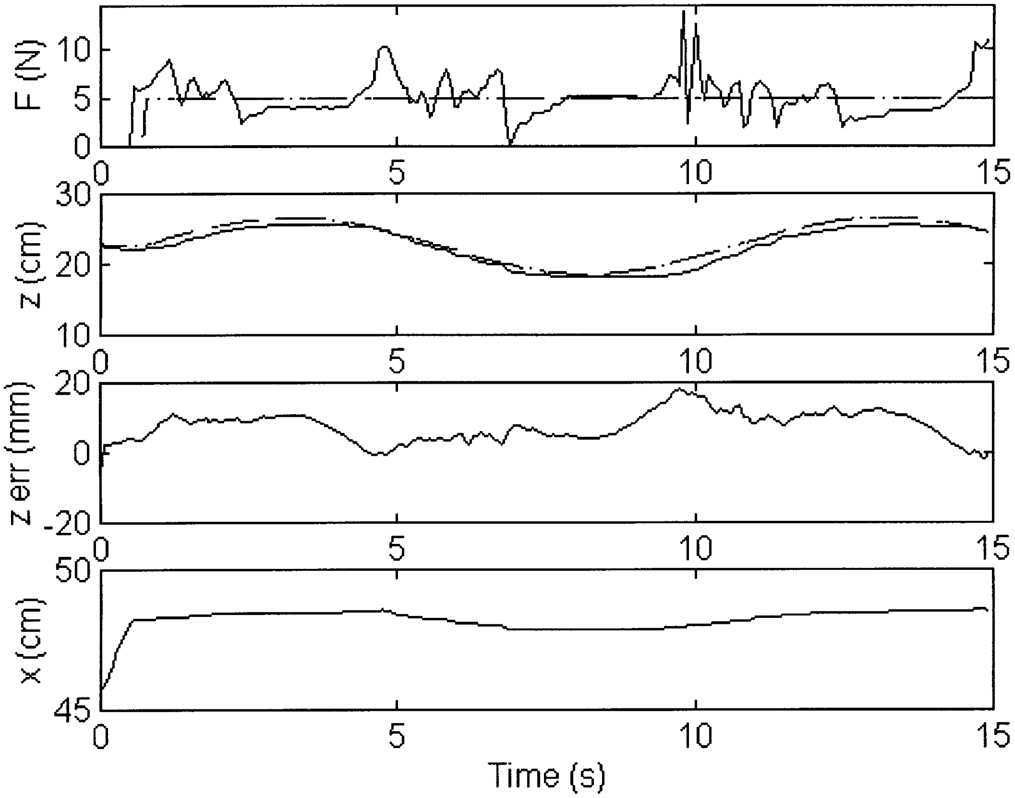


Figure 3-8: Puma Nugget Test Results, Nugget in Motion

3.1.4 Summary of Puma Experimental Results

This testing demonstrates that the system can accurately maintain a desired force while the end-effector is held fixed. Performance is degraded when position control is not merely used as a regulator. Additional experiments were performed to isolate the cause of this degradation.

Experiments were performed in which the surfaces were more compliant and damped; however, the force spikes were still present. It was observed that in all experiments the manipulator would stick and slip during pure position motions, and that the actual velocity of the end-effector would briefly have a significant component in a direction other than that desired.

These results indicate that the performance degradation is due largely to the presence of high amounts of backlash and nonlinear joint friction in the Puma 250, which causes excessive coupling in the motion of the endpoint. Thus the manipulator tends to exert forces against the surface while it moves along the surface. It is expected that the AdeptOne manipulator in the RACS laboratory demonstration system exhibits much less friction and backlash.

3.2 Experiments on the AdeptOne Manipulator

The Jacobian Inverse hybrid position/force control algorithm presented in Chapter 2 was also implemented and tested on the AdeptOne manipulator for the laboratory demonstration system. These experiments were performed to further verify the suitability of hybrid position/force control for the RACS wall and crown building modes.

3.2.1 Apparatus

Figure 3-9 depicts the AdeptOne system apparatus. Torque commands are generated by an algorithm running on the control computer. These are sent to the joint controller cards in the controller unit. These cards convert the torque commands into current commands, which are specifically computed to compensate for the AdeptOne motor nonlinearities (Goldenberg *et al.*, 1994). These command signals are fed to the power amplifiers, which in turn provide current to the robot's motors. Encoder signals reflecting joint positions are sent back to the joint controller cards, which send this information back to the control computer. Measurements from the wrist force/torque sensor are also supplied to the control computer.

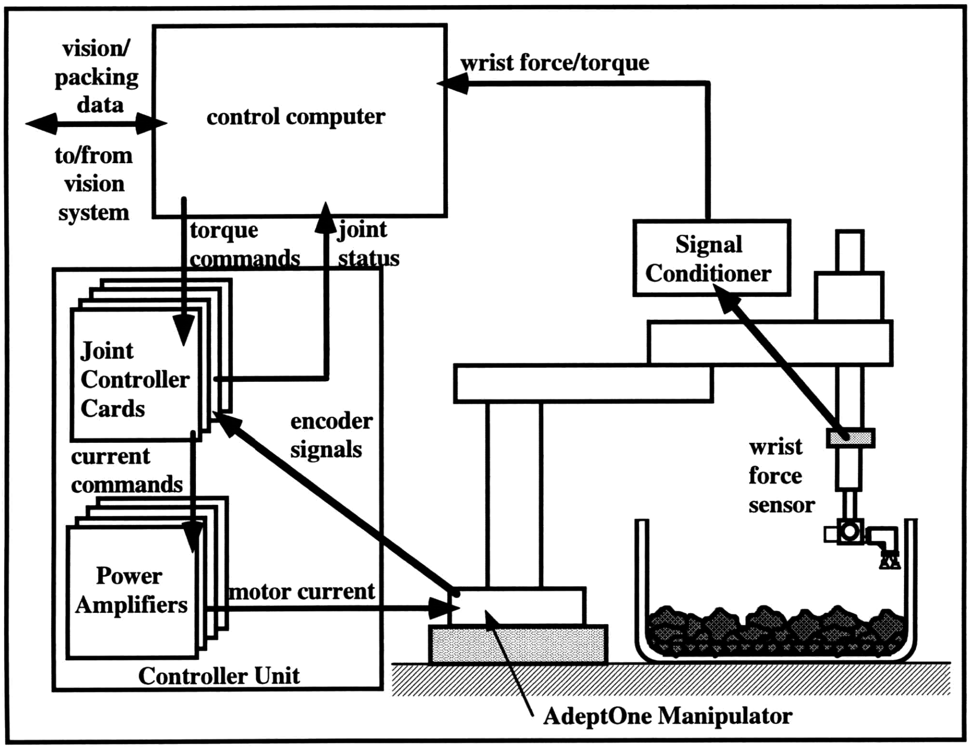


Figure 3-9: Experimental Apparatus for the AdeptOne System

Figure 3-10 shows the structure of the AdeptOne manipulator. It is a four degree-of-freedom SCARA robot. The three rotational joints (1, 2, and 4) rotate their associated links in the *x-y* plane, and the prismatic joint (joint 3) translates linearly in the *z* direction. See Appendix A for the manipulator kinematics.

3.2.2 Experimental Procedure – Roller and Nugget Tests

As with the Puma 250, roller experiments were performed on the AdeptOne system. The end-effector for these experiments is a Delrin cylindrical roller mounted onto the six-axis wrist force/torque sensor. The horizontal surface is a thick aluminum plate. Refer to section 3.1.2 above for a detailed description of the experimental procedure.

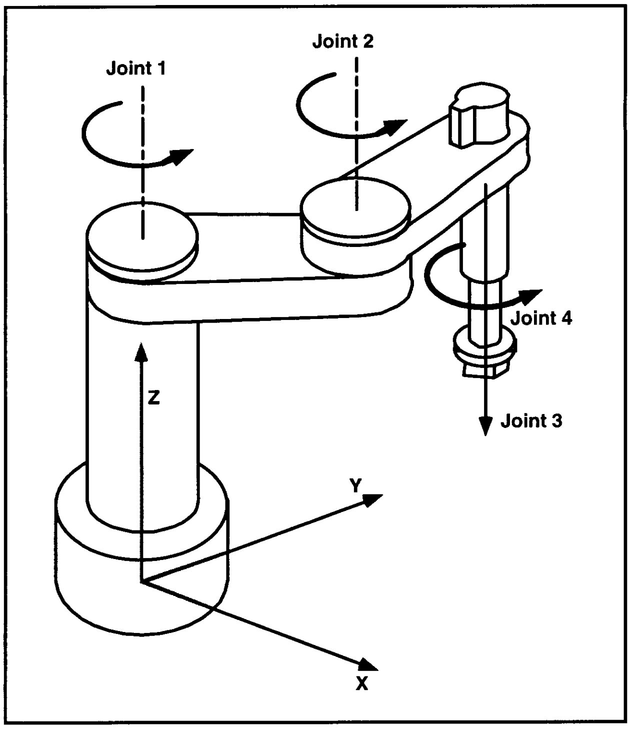


Figure 3-10: The AdeptOne Manipulator

Nugget experiments were also performed against a vertical wall, as described in section 3.1.2 above. For these experiments, a stiff metal surface replaced the glass wall. Since the AdeptOne manipulator is much more powerful than the Puma 250, it poses a greater risk of damage to the environment. The metal surface provides valid information about the important behavior of the control algorithm, without constant risk of breaking the glass, with its associated hazards and experimental delays.

3.2.3 Experimental Results

Figure 3-11 shows the typical results of a roller experiment in which the end-effector is held fixed after contact. The endpoint begins approximately seven centimeters above the surface and moves downward. Contact is detected at approximately 6 seconds. The desired contact force is 20 Newtons. It can be seen that after an initial contact force spike, the contact force quickly converges on the desired force.

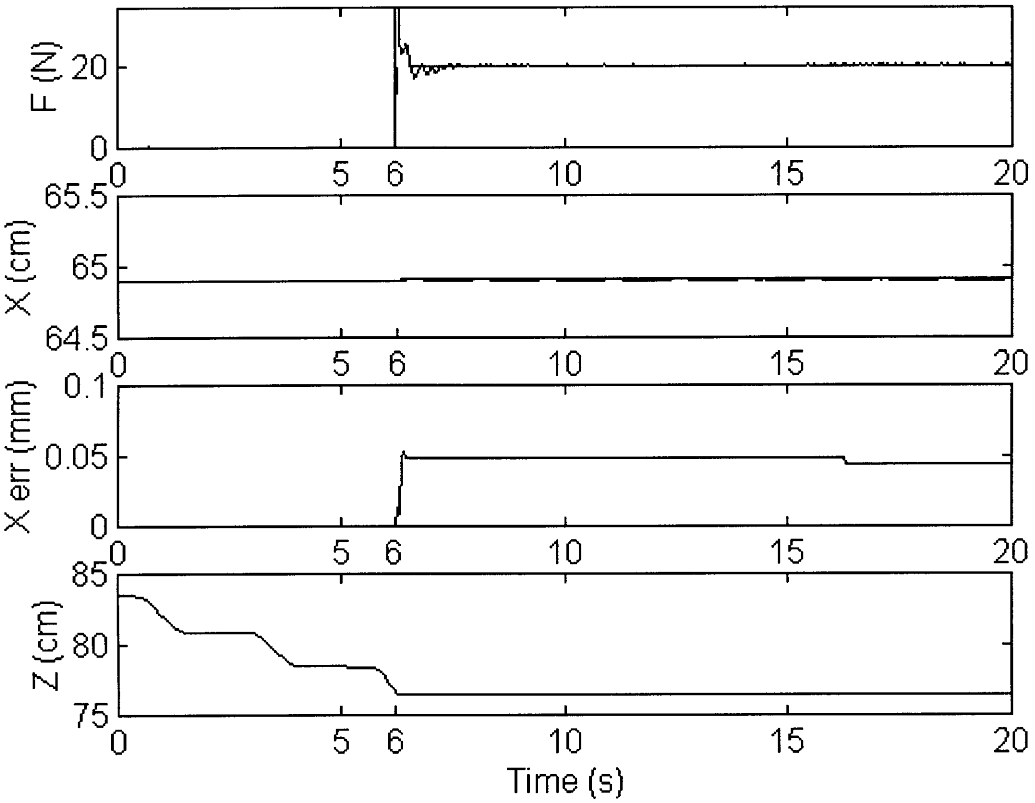


Figure 3-11: AdeptOne Roller Test Results, Roller Held Fixed

Figure 3-12 shows the typical results of a roller experiment in which the end-effector is commanded to roll along the surface once contact is established. The desired position trajectory is a one-dimensional sinusoid with a 3-cm amplitude and a 0.25-Hz frequency. The endpoint begins approximately one centimeter above the surface and moves downward. Contact is detected at approximately 1.8 seconds. The desired contact force is 20 Newtons.

As with the Puma 250, the system maintains contact with the surface; the contact force is always nonzero after initial contact. The position controller shows improved performance over the Puma 250, but there is still significant lag when the roller direction is commanded to change. The plateaus persist, and there is a maximum position error of about 5 millimeters. In this case, the force regulation is moderately successful; the average (zero-frequency) force is maintained at 20 Newtons. However, there is also a disturbance which causes a momentary deviation of up to 15 Newtons from the desired force.

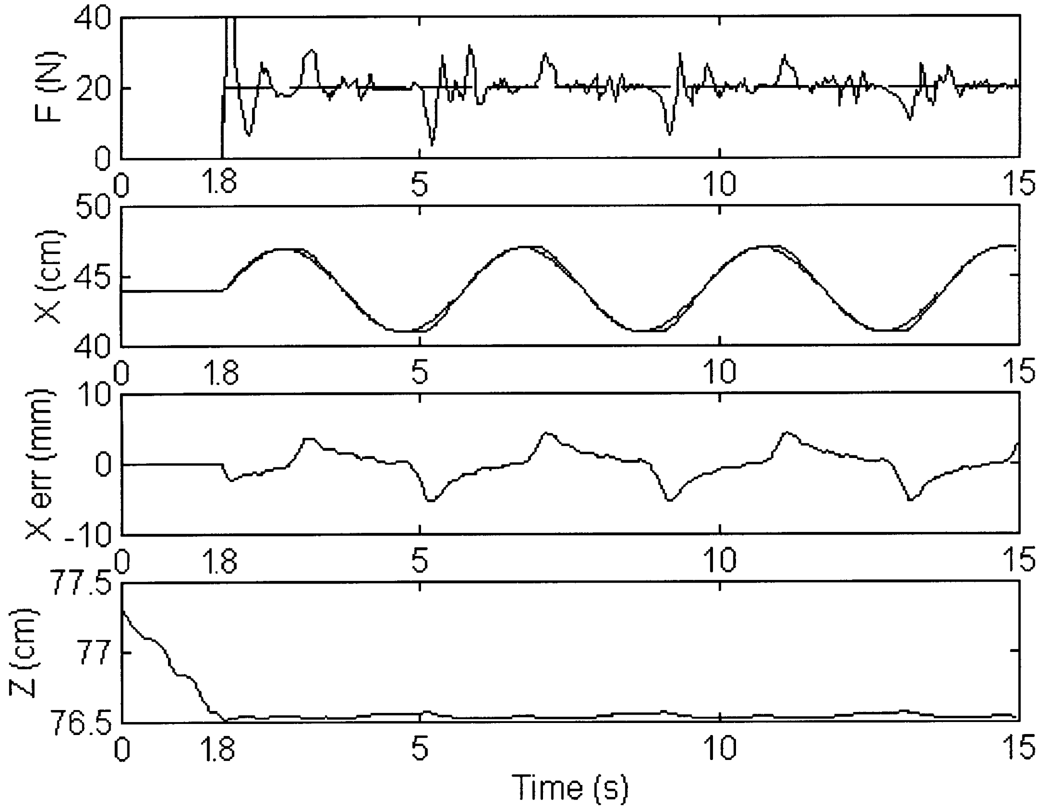


Figure 3-12: AdeptOne Roller Test Results, Roller in Motion

The nugget tests showed similar results. Figure 3-13 depicts the result of a test in which the nugget is held fixed against the vertical wall. Once contact is detected, the force profile oscillates for several seconds and settles on the chosen value of 5 Newtons, which was experimentally determined not to cause scratching of the glass. Deviations of approximately 0.5 Newtons are purely the result of sensor noise. When the sensor is stationary and not in contact with any object, tests show that the force signal tends to deviate from zero by up to 0.5 Newtons.

The *z*-positioning error was very small in this example (less than 0.02 millimeters) because the system was acting as a regulator in this direction, and the dynamics of the prismatic joint (*z*-direction) are decoupled from that of the rest of the manipulator system.

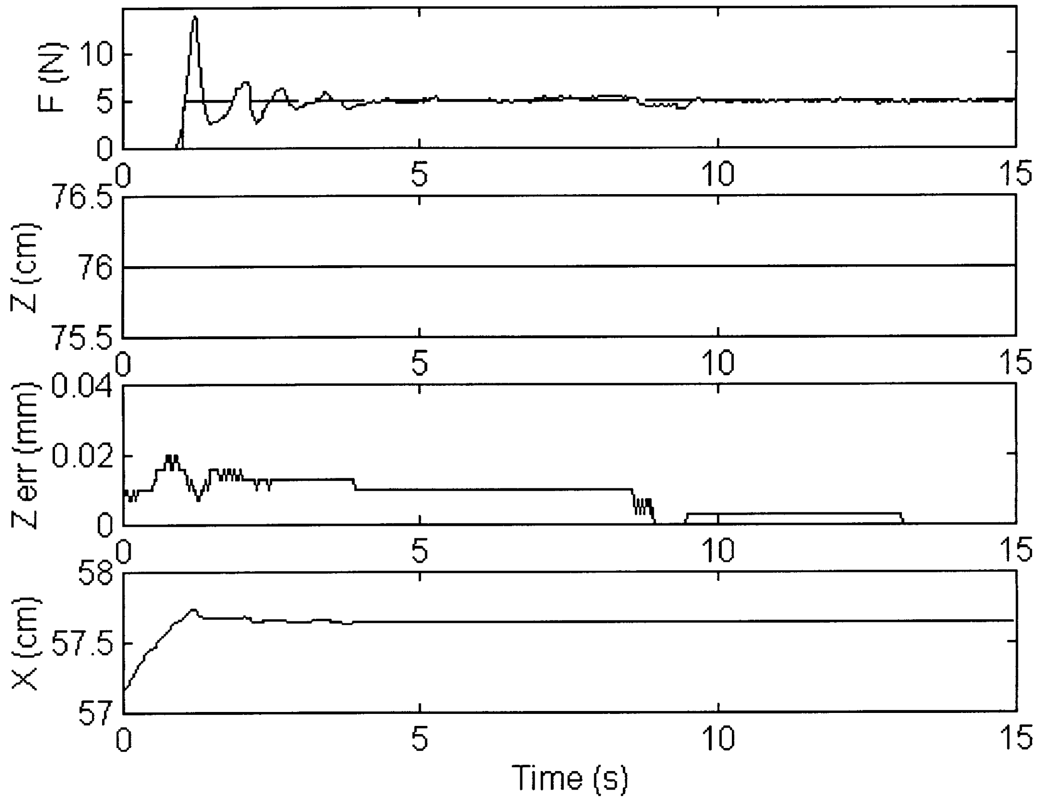


Figure 3-13: AdeptOne Nugget Results, Nugget Held Fixed

Figure 3-14 shows the results of a trial in which the nugget is commanded to slide in the vertical (*z*) direction while in contact with the wall. The desired contact force is 5 Newtons. The desired position trajectory is a sinusoid of amplitude 1 cm and period 10 seconds. Contact is detected at approximately one second. The force controller is somewhat able to maintain the desired force, with deviations ranging from +1 to –2.5 Newtons. The position controller also shows moderate performance, maintaining the vertical position error to within ±2 mm. This performance is certainly improved over that of the Puma 250.

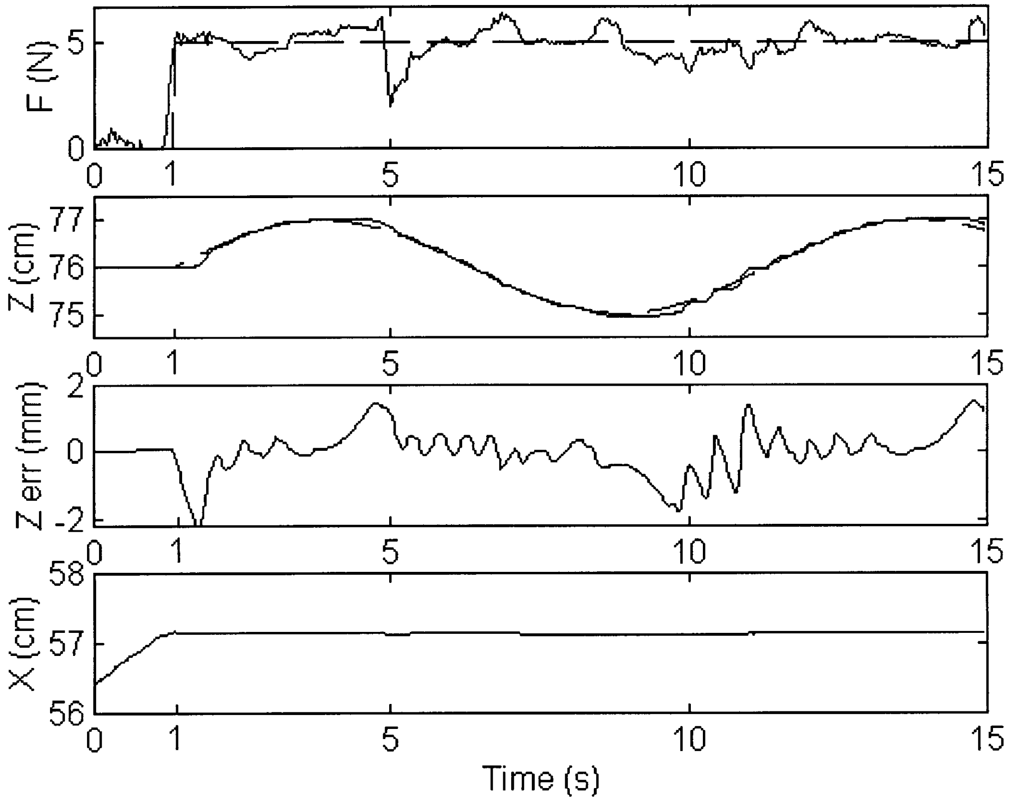


Figure 3-14: Adept Nugget Test results, Nugget in Motion

3.2.4 Summary of AdeptOne Experiments

This testing demonstrates that, like the Puma 250, the AdeptOne system can maintain a desired force while the end-effector is held fixed. Performance is still degraded when position control is not merely used as a regulator. Experiments were performed to determine the cause of this degradation.

Further roller tests were conducted in which the steel plate was covered with a thin layer of rubber to prevent force spikes due to the nature of contact. The results were qualitatively similar to those of Figure 3-12; this suggests that the primary cause of disturbance is not contact surface roughness.

The experimental results suggest that several factors influence this degradation: dynamic coupling in the control algorithm, force sensor cross-talk, and joint friction.

The deviations in the force profile seen in the nugget experiments may be caused by the approximations made in the control algorithm, see section 2.1.2. Under pure position motions of the manipulator, the actual velocity was seen to deviate in direction from the desired. Although the position controller and the force controller are supposed to act in orthogonal directions, there is a certain degree of dynamic coupling. Note that this was not seen in the roller experiments, where the force domain consisted only of the vertical z direction. The motion of the prismatic joint is decoupled from the other degrees of freedom, as can be intuited from its Jacobian matrix (see Appendix A).

It was also observed that there was cross-talk in the wrist force-torque sensor. Experiments showed that forces perpendicular to the target direction would register significant readings. The system cannot distinguish between actual forces in the contact direction and these false readings, and thus it attempts to control these deviations. This effect becomes significant when the manipulator slides along the surface, generating significant frictional contact forces.

Finally, although the AdeptOne manipulator exhibits generally lower joint friction than the Puma 250, it is apparent that the effect of joint friction is still significant. This leads to the joint positioning plateaus, as well as the same sort of coupling between the force and position domains seen in the Puma 250.

It was observed during these tests that the angular position integrator in the wrist joint (joint 4) would sometimes “fight” with the force integrator, as depicted in Figure 3-15. The force performance against glass would suffer while the orientation of the wrist would be corrected. This problem arose because of the nature of the nugget contact with the wall. In order to fix this problem, the position integrator for the wrist was deactivated during the wall contact. Therefore the wrist joint is under joint PD control during contact. Perfect steady-state positioning of the wrist was sacrificed in favor of more robust force performance.

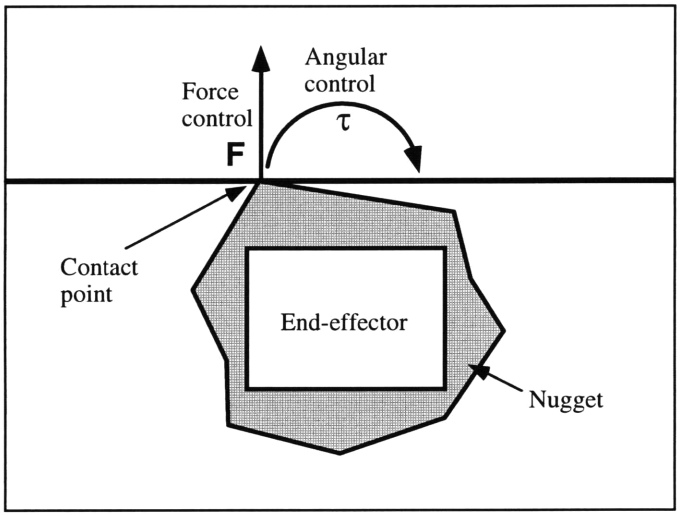


Figure 3-15: Contention Between Force and Position Integrators

3.3 Summary and Discussion of Experimental Results

The experimental results demonstrated above show that the implemented hybrid position/force control algorithms perform very well when the position controller is merely a regulator, and show some degree of success with simultaneous motion and force profiles. However, the performance degradation is significant. It is not clear that the disturbance force oscillations will always remain in a safe range, although the manipulators did not significantly exceed this range during the course of the experiments.

Additional experiments suggest that the force control performance degradation is caused by a variety of phenomena; this indicates that the force domain of the hybrid control algorithms is very sensitive to disturbances of any kind. Unmodelled dynamics, joint friction and transverse forces introduce time-varying disturbance forces into the control loop. The position domain is generally robust to these forces; the manipulator dynamics filters disturbances by integrating them twice. The force domain is not robust to these disturbances, however. They directly affect the endpoint force, and there is no filter to attenuate their effects. To obtain better performance with a hybrid controller, it may therefore be necessary to eliminate these disturbances. Additionally, the elimination of these disturbances would further verify their roles as the sources of the questionable behavior.

Degradation due to unmodelled dynamics could be removed by the implementation of a more complex control algorithm. Such a control algorithm would take into account the full dynamic model from (2 - 1). This requires the knowledge of inertial parameters. Cross-talk in the force sensor could be reduced by determining an appropriate decoupling matrix.

It may also be necessary to eliminate the effects of joint friction. Significant stick-slip behavior was seen in even the low-friction AdeptOne. A friction compensation scheme such as BaST control (Morel and Dubowsky, 1996) could remove the effects of friction at the low speeds required by the RACS control system.

The implemented hybrid position/force control algorithms are sufficient for pure regulation of position and force. Trajectory tracking performance is less reliable, but may be acceptable for controlling contact with the crucible. Further research is recommended to improve the behavior before the algorithms are incorporated into a RACS system.

Chapter 4

Surface Estimation Algorithm

In the preceding chapters, it has always been assumed that the orientation of the contact surface in task-space is known. When the nugget is in contact with the crucible wall, this is indeed true. However, when the nugget is brought into contact with other nuggets, this assumption is no longer valid. The nuggets are irregularly shaped and although it may be expected that the vision/packing subsystem (Sujan, 1998) will have some knowledge of the surface geometry, there is sufficient uncertainty such that this cannot alone be used to determine the direction of force control.

The presence of a force sensor near the endpoint provides contact force information which can in principle be used to determine the surface direction instantaneously. However, sensor data is likely to be corrupted by noise, and therefore the instantaneous estimate may be combined with previous data to yield a noise-optimal estimate. The following section outlines a method which uses the available data to arrive at such a best estimate of contact direction. This direction can then be converted into the appropriate **P** and **F** projection matrices which are used by the hybrid position/force control algorithm shown in Figure 2-1.

The method given in (Muto and Shimokura, 1993) makes use of both wrist force and torque information to determine an estimate of the surface normal and contact point location. This method also assumes accurate knowledge about the shape of the object in contact with the environment. The algorithm presented herein does not solve for contact point location, and thus does not require torque data. Nor does it assume any knowledge of the shape of the grasped nugget. Instead, the algorithm directly calculates the effects of sensor noise on an instantaneous surface normal estimate, and applies observer techniques to reduce these effects.

4.1 Instantaneous Measurement of Contact Surface

Figure 4-1 depicts the situation which is assumed in order to calculate the surface direction. For readability, it is portrayed in 2-D, but the analysis is valid in 3-D. A robot manipulator grips a nugget with its end-effector. The grasped nugget is in contact with another nugget with unknown surface normal direction $\hat{N}$. The contacted nugget surface is assumed to be continuous, and the radius of curvature of the nugget surface is assumed to be lower bounded by a known value ρ_{min}. It is also assumed that the nuggets are making contact at a single point, and that the nuggets are perfectly rigid. A wrist force sensor is located between the robot arm and the gripper and measures the contact force F. The gripper stiffness **K** is assumed to be known and constant in this case, but this restriction can be relaxed in favor of added algorithm complexity. It is also assumed that the manipulator moves slowly enough that the dynamics of the gripper can be represented entirely by this stiffness. That is, the motion is quasi-static and does not excite any inertial effects. The nominal Cartesian velocity of the end-effector v_{nom} is known exactly. The unknown actual velocity of the end-effector v deviates from v_{nom} due to compression or extension of the gripper. The contact force F and its time derivative $\dot{F}$ are measured by the force sensor and known to within the scalar uncertainties u_F and $u_{\dot{F}}$ respectively.

It is expected that $u_{\dot{F}}$ is quite large, as force sensor signals tend to be quite noisy, and numerical derivatives are imprecise. This analysis does not require any knowledge of friction coefficients.

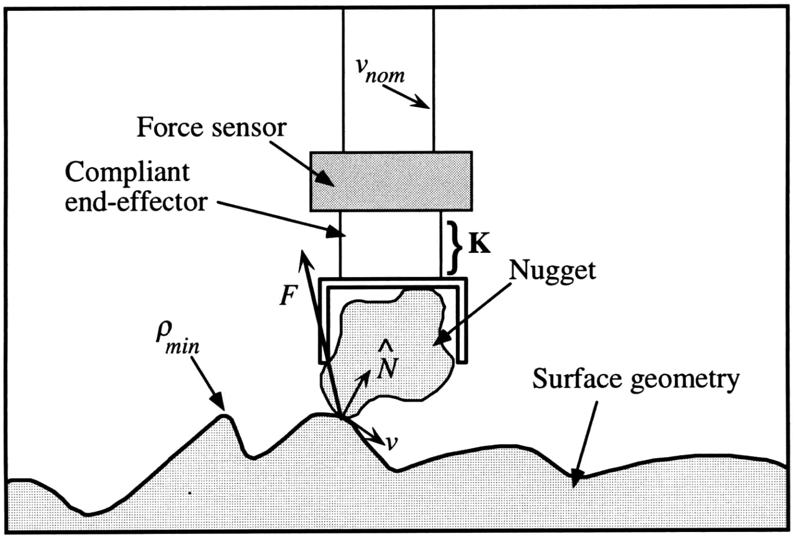


Figure 4-1: Surface Estimation

If there were no friction, the contact force would consist purely of a normal reaction force. Thus the vector $\hat{N}$ would simply be F scaled to unit length. However, friction does exist and therefore contributes to the force sensor signal. If the contact point is sliding along the surface with velocity v, the friction force acts to oppose this motion. Thus the total force F can be represented as

$$F = n\,\hat{N}_{meas} - f\,\hat{v} = N - f\,\hat{v}, \qquad \textbf{(4 - 1)}$$

where n is the magnitude of the normal force, $\hat{N}_{meas}$ is the surface normal unit vector that this procedure attempts to measure, N is the normal force vector, f is the magnitude of the friction force, and $\hat{v}$ is the unit vector in the direction of v.

The measured force F also directly determines the compression of the gripper, so that the actual position x of the endpoint deviates from the nominal position x_{nom} by the relation

$$F = \mathbf{K}(x - x_{nom}). \tag{4 - 2}$$

When (4 - 2) is differentiated with respect to time and re-arranged, the actual velocity v of the contact point can be estimated as

$$v = v_{nom} + \mathbf{K}^{-1}\dot{F}, \tag{4 - 3}$$

since $\mathbf{K}$ is positive definite and therefore invertible. Once v is calculated, determining the normal vector can be achieved by selecting out the friction component of the contact force. When contact is maintained, the end-effector can only move in a direction perpendicular to the nugget surface, since the nuggets are assumed to be perfectly rigid. Hence,

$$\hat{v}^{\mathrm{T}}\hat{N}_{meas} = 0. \tag{4 - 4}$$

Multiplying both sides of (4 - 1) by the appropriate projection matrix,

$$\left(\mathbf{I} - \frac{vv^{\mathrm{T}}}{v^{\mathrm{T}}v}\right)F = n\left(\mathbf{I} - \frac{vv^{\mathrm{T}}}{v^{\mathrm{T}}v}\right)\hat{N}_{meas} - f\left(\mathbf{I} - \frac{vv^{\mathrm{T}}}{v^{\mathrm{T}}v}\right)v, \tag{4 - 5}$$

It turns out that the rightmost term of (4 - 5) is zero. Plugging (4 - 4) into (4 - 5) produces the equation

$$N = n\,\hat{N}_{meas} = \left(\mathbf{I} - \frac{vv^{\mathrm{T}}}{v^{\mathrm{T}}v}\right)F. \tag{4 - 6}$$

Notice that the right-hand side of (4 - 6) consists only of known quantities. The unit vector $\hat{N}$ is simply calculated from N as

$$\hat{N}_{meas} = \frac{N}{\sqrt{N^{\mathrm{T}}N}}. \tag{4 - 7}$$

Therefore, an instantaneous estimate of the contact surface direction can be obtained from sensor data and Equations (4 - 3), (4 - 6) and (4 - 7).

4.2 Uncertainty of the Instantaneous Estimate

It is apparent that this estimate is likely to be in error, as the uncertainties u_F and $u_{\dot{F}}$ in the sensor measurements propagate through to an uncertainty u_{meas} in the surface normal.

4.2.1 Uncertainty due to $\mathbf{u_F}$

Figure 4-2 depicts the situation where there is no uncertainty in $\dot{F}$ and hence no uncertainty in v, but that the actual value of F deviates from its measured value by a vector of length u_F. The orthogonal coordinate frame a-b-c is oriented such that a points in the direction of v, and b points in the direction of N.

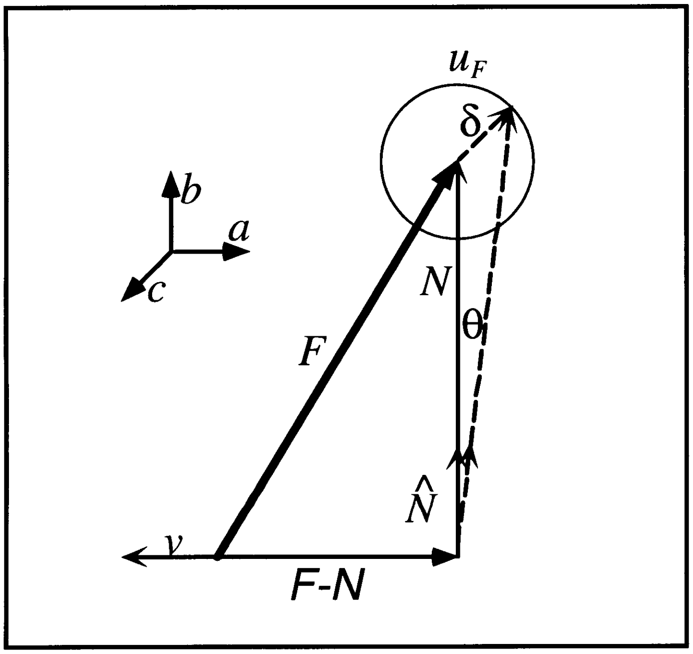


Figure 4-2: Uncertainty Propagation due to u_F

It is clear that a deviation of F in the a direction does not alter the estimate of $\hat{N}$, because all components of F in this direction are already ignored. Similarly, a deviation of F in the b direction does not alter the estimate of $\hat{N}$, serving only to lengthen or contract the vector *N*. So, only deviations in the *c* direction remain. A small deviation of size δ in the *c* direction causes *N* to swing by an angle of approximately

$$\theta \approx \frac{\delta}{\sqrt{N^T N}}. \qquad \textbf{(4 - 8)}$$

This angle is also the magnitude of the deviation (uncertainty) in the unit vector $\hat{N}$. If it is assumed that the deviation vector does not, on average, favor any direction *a-b-c*, the component of the vector in the *c* direction is likely to be

$$\delta = \frac{u_F}{\sqrt{3}}. \qquad \textbf{(4 - 9)}$$

Thus, the uncertainty in $\hat{N}$ due exclusively to u_F is given by

$$u_{\hat{N}}\Big|_v = \frac{u_F}{\sqrt{3N^T N}}. \qquad \textbf{(4 - 10)}$$

4.2.2 Uncertainty due to $u_{\dot{F}}$

Now consider the situation in which there is no uncertainty in *F*, but where the actual value of $\dot{F}$ deviates from its measured value by a vector of length $u_{\dot{F}}$. From (4 - 3), this uncertainty contributes directly to an uncertainty in *v* by the (conservative) relation:

$$u_v = \frac{u_{\dot{F}}}{\min(\mathrm{eig}(\mathbf{K}))}, \qquad \textbf{(4 - 11)}$$

where the denominator represents the smallest eigenvalue of the stiffness matrix **K**, which is expected to be positive definite. Thus the denominator is greater than zero. This equation indicates that the higher the stiffness of the end-effector, the more accurately *v* can be estimated in the presence of sensor noise.

Again, the orthogonal coordinate frame *a-b-c* is oriented such that *a* points in the direction of *v*, and *b* points in the direction of *N*.

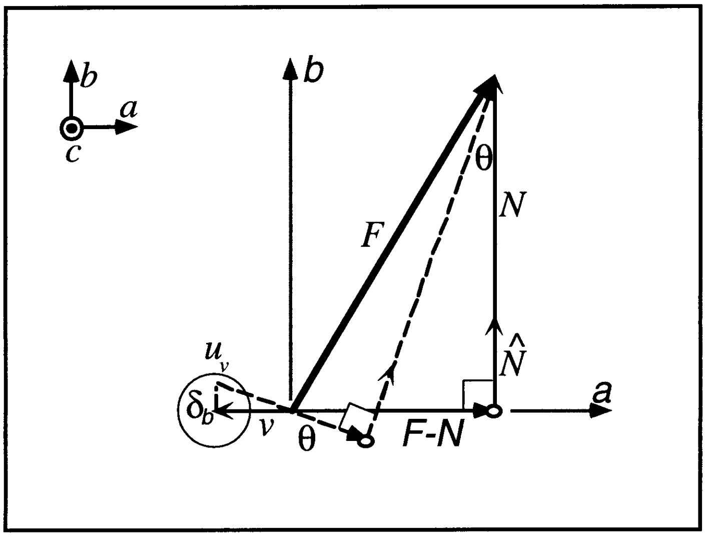


Figure 4-3: Uncertainty Propagation due to u_v in the b Direction

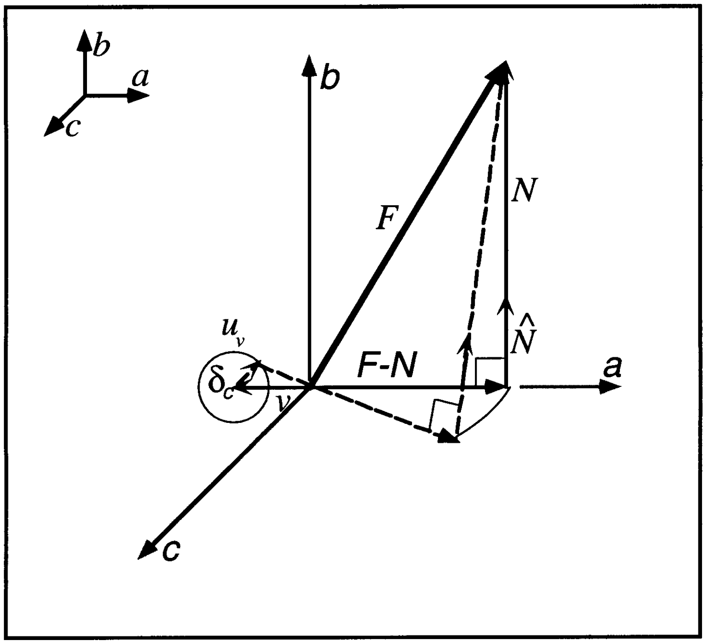


Figure 4-4: Uncertainty Propagation due to u_v in the c Direction

It is clear that a deviation of v in the a direction does not alter the estimate of $\hat{N}$, because the direction of v does not change. In the case of a deviation δ_b in the b direction, it can be seen from Figure 4-3 that both N and v rotate by an angle θ which is given by

$$\theta \approx u_{\hat{N}b} \approx \frac{\delta_b}{\sqrt{v^T v}}. \tag{4 - 12}$$

The effect of a deviation δ_c in the c direction, shown in Figure 4-4, can be calculated as follows: It can be seen that this change in v scales to a change in the vector (F-N) by the following relation.

$$u_{(F-N)} = \frac{\delta_c \sqrt{(F-N)^T(F-N)}}{\sqrt{v^T v}}. \tag{4 - 13}$$

The change in (F-N) equals in magnitude the change in N (F is fixed). The resulting change in $\hat{N}$ is obtained by scaling down from N to unit length. In other words,

$$u_{\hat{N}c} = \frac{u_N}{\sqrt{N^T N}} = \frac{u_{(F-N)}}{\sqrt{N^T N}} = \frac{\delta_c \sqrt{(F-N)^T(F-N)}}{\sqrt{v^T v}\sqrt{N^T N}}. \tag{4 - 14}$$

If it is again assumed that the deviation vector does not, on average, favor any direction a-b-c, the component of the vector in the b,c directions are likely to be

$$\delta_b = \delta_c = \frac{u_F}{\sqrt{3}}. \tag{4 - 15}$$

The worst-case magnitude of the total $u_{\hat{N}}$ vector is the sum of the two magnitudes calculated in (4 - 12) and (4 - 14). Thus, the uncertainty in $\hat{N}$ due exclusively to $u_{\dot{F}}$ is given by

$$u_{\hat{N}}\Big|_F = \frac{u_{\dot{F}}}{(\min(\mathrm{eig}(\mathbf{K}))\sqrt{3v^T v}}\left(1 + \frac{\sqrt{(F-N)^T(F-N)}}{\sqrt{N^T N}}\right). \tag{4 - 16}$$

4.2.3 Total Uncertainty of Measured Normal Vector

It is additionally assumed that the uncertainty u_F in the force sensor signal and the uncertainty $u_{\dot{F}}$ in its time derivative are independent. This assumption is motivated by the fact that sensor noise is generally not regular or periodic, and thus its value does not show much correlation with its time derivative. In this case, the most likely magnitude of the error in the measured value of $\hat{N}$ is determined by the error propagation formula for standard deviations (Beckwith *et al.,* 1993):

$$u_{meas} = \sqrt{\left(\frac{\partial u_{\hat{N}}}{\partial u_F} \bigg|_{\dot{F}} \right) u_F^2 + \left(\frac{\partial u_{\hat{N}}}{\partial u_F} \bigg|_F \right) u_{\dot{F}}^2} \; . \qquad \textbf{(4 - 17)}$$

The partial derivatives are simply derived from the results of (4 - 10) and (4 - 16). Substituting these into (4 - 17) yields the final value of u_{meas},

$$u_{meas} = \sqrt{\left(\frac{1}{\sqrt{3N^T N}} \right) u_F^2 + \left(\frac{\sqrt{N^T N} + \sqrt{(F-N)^T (F-N)}}{(\min(\mathrm{eig}(\mathbf{K}))\sqrt{3v^T v}\sqrt{N^T N}} \right) u_F^2} \; . \qquad \textbf{(4 - 18)}$$

This expression represents the total uncertainty in the measured surface normal unit vector as a function of sensor noise. It is readily seen from this formula that the uncertainty in the measured normal vector decreases with an increase in the magnitudes of *N* and *v*.

4.3 The Surface Estimation Algorithm

The measured surface normal $\hat{N}_{meas}$ is likely to be significantly corrupted by measurement noise, u_{meas}. A better estimate can be obtained by implementing the following observer algorithm to reconstruct the actual unknown state, $\hat{N}$. This algorithm operates in discrete time. The surface normal can be thought of as displaying the following dynamics:

$$\begin{aligned} \hat{N}(k+1) &= \hat{N}(k) + W(k), \\ \hat{N}_{meas}(k) &= \hat{N}(k) + V(k). \end{aligned} \qquad \textbf{(4 - 19)}$$

In this case, the model assumes that the surface normal is nominally expected not to change from one iteration step k to the next step (k+1). However, there is a "driving noise" term *W*(*k*) which represents the actual change during the time step. The sensor-based measurement of the surface normal $\hat{N}_{meas}$, given by (4 - 7), is expected to differ from the actual vector by measurement noise $V(k)$, which is related to the uncertainty u_{meas} given in (4 - 18).

The expected value of the driving noise $W(k)$ is zero, and its covariance matrix $\mathbf{Q}$=E[WW^T] (where E[x] represents the expected value of x) can be estimated thusly: during the time T between iterations, the contact point has moved a distance of approximately $|vT|$. Since the nugget surface has a minimum radius of curvature ρ_{min}, the normal vector direction has changed from the previous estimate by at most $|vT/\rho_{min}|$, and the square of the change is at most $(vT/\rho_{min})^2$. It is approximated that the change in each direction is uncorrelated, and thus this error is distributed evenly along the main diagonal of the covariance matrix. Therefore, an estimate of the covariance matrix is:

$$\mathbf{Q}(k) = E[W(k)W^T(k)] = \frac{1}{3}\left(\frac{v(k)T}{\rho_{\min}}\right)^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \qquad \textbf{(4 - 20)}$$

For the measurement noise $V(k)$, the expected value is also zero. Its covariance matrix $\mathbf{R}$=E(VV^T) is estimated as follows: It is assumed for simplicity that the variance is approximately the square of the uncertainty u_{meas} given in (4 - 18), and that this uncertainty is evenly distributed among the three orthogonal directions. Therefore, it is assumed that:

$$\mathbf{R}(k) = E[V(k)V^T(k)] = \frac{1}{3}\left(u_{meas}(k)\right)^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \qquad \textbf{(4 - 21)}$$

A linear, recursive observer can now be derived. The computation begins with the algorithm's previous best estimate of $\hat{N}_{\text{best}}(k)$ and the associated covariance matrix of its error $\mathbf{P}(k)$. The $\mathbf{P}(k)$ matrix is a measure of the uncertainty in the algorithm's estimate at time step k. The linear, recursive algorithm takes the form:

$$\hat{N}_{best}(k+1) = \hat{N}_{best}(k) + \mathbf{G}(\hat{N}_{meas}(k) - \hat{N}_{best}(k)) \,. \tag{4 - 22}$$

Thus the new best estimate of the normal vector deviates from the old best estimate by a linear combination **G** of the error between the new measurement and the old best estimate. The gain matrix **G** can be optimized to minimize the new error covariance matrix **P**(k+1) while ensuring that the new estimate is indeed of unit length. It can be shown (Gelb, 1974) that the new covariance matrix is given by:

$$\mathbf{P}(k+1) = (\mathbf{I} - \mathbf{G})\mathbf{P}_d(k)(\mathbf{I} - \mathbf{G})^{\mathrm{T}} + \mathbf{G}\mathbf{R}(k)\mathbf{G}^{\mathrm{T}} \,. \tag{4 - 23}$$

where $\mathbf{P}_d(k)$ is the covariance matrix of the old estimate degraded by the driving noise:

$$\mathbf{P}_d(k) = \mathbf{P}(k) + \mathbf{Q}(k) \,. \tag{4 - 24}$$

The best estimate is the unit vector which minimizes the "magnitude" of the new covariance matrix **P**(k+1). A convenient metric for the magnitude of a positive-definite matrix is its trace (the sum of the main diagonal elements). Thus the optimal gain **G** must be chosen to minimize

$$\mathrm{trace}(\mathbf{P}(k+1)) + \lambda(\hat{N}^T_{best}(k+1)\hat{N}_{best}(k+1) \; - \; 1), \tag{4 - 25}$$

where λ is a Lagrange multiplier which enforces the vector unit magnitude constraint. This minimization occurs when the following two equations are solved simultaneously:

$$\begin{gathered} \mathbf{G} = [\mathbf{P}(k) - \lambda\hat{N}_{best}(k)e^T(k)][\mathbf{P}(k) + \mathbf{R}(k) + \lambda e(k)e^T(k)]^{-1} \\ 2\hat{N}^T_{best}(k)\mathbf{G}e(k) + e^T(k)\mathbf{G}^T\mathbf{G}e(k) = 0 \end{gathered} \tag{4 - 26}$$

where *e(k)* is defined to be:

$$e(k) \equiv \hat{N}_{meas}(k) - \hat{N}_{nest}(k). \tag{4 - 27}$$

In general, the equations in (4 - 26) must be solved numerically (via, for example, Newton's method) to eliminate the Lagrange multiplier λ. Notice that if the unit vector constraint is relaxed, (if the second equation is not satisfied and λ=0), the gain **G** reduces to that of a standard Kalman filter, as presented in (Gelb, 1974).

The entire observer algorithm has been presented; one employs (4 - 26) to determine the appropriate gain, and then (4 - 22) and (4 - 23) to solve for the new best estimate $\hat{N}_{\text{best}}(k+1)$ and its covariance matrix $\mathbf{P}(k+1)$. These are the values which are required to begin the next iteration.

4.4 Expected Behavior of the Surface Estimation Algorithm

There are several important cases to explore when describing the behavior of the algorithm: the first is the response when the end-effector halts on the surface. The second is the algorithm's response to losing contact. The next case is the system behavior as a function of various sensor noise levels. The next case is the system behavior as a function of various minimum radii of curvature. After these extremes are discussed, the typical behavior of the algorithm can be described.

4.4.1 End-effector Halts

If the end-effector comes to a halt, and thus the contact point stops moving (v=0), the uncertainty in the measurement becomes infinite, as seen in (4 - 18). Thus $\mathbf{R}$ is also infinite, see (4 - 21). Without motion, the instantaneous force data cannot provide any knowledge of the contact surface orientation. Whereas it is true that the contact force must lie in the friction cone to maintain zero velocity, no knowledge of friction coefficient has been assumed, and so no extra information is provided.

Also, if the contact point does not move, (4 - 20) shows that $\mathbf{Q} = 0$. There is no degradation of the old estimate; the surface normal in the current time step is exactly identical to that of the previous time step. Therefore, the Kalman filter chooses to ignore the current force data in favor of its previous estimate, and the covariance matrix $\mathbf{P}$ does not change. When the contact point comes to a stop, so does the estimation. The estimate is neither improved nor degraded but remains stagnant until the end-effector begins to move.

4.4.2 Contact Broken

When contact is broken, the force F immediately drops to zero, causing the uncertainty $\mathbf{R}$ in the measured data to be infinite. Thus the Kalman filter again completely ignores the force data in determining its estimate of the normal vector. However, the end-effector is now moving, and Equation (4 - 20) yields a nonzero $\mathbf{Q}$. There is therefore a degradation of the old estimate, and the covariance matrix $\mathbf{P}$ diverges steadily until the end-effector stops moving or re-establishes contact with the surface.

4.4.3 Varied Sensor Noise Level

If the sensor noise is very small compared to the sensor signals, the uncertainty $\mathbf{R}$ in the measured normal vector is also very small, unless v=0 or F=0 as discussed before. This causes increased reliance on the new data and produces a small covariance matrix $\mathbf{P}$, as expected. If the sensor noise is very large compared to the sensor signals, the uncertainty $\mathbf{R}$ in the measured normal vector is also very large. This causes increased reliance on the old data, and produces a larger covariance matrix $\mathbf{P}$. Obviously it is preferable to minimize the effect of sensor noise.

The stiffness matrix $\mathbf{K}$ of the end-effector alters the effect of sensor noise. The higher the stiffness, the less damaging the effect of noise in the derivative of the sensor force. Increasing the end-effector stiffness is generally easier than decreasing sensor noise (especially in the time derivative), and therefore is a more reasonable method of minimizing sensor noise.

It is assumed that **K** is exactly known; the algorithm could be modified to accommodate uncertainty in the stiffness. The effect of this uncertainty would be similar to uncertainty in $\dot{F}$; the sensitivity is lower for stiffer end-effectors.

4.4.4 Varied Minimum Radius of Curvature

If ρ_{min} is very small, then either the *a priori* knowledge of the minimum surface curvature is very poor, or the surface is known to be extremely rough. In the limiting case that it is exactly zero (the surface may have a completely discontinuous slope) the presented algorithm has infinite **Q** unless the contact point is stationary. This implies that the observer relies exclusively on the force data, because there is zero confidence that the previous estimate is still valid.

If, on the other hand, ρ_{min} is chosen to be very large, then the surface is known to have a nearly constant slope. In the limiting case that it is infinite, the presented algorithm always has **Q**=0. New force data is used to improve the estimates, and the covariance matrix **P** converges rapidly to zero. After **P** becomes sufficiently small compared to the uncertainty in the new data **R**, the system effectively ignores the data. Thus after a short time, the Kalman filter “goes to sleep”. This is only acceptable if the surface is known with absolute certainty to have no change in slope. In general, if the value of ρ_{min} is overestimated, the system may not respond as quickly as it must to maintain an accurate estimate.

It is thus important to pick a reasonable value of ρ_{min} which represents the maximum possible knowledge of the surface smoothness without being overly ambitious in either direction.

4.4.5 Typical Behavior

Typically, for set values of all the parameters, a balance is struck between the increase in uncertainty **Q** due to the unknown surface and the decrease in uncertainty **P** through an appropriate choice of gain matrix. In general, it is not expected that the estimate uncertainty converges to zero or diverges to infinity, but rather finds some relatively stable level of uncertainty which places the estimate within a certain distance of the actual value.

Hybrid force-position control should be most successful if this level is small. Using the current algorithm, this can be accomplished through the use of less noisy sensors or by making stronger assumptions about the minimum radius of curvature. Also, the accuracy increases if the sensor noise is small compared to the measurements, implying that larger contact forces improves the estimate.

4.5 Possible Modifications to the Algorithm

The algorithm of the form shown in (4 - 22) is not necessarily the optimal estimator, but claims merely to be the optimal linear recursive estimator. The constraint that the estimate be of unit length is inherently nonlinear, and it is not possible to linearize the observer without the discarding of some information. There is, in theory, an estimator which is recursive and nonlinear that would use the same information to arrive at a more accurate estimate of the normal vector, (i.e., the covariance matrix **P** would be smaller). Unfortunately, this estimator is much more complex in form than (4 - 22), and it may not be possible to solve the appropriate equations in a closed form. A nonlinear estimator which requires numerical solution of simultaneous transcendental equations may involve more computation than is allowable in the time *T* between iterations. Hence the motivation to rely on a simpler linear estimator.

Additionally, the uncertainties calculated in section 4.2 and the covariance matrices **P** and **Q** estimated in (4 - 20) and (4 - 21) may be rather conservative and more rigorous calculation could be performed to attempt to tighten the boundaries. This would result in a slightly more convergent estimator. Similarly, this may require significantly more complex calculations for a small increase in system performance.

The end-effector stiffness matrix **K** is assumed to be a known constant. In an actual system, this is not true of any manipulator except for possibly a Cartesian robot. For the RACS system the algorithm would need to be modified to reflect the actual stiffness of the end-effector as a function of its configuration.

Finally, the current algorithm assumes that there is no source of information other than the force sensor data. In the actual RACS system, there will be a vision subsystem which will store a representation of the nugget landscape in the crucible, see (Sujan, 1998). This information could be used to alter the dynamic model of equation (4 - 19) so that the surface normal is not expected to remain the same from one iteration to the next, but is expected to follow the vision-system predicted surface normal. This added knowledge would serve to decrease the driving noise covariance matrix **Q** and thus help assure convergence. This would not introduce significant complexity to the algorithm and therefore would be a feasible modification.

Chapter 5

Simulation Results of Surface Estimation Algorithm

5.1 Overview of Simulation

The surface estimation algorithm presented in Section 4.3 is explored via a 2-dimensional simulation with the Simulink software. The initial set of simulations evaluates the ability of the system to determine the correct normal vector. Further simulations test the ability of the system to use this normal vector while performing hybrid position/force control.

5.2 Determination of the Normal Vector

The block diagram representing the initial set of surface estimation tests is shown in Figure 5-1. The system generates a nominal trajectory x_{nom} and v_{nom} as a function of time. This is fed to a contact model, which knows the shape and location of the surface, as well as the end-effector stiffness **K**. The contact model then determines the actual location of the contact point, and the contact forces generated by this trajectory. The output of the contact model is the contact force F and its derivative $\dot{F}$.

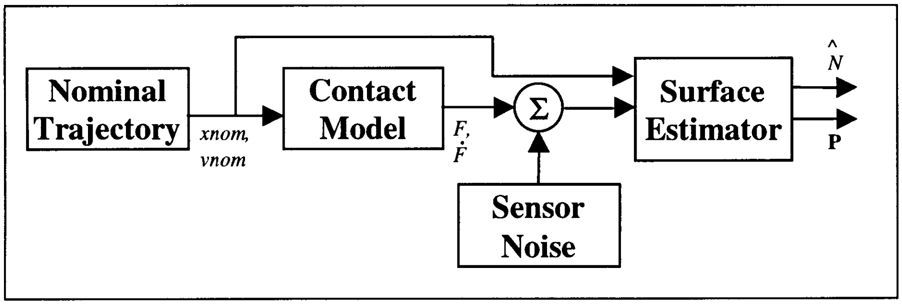


Figure 5-1: Block Diagram for Surface Estimation Tests

The contact model output is then polluted with noise which adds uncertainties u_F and $u_{\dot{F}}$ to form the actual sensor measurements. These measurements and the nominal trajectory are then passed to the surface estimation algorithm, which generates its best current estimate of the normal vector $\hat{N}$ and the associated error covariance matrix, **P**. Thus the system runs "open-loop" in that the surface estimator merely observes the system behavior and does not influence it in any way.

5.2.1 The Contact Model

The surface simulated for these experiments is shown in Figure 5-2. The surface shape is the parabola:

$$y = 0.05x^2 . \tag{5 - 1}$$

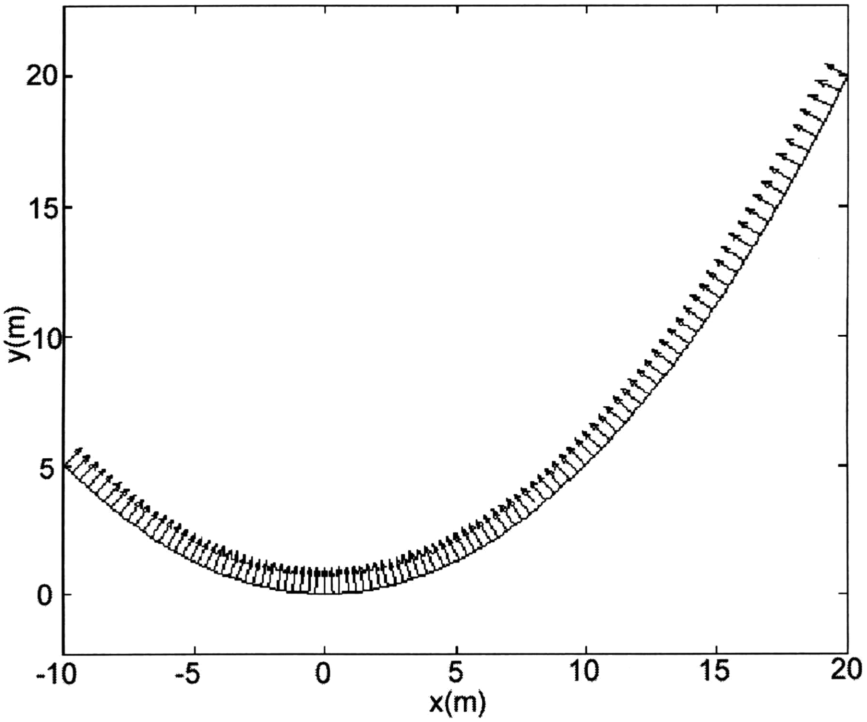


Figure 5-2: The Contact Surface

Also shown in the figure is the orientation of the normal vector along the surface. A two-dimensional surface normal is most easily described by the angle θ made with the horizontal. For this surface, the angle θ is given by:

$$\theta = 90° + \tan^{-1}\left(x/_{10}\right). \qquad (5 - 2)$$

The radius of curvature ρ of this surface is given by the formula

$$\rho = 10(1 + 0.01x^2)^{3/2}, \qquad (5 - 3)$$

which achieves its minimum value of 10 when $x = 0$. Thus $\rho_{\min}$ is 10, although this may not be known to the surface estimator.

The contact model simulates coulomb friction with a static coefficient of friction μ. The actual endpoint x and nominal endpoint x_{nom} are known, and the contact force is calculated by (4 - 3). If the force component in the surface normal direction is negative, contact is considered to be broken, and the endpoint is moved to the nominal value. Conversely, contact is considered to be made if the current nominal position is below the surface, and the endpoint is moved onto the surface. If the transverse force along the surface is less than the maximum friction force, the contact point remains stationary. If the transverse force begins to exceed the maximum friction force, the contact point begins to slide along the surface. When the contact point slides, the friction force equals its maximum value. These algebraic constraints can be iterated to determine the current contact point location and contact force.

In the cases explored, the contact model assumes that the end-effector stiffness **K** is equal to 100 **I**, and the friction coefficient $\mu = 1$.

5.2.2 Simulation Results

A simulation was performed in which the sensor noise $u_F = 1$, $u_{\dot{F}} = 5$, and the assumed ρ_{min} is 0.75, (which is much smaller than the actual minimum radius.) The nominal trajectory of the endpoint is given by

$$\begin{bmatrix} x_{nom}(t) \\ y_{nom}(t) \end{bmatrix} = \begin{bmatrix} 0.2t \\ -0.1 \end{bmatrix} for\ t < 15, \begin{bmatrix} 3 \\ -0.1 \end{bmatrix} for\ t \geq 15.. \qquad (5-4)$$

Thus the system end-effector attempts to move toward the right with velocity 0.2 until t=15, when the end-effector halts. Figure 5-3 depicts the typical results.

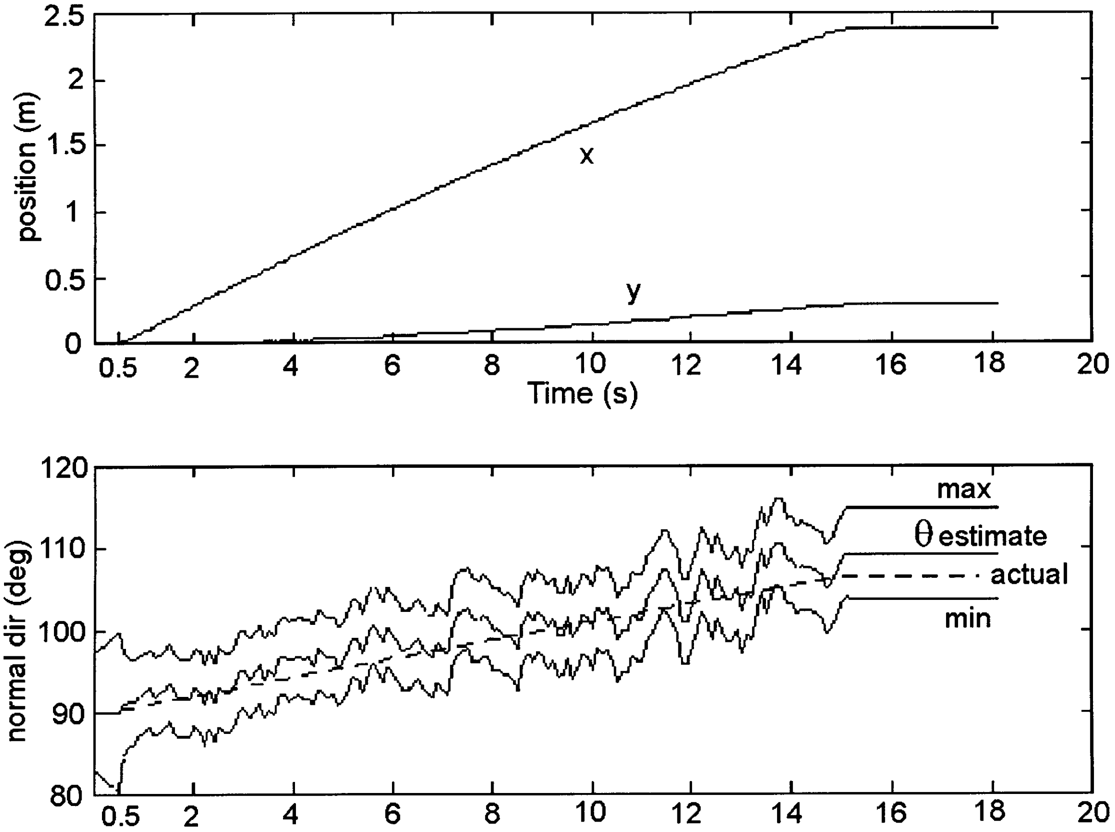


Figure 5-3: Typical Open-Loop Estimation Results

The position of the endpoint is plotted in the top graph, and the normal vector estimation results are plotted in the bottom graph. The estimate $\bar{\theta}$ is surrounded by a "boundary layer" which corresponds to the covariance matrix **P**. The wider this layer, the larger the uncertainty.

In the beginning of the trial, until approximately 0.5 seconds, the contact force is small. The uncertainty in the instantaneous normal vector estimate is large enough so that the measured vector is completely ignored. Thus the algorithm's estimate is temporarily fixed at its original value of 90°. The uncertainty can be seen to diverge until the uncertainty in the incoming data is small enough (at about 0.5 seconds). After this, the system estimate tracks the actual value at least as well as the covariance matrix predicts. The covariance matrix achieves an equilibrium in uncertainty of about ±6°.

At t=15 seconds, the contact point comes to rest, and the estimate and associated uncertainty stagnate, as discussed.

Figure 5-4 shows the results of the same system with a more liberal estimate of minimum curvature radius, ρ_{min} = 10. These results are similar qualitatively to those shown in Figure 5-3, but the uncertainty in the surface normal estimate has dropped to ±3°. This more reasonable curvature estimate has improved the accuracy of the algorithm significantly, as was mentioned in section 4.4.4.

If sensor noise is removed, the results are those shown in Figure 5-5. As predicted, the uncertainty immediately converges to zero. Also, the estimate is exactly equal to the actual value of the normal vector, as is expected.

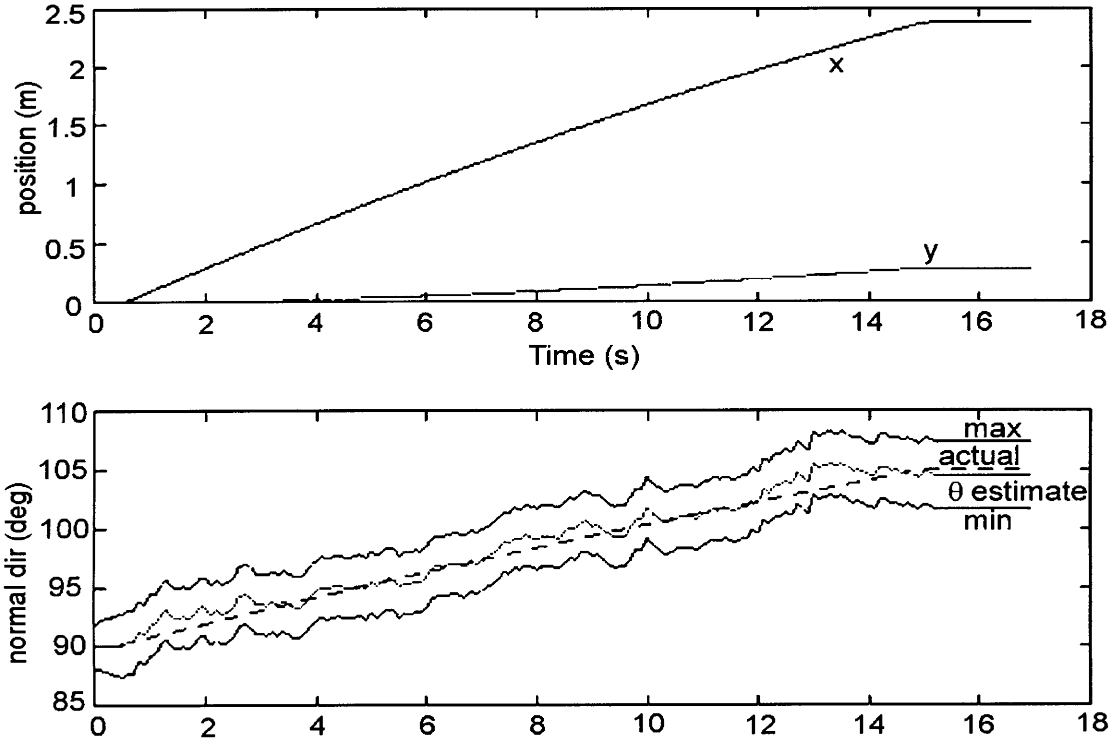


Figure 5-4: Results with Less Conservative Curvature Estimate

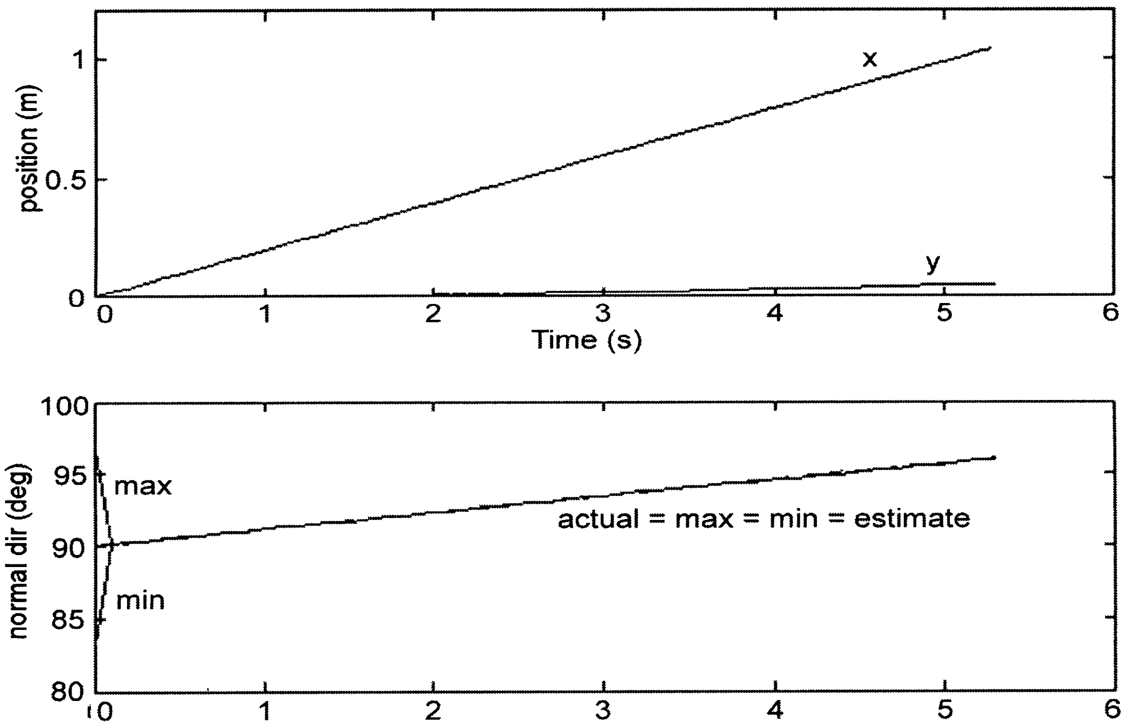


Figure 5-5: Results with No Sensor Noise

The contrasting case, where the sensor noise is made very large, is shown in Figure 5-6. Notice that the incoming force data is never used to determine the estimate; instead, it relies on its initial estimate of 90° while the uncertainty increases at a significant rate. At the end of 3.5 seconds, the uncertainty is already ±20°. It is clear that in this case the algorithm performs very poorly, because it cannot update its estimate.

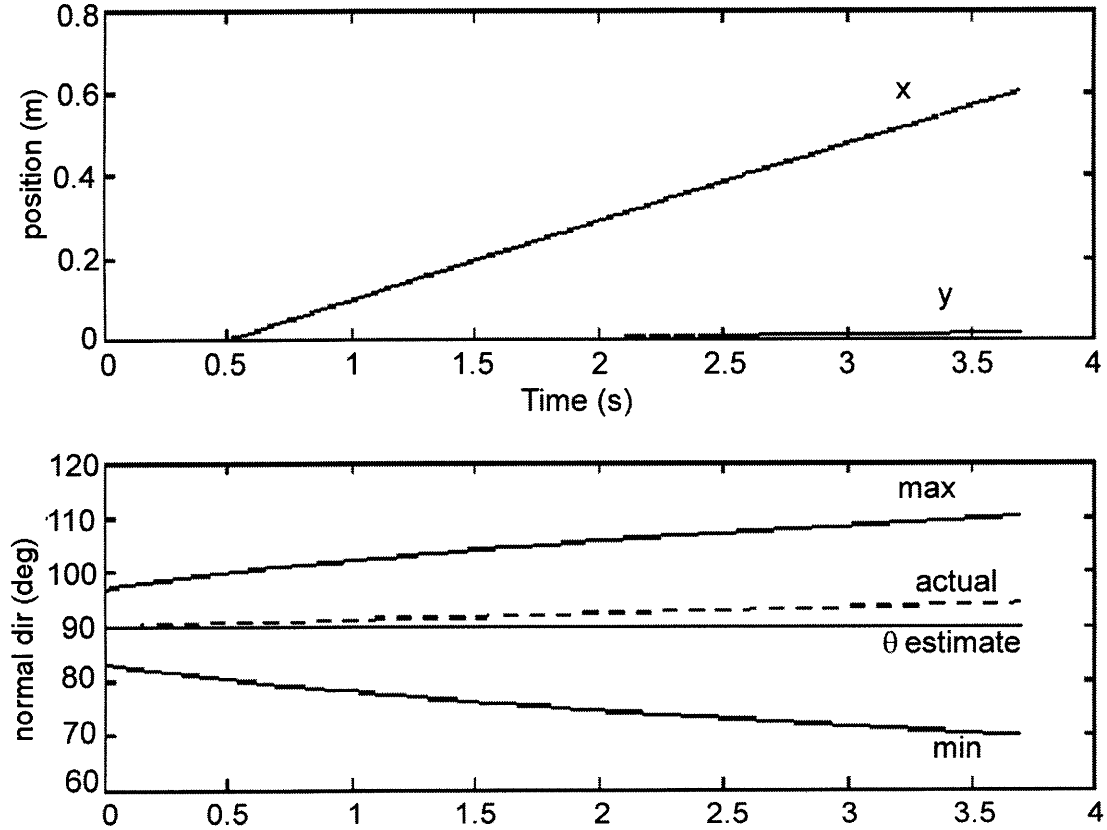


Figure 5-6: Results with Large Sensor Noise

5.3 Integration of Estimator with Hybrid Position/Force Control

Now that the algorithm has been demonstrated in a purely observational capacity, the use of the algorithm in conjunction with hybrid position/force control is explored.

5.3.1 Overall System

The block diagram for these simulations is shown in Figure 5-7. Now the force sensor data as well as the normal vector estimate $\hat{N}$ is being fed into a hybrid controller, which generates a control effort to be exerted by the manipulator. The manipulator dynamics are a function of the control effort and the actual contact forces (without noise). These manipulator dynamics then generate the nominal trajectory which is available to both the contact model and the surface estimator.

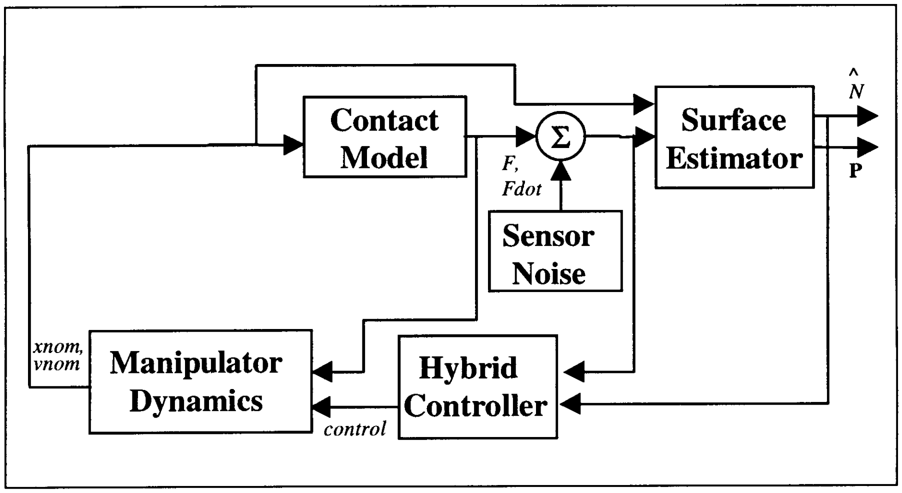


Figure 5-7: Block Diagram for Control Using Surface Estimation

5.3.2 The Control Algorithm

The hybrid control algorithm is essentially the same as shown in Figure 2-2. In this case, the selection matrices **P** and **F** which define the position and force subspaces respectively are calculated to be:

$$\begin{aligned} \mathbf{P} &= \mathbf{I} - \hat{N}\hat{N}^T, \\ \mathbf{F} &= \hat{N}\hat{N}^T, \end{aligned} \qquad (5\text{-}5)$$

which are complementary projection matrices since $\hat{N}$ is of unit length. Also, the reference position and force profiles are functions of the surface estimate, since it is desired that the contact force be normal to the surface and that the motion is along the surface:

$$F_{des} = -F_d \hat{N}, \qquad \textbf{(5 - 6)}$$
$$v_{des} = V_d (\hat{N} \times \hat{z}),$$

where F_d and V_d are the scalar values of desired contact force and velocity. The vector $\hat{z}$ is the unit vector in the z direction, perpendicular to the x-y plane. The cross-product of this vector with the surface normal ensures that the desired motion is rotated 90° in the x-y plane. For stability, it is important that F_d be positive to maintain contact. However, V_d may be positive or negative, depending on the desired direction of motion. Also note that the desired position profile is described in terms of velocity, which more conveniently represents the constraint on motion. The desired position profile can be generated in real-time by integrating the second equation in (5 - 6).

5.3.3 Contact Model and Manipulator Dynamics

In the following experiments, the surface to track is the same as shown in Figure 5-2; the surface exerts a normal force and a sliding coulomb friction. The stiffness matrix **K** is again assumed to be 100 **I**.

The manipulator dynamics written in Cartesian endpoint coordinates (x,y) are assumed to be of the following form:

$$u_x = m_x \ddot{x} + b_x \dot{x} - F_{contact,x} \qquad \textbf{(5 - 7)}$$
$$u_y = m_y \ddot{y} + b_y \dot{y} - F_{contact,y}$$

where m_x, m_y, b_x, b_y, are the decoupled inertial and damping parameters for the system, u_x and u_y are the efforts supplied by the control algorithm, and $F_{contact}$ represents the forces supplied by the contact model. These dynamics, which are assumed to be linear and decoupled for simplicity, could be achieved with a real manipulator by a sufficiently accurate model-based inner control loop.

5.3.4 Simulation Results

Experiments were performed in which the noise levels are again $u_F = 1$, $u_{\hat{F}} = 5$, and the assumed $\rho_{\min}$ is again 0.75. The desired velocity V_d is 0.5 and the desired contact force F_d is 5. Thus the system end-effector attempts to move along the unknown surface at constant velocity while maintaining a constant contact force. Typical results for this experiment are shown in Figure 5-8.

The contact force is shown in the top graph; the position of the endpoint is plotted in the center graph, and the normal vector estimation results are plotted in the bottom graph. The estimate $\bar{\theta}$ is again surrounded by a "boundary layer" which corresponds to the covariance matrix **P**.

At the beginning of the trial, the contact force is initially too large, and within one second, the force is regulated to approximately 5 Newtons. For the remainder of the trial, the system moves along the surface, maintaining a contact force which varies between 4.5 and 5.2 Newtons. The force error is kept less than 10% in the face of the uncertainty in the normal vector of approximately ±6°. The trial is successful in that the system remains stable and within a moderately tight envelope of the desired behavior.

If the noise in the sensor is removed, the results are as shown in Figure 5-9. Again, the surface normal estimate immediately converges to its actual value, and the uncertainty boundary layer is infinitesimally narrow.

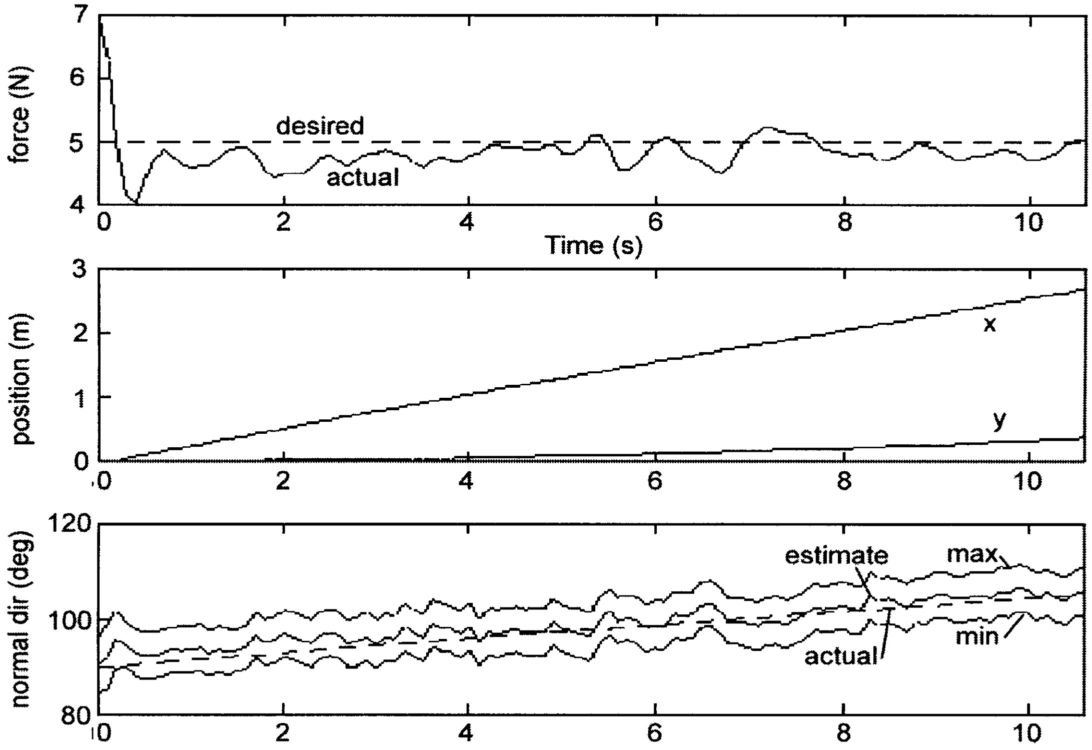


Figure 5-8: Typical Surface-Tracking Results

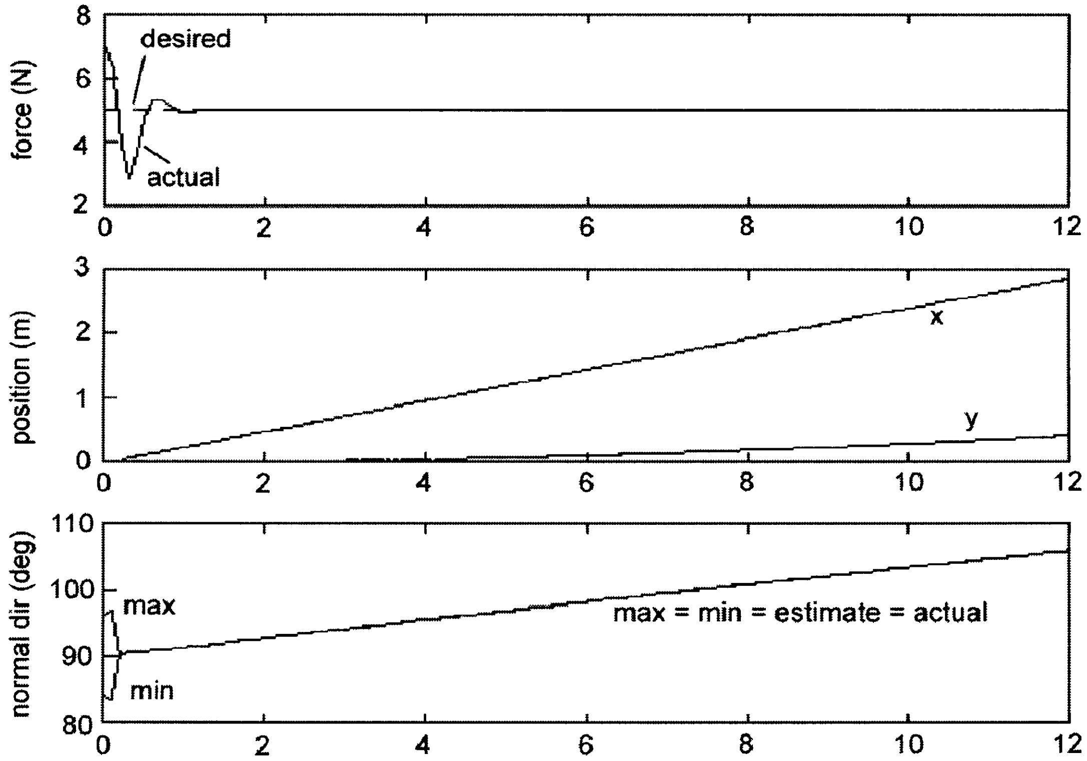


Figure 5-9: Surface Tracking Results with No Sensor Noise

Again, the contact force is initially too high and converges to its final state within one second. Now, however, the contact force converges to almost exactly 5 Newtons and remains there for the remainder of the trial. Therefore the 10% errors seen in Figure 5-8 result solely from the uncertainty in the normal vector, and not from any inherent errors in the control algorithm.

5.4 Summary

The results presented in this chapter show that the proposed surface estimation algorithm has the potential for use in hybrid control problems without extensive prior knowledge of the contact surface. Also, as was mentioned in Section 4.5, any additional information about the surface can be factored into the algorithm to improve the system response.

Chapter 6

Discussion & Conclusions

6.1 Summary of Work

It has been demonstrated that hybrid position/force control based on simple Jacobian Transpose or Jacobian Inverse control performs very well when the position controller is merely a regulator, and shows some degree of success with simultaneous motion and force profiles. In cases where the robot loses contact with the surface, contact is re-established safely. However, the reliability of the control system under simultaneous motion and force control is questionable. It was not clearly determined that the system would always produce safe forces. Sensor cross-talk and stick-slip friction produce force disturbances which degraded controller performance.

The surface estimation routine shows promising results in simulation, both as a pure observer and as a supplement to the hybrid control algorithms. The algorithm uses force sensor information to arrive at a best estimate of the surface normal and its associated covariance matrix. The results are expected to be best for contact in a very stiff environment. The algorithm may be used to achieve hybrid control even in the unknown environment of the pre-existing nugget landscape.

6.2 Laboratory System Integration

The laboratory demonstration system shown in Figure 1-2 was assembled. This system consists of the robot manipulator and control subsystem, vision/packing subsystem, and wrist/gripper subsystem. This section briefly describes the manipulator control subsystem, and outlines the steps taken to integrate it with the other subsystems, with the goal of achieving a functional RACS.

6.2.1 Interrupt-Driven Control Code Implementation

The control system was implemented on a 133-MHz Pentium computer, using the C++ programming language. The control code is interrupt-driven; a timer card inside the computer generates interrupt requests at fixed time steps, chosen in this case to be 5 milliseconds. The computer system multitasks between two programs: a slow outer loop which handles interaction of the system and the user, and a faster time-critical inner control loop which processes the encoder information and produces an output torque command.

The system runs the slow outer loop until an interrupt request is generated. The outer loop is halted, and the time-critical control loop is executed. When the control loop is finished, the outer loop is resumed. It is therefore guaranteed that the control action occurs when required to ensure a reliable emulation of the continuous control scheme while still allowing user interaction.

Information is passed between the two loops via data latching and semaphore (Ward and Halstead, 1990). Since the outer loop can be interrupted at any time, including while writing data to memory, it is necessary to set up strict guidelines about the validity of data being transferred between the two programs. This is an added complication inherent in multitasking or parallel processing.

This interrupt-driven method contrasts with the simpler single-loop method, where the computer runs only the control loop. Generally, the loop should not last longer than several milliseconds. Displaying information to the computer screen and inputting information from the keyboard takes tens of milliseconds. Thus the single-loop method precludes such interaction during robot control. Therefore, control is limited to pre-determined actions during which the reference trajectory cannot be altered. Also, during interaction with the user, the system cannot be controlled, so the manipulator must either be uncontrolled or a third-party controller must take over.

Since the RACS environment can be damaged rather easily, it is very important to allow user interaction to alter the manipulator's behavior on-line. Therefore the interrupt-driven code was chosen despite its added complexity.

6.2.2 Robot Control Code

The software developed for the laboratory demonstration system consists of the following routines: The homing procedure allows the system to find its absolute joint zero positions. The joint-space PD control mode allows the user to perform gross positioning of each joint. The hybrid control mode performs the core function of the software. The gain-changing routines allow the various position and force control gains to be set on-line. Finally, a data saving routine provides output of data collected during the hybrid control.

Code was written to implement the Cartesian hybrid position/force control scheme. While the system is under control, the current and desired endpoint locations and contact forces are displayed on the screen. Also displayed are the joint locations and the output torque commands.

The user is prompted to choose from a variety of different actions. Firstly, the user can request pure position motions, such as Cartesian straight-line or circular trajectories. These trajectories are smoothed so that the reference acceleration profile is finite and the velocity profile is continuous. Secondly, the user can request hybrid position/force trajectory trials, such as the roller and nugget tests. Finally, the user can also start and stop the collection of data for the data saving routine, and issue a keyboard halt command which zeroes the output torque and serves as a recoverable alternative to the more extreme kill-switch solution.

6.2.3 Integration with Vision/Packing Subsystem

A scanning routine was written in which the end-effector is brought to a point near the scanning camera. The end-effector then passes a grasped nugget over the scanner at a constant specified velocity. The control code is written so that the end-effector starts from rest and accelerates until it reaches the desired speed just as the nugget is at the beginning of the scan position. At this time, nugget profile data is gathered by the vision system (Sujan, 1998).

6.2.4 Incorporation of Gripper/Wrist Subsystem

Control electronics for the wrist were developed so that the computer could receive encoder positions from the three wrist joints and send output current commands to drive the motors. See Appendix D for details. Communications code for the wrist subsystem was incorporated into the control system, and a simple PI controller was implemented to control the orientation of these joints.

6.3 Future Work

Because joint friction plays an important role in the degradation of performance, even with the relatively frictionless AdeptOne manipulator, it may be beneficial to implement a friction compensation scheme such as BaST control (Morel and Dubowsky, 1996) to remove the effects of friction at the low speeds required by the RACS control system. This would require the installation of a base force/torque sensor under the AdeptOne and the coding of the torque estimation algorithm.

Further research in delicate force manipulation is warranted to guarantee that forces will never exceed the safe range. The effects of sensor noise and cross-talk should be studied in order to determine whether appropriate compensation can realistically be performed.

Encouraging simulation results of the surface estimation algorithm have been obtained. The next logical step in this research would be to perform experiments with the AdeptOne system to determine whether the algorithm could be used in practice to control forces against unknown surfaces. It would also be beneficial to modify the algorithm to include surface estimates provided by the vision and packing system (Sujan, 1998). The realization of this algorithm would require the determination of the end-effector stiffness matrix **K**, and a reasonably tight estimate of the sensor noise. Some of the equations performed during the control loop involve numerical solution, and care should be taken to employ an algorithm which converges rapidly and accurately.

It may be desired that the controller perform various position and force control tasks. The control code as it stands can easily be extended to include other families of reference position and force trajectories.

One conceptually simple change to the controller code which would require extensive programming time would be the incorporation of the three wrist joints into the hybrid position/force controller. This would involve the extension of the endpoint vector to seven elements, which represent the position and orientation of the grasped nugget as well as an additional degree of freedom relating to the orientation of gripper itself. This extra degree of freedom is necessary to ensure that the end-effector does not hit the crucible wall before the nugget does. See (Leier, 1998) for details.

Extending the endpoint vector to seven elements will require the solution of forward and inverse kinematics, and the numerical calculation of the 7×7 Jacobian matrix and its inverse. In addition, all of the code must be updated so that the linear algebra and data storage routines accommodate the longer vectors. However, once this work is done, the control system would be more integrated and the reference position and force profiles could involve more elaborate trajectories than are currently possible.

Finally, a major aspect of this project which remains to be completed is the full integration of the three subsystems into a single RACS system. This system should be capable of acquiring and scanning nuggets, locating the optimal placement location within the crucible, delicately placing the wall and crown nuggets, and manipulating the bulk filling device. Once this is achieved, the unified RACS system should be able to successfully charge a crucible, demonstrating by example its technical feasibility.

References

Adept Technology, Inc. “AdeptOne Robot Product Fact Sheet,” *Technical Specifications Bundling,* pp. 48-51, February 1997

Bach, J., Herard, S., Hermann, J.P., Michard, G., and Pardo, P., “Bin-Picking Using a 3-D Sensor and a Special Gripper,” *Proceedings of the 5th International Conference on Robot Vision and Sensory Controls,* pp. 529-540, October 1985

Beckwith, T., Marangoni, R., and Lienhard, J., *Mechanical Measurements,* Addison-Wesley, Reading, USA 1993

Cutkosky, M., *Robotic Grasping and Fine Manipulation*, Kluwer Academic Publishers, Boston, USA 1985

Craig, J., *Introduction to Robotics,* Addison-Wesley, New York, USA 1989

Dietze, W., Keller, W., and Mühlbauer A., “Float-Zone Grown Silicon,” *Crystals, Growth and Applications,* Vol 5., Springer-Verlag, New York, USA 1981, p. 6

Dubowsky, S., “Robot Assisted Crucible Charging System Two Year Progress Report,” MIT Internal Report, 1997

Eppinger, S., and Seering, W., “Understanding Bandwidth Limitations in Robot Force Control,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 904-9, 1987

Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, USA, 1974

Goldenberg, A., Laniado, I., Kuzan, P., and Zhou, C., “Control of Switched Reluctance Motor Torque for Force Control Applications,” *IEEE Transactions on Industrial Electronics,* Vol. 41, No. 4, pp. 461-466, August 1994

Hogan, N., "Impedance Control: An Approach to Manipulation," *Proceedings of the American Control Conference*, pp. 304-13, June 1984

Hogan, N., "Stable Execution of Contact Tasks using Impedance Control," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 2, pp. 1047-1054, April 1987

Idris, H., *Real-time Control of a Six Degree-of-Freedom Vehicle Emulation System*, S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA 1992

Leier, A., *The Grasping and Manipulation of Irregular Objects with Application to Semiconductor Industry,* S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA 1998

Leier, A., Private Correspondence, 1998-2

Li, Y.F., "A Sensor-Based Robot Transition Control Strategy," *International Journal of Robotics Research*, Vol. 15, No. 2, pp. 128-136, April 1996

Liu, G.J. and Goldenberg, A.A., "Robust Hybrid Impedance Control of Robot Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 287-292, April 1991

Maples, J. and Becker, J., "Experiments in Force Control of Robotic Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.695-702, April 1986

Mills, J.K., "Simultaneous Control of Robot Manipulator Impedance and Generalized Force and Position," *Journal of Mechanisms and Machine Theory,* Vol. 31, No. 8, pp. 1069-1080, 1996

Morel, G. and Dubowsky, S., "The Precise Control of Manipulators with Joint Friction: A Base Force/Torque Sensor Method," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol.1, pp. 360-365, 1996

Muto, S. and Shimokura, K., "Accurate Contact Point Detecting Using Force and Velocity Information Complementarily," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 1, pp. 738-744, 1993

Neal, M.J., Rowland, J.J. and Lee, M.H., "A Behaviour-based Approach to Robotic Grasp Formulation: Experimental Evaluation in a Food Product Handling Application," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 304-309, April 1997

Niemeyer, G. and Slotine, J.-J.E., "A Simple Strategy for Opening an Unknown Door," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 2, pp 1448-1453, April 1997

Raibert, M.H. and Craig, J.J., "Hybrid Position/Force Control of Manipulators," *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control,* Vol. 103, No. 2, pp. 126-133, June 1981

Seraji, H., Lim, D., and Steele, R., "Experiments in Contact Control," *Journal of Robotic Systems*, Vol. 13, No. 2, pp. 53-73, 1996

Shackelford, J.F., Alexander U., and Park J.S., eds., *CRC Materials Science and Engineering Handbook 2nd Edition,* CRC Press, Cleveland, USA 1994

Sujan, V., *The Sensor Based Manipulation of Irregularly Shaped Objects with Special Application to the Semiconductor Industry,* S.M. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA 1998

Sujan, V., Private Correspondence, 1998-2

Volpe, R. and Khosla, P., "An Analysis of Manipulator Force Control Strategies Applied to an Experimentally Derived Model," *Proceedings of the IEEE / RSJ International Conference on Intelligent Robots and Systems*, pp. 1989-1997, July 1992

Ward, S.A. and Halstead, R.H. Jr., *Computation Structures*, MIT Press, Cambridge, USA 1990

Wolf, T. and Tauber, R.N., *Silicon Processing for the VLSI Era, Volume 1: Process Technology*, Lattice Press, Sunset Beach, USA 1986

Whitney, D., "Historical Perspective and State of the Art in Robot Force Control," *International Journal of Robotics Research,* Vol. 6, No. 1, pp. 3-14, Spring 1987

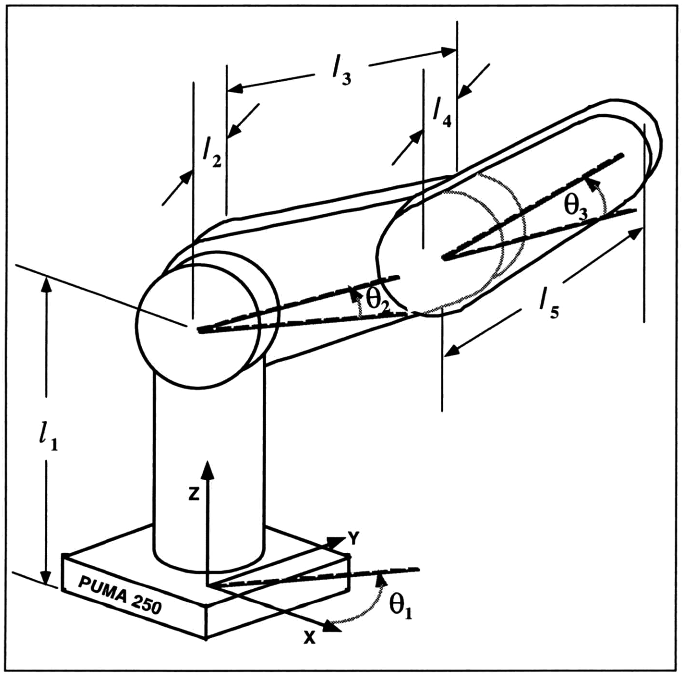
Youcef-Toumi, K. and Gutz, D.A., "Impact and Force Control," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 410-16, 1989

Appendix A

Manipulator Kinematics

The following is a presentation of the kinematic properties of the robotic manipulators mentioned in Section 3.1.1 and Section 3.2.1.

A.1 The Puma 250 Manipulator



$l_1 = 0.3429$ m, $l_2 = 0.0660$ m, $l_3 = 0.2032$ m,

$l_4 = -0.0550$ $l_5 = 0.3400$ m,

Figure A-1: Puma 250 Kinematics

Figure A-1 depicts the kinematic structure of the Puma 250 Manipulator. The Puma is a vertically articulated manipulator with three rotational degrees of freedom. Joint angle θ_1 represents a rotation of the entire manipulator about the world Z axis. Joint angle θ$_2$ represents a rotation of the upper arm about a horizontal axis. Joint angle θ$_3$ represents the relative rotation of the forearm about the elbow. The characteristic link widths and lengths are represented by the values l_1 through l_5.

Given these geometrical properties of the manipulator, the endpoint $[x_{end}, y_{end}, z_{end}]^T$ can be located with the following equations:

$$\begin{aligned} x_{end} &= l_5 \cos\theta_1 \cos(\theta_2+\theta_3) + l_3 \cos\theta_1 \cos\theta_2 + (l_2+l_4)\sin\theta_1, \\ y_{end} &= l_5 \sin\theta_1 \cos(\theta_2+\theta_3) + l_3 \sin\theta_1 \cos\theta_2 - (l_2+l_4)\cos\theta_1, \\ z_{end} &= l_1 + l_5 \sin(\theta_2+\theta_3) + l_3 \sin\theta_2 \end{aligned} \qquad \textbf{(A - 1)}$$

Differentiating both sides of (A - 1) with respect to time yields the expression:

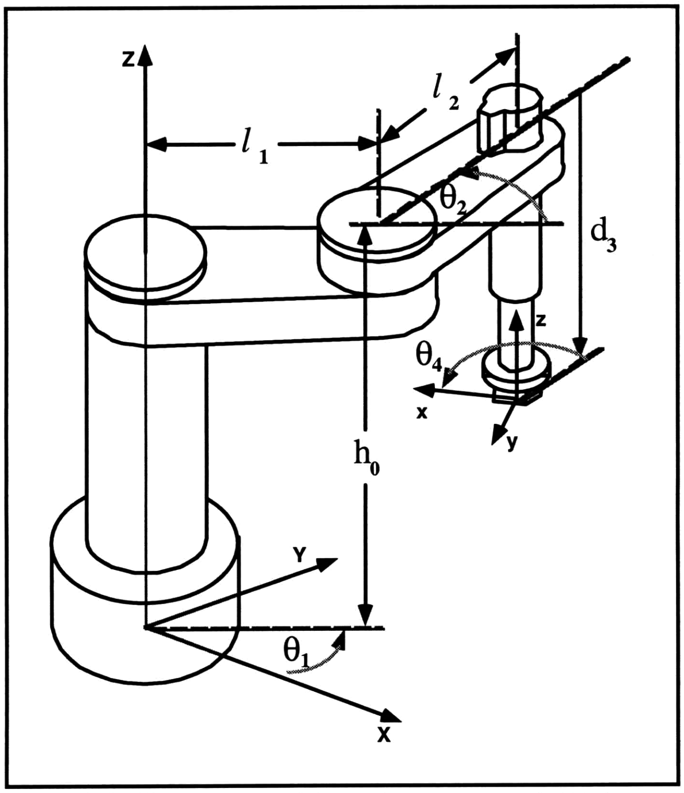
$$\begin{bmatrix} \dot{x}_{end} \\ \dot{y}_{end} \\ \dot{z}_{end} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \qquad \textbf{(A - 2)}$$

where **J** is the manipulator Jacobian matrix. Using the shortened notation $s_i \equiv \sin\theta_i$, $c_i \equiv \cos\theta_i$, $s_{ij} \equiv \sin(\theta_i+\theta_j)$, and $c_{ij} \equiv \cos(\theta_i+\theta_j)$, the manipulator Jacobian is defined as:

$$\mathbf{J} \equiv \begin{bmatrix} -s_1(c_{23}l_5 + c_2l_3) + c_1(l_2+l_4) & -s_{23}c_1l_5 - s_2c_1l_3 & -s_{23}c_1l_5 \\ c_1(c_{23}l_5 + c_2l_3) + s_1(l_2+l_4) & -s_{23}s_1l_5 - s_2s_1l_5 & -s_{23}s_1l_5 \\ 0 & c_2l_3 + c_{23}l_5 & c_{23}l_5 \end{bmatrix}. \qquad \textbf{(A - 3)}$$

It is this matrix which is necessary for the implementation of Jacobian Transpose and Jacobian Inverse control.

A.2 The Adept One Manipulator



l_1 = 0.4250 m, l_2 = 0.03775 m, h_0 = 0.8770 m,

Figure A-2: Adept One Kinematics

Figure A-2 depicts the kinematic structure of the Adept One Manipulator, as described in (Adept, Inc., 1997). The Adept is a SCARA manipulator with three rotational degrees of freedom and one prismatic degree of freedom. Joint angle θ_1 represents a rotation of link 1 about the world Z axis. Joint angle θ_2 represents a relative rotation of link 2 about the same axis. Prismatic joint position d_3 represents the extension and retraction of the prismatic link in the negative Z direction. Finally, joint angle θ_4 represents the rotation of the final link about the Z axis. In actuality, the last joint is a screw joint which translates as well as rotates. However, the parameters can be written so that this translation is included in the motion of the prismatic joint. The characteristic link lengths and heights are represented by the values l_1, l_2 and h_0.

Given these geometrical properties of the manipulator, the endpoint position and orientation $[x_{end}, y_{end}, z_{end}, \phi_{end}]^T$ can be located with the following equations:

$$\begin{aligned} x_{end} &= l_1 \cos\theta_1 \ + l_2 \cos(\theta_1 + \theta_2), \\ y_{end} &= l_1 \sin\theta_1 \ + l_2 \sin(\theta_1 + \theta_2), \\ z_{end} &= h_0 - d_3, \\ \phi_{end} &= \theta_1 + \theta_2 + \theta_4 . \end{aligned} \tag{A - 4}$$

Differentiating both sides of (A - 4) with respect to time yields the expression:

$$\begin{bmatrix} \dot{x}_{end} \\ \dot{y}_{end} \\ \dot{z}_{end} \\ \dot{\phi}_{end} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \end{bmatrix}, \tag{A - 5}$$

where **J** is the manipulator Jacobian matrix. Again, using the shortened notation $s_i \equiv \sin\theta_i$, $c_i \equiv \cos\theta_i$, $s_{ij} \equiv \sin(\theta_i + \theta_j)$, and $c_{ij} \equiv \cos(\theta_i + \theta_j)$, the manipulator Jacobian is defined as:

$$\mathbf{J} \equiv \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}. \tag{A - 6}$$

It is this matrix which is necessary for the implementation of Jacobian Transpose and Jacobian Inverse control. Notice that the third row and column contain only a negative one along the main diagonal. This indicates that the motion of the third joint is completely decoupled from the that of rest of the manipulator; also, the dynamics of the prismatic joint should not depend on nor affect that of any other joint.

Appendix B

Material Properties Information of Silicon and Glass

The following information was obtained from various handbooks (Shackelford *et al.,* 1994; Dietze *et al.,* 1981); these values can be used as guidelines to simulate and characterize the behavior of the interaction between silicon and glass.

B.1 Silicon Properties

Crystal structure:	Diamond lattice
Density at 20°C	2.33 g/cm^3
Modulus of Elasticity	150 Gpa
Poisson's Ratio	0.17
Indentation Hardness	1000 kg/mm^2

B.2 Glass Properties

	Window Glass	Fused Silica
Chemical Description	72% SiO_2 14% Na_2O 10% CaO 2% MgO 1% Al_2O_3 1% Impurities	99% SiO_2 1% (or less) Impurities
Density	2180 kg/m^3	2200 kg/m^3
Modulus of Elasticity	69 Gpa	72 GPa
Shear Modulus	30.3 Gpa	29.6 GPa
Poisson's Ratio	0.23	0.17
Indentation Hardness	400-600 kg/mm^2	600-679 kg/mm^2
Impact Abrasion Hardness†	1.0	3.5
Tensile Strength	30-70 Mpa	10 MPa
Compressive Strength	350 Mpa	137 MPa
Working Stress*	6.6 Mpa	6.6 MPa
Coefficient of Expansion	92x10^{-7} /Celsius	56x10^{-7}/ Celsius

*Explicit differences between window glass and fused silica not found

†Glass has its own hardness scale, determined by relative resistance to sandblasting. Window glass is given an arbitrary Impact Abrasion Hardness of 1.0

Appendix C

Material Properties Testing

Several kinds of tests were conducted involving polycrystalline silicon nuggets and window glass. These tests serve to determine the performance specifications required to ensure that the crucible not be damaged by its interaction with nuggets. It is assumed that window glass and fused silica demonstrate similar properties, as shown in Appendix B. The experiments explored the two most probable failure modes, scratching and impact damage.

C.1 Scratch Tests

Tests were performed to determine the force necessary to produce scratches on window glass. A polycrystalline silicon nugget was placed on a sample of window glass, making a three-point contact. The nugget was loaded with known weights and dragged along the glass. The glass samples were inspected under a 66x optical microscope. Scratching was detected at sustained contact forces exceeding 5 Newtons.

C.2 Impact Tests

Polycrystalline nuggets of various masses were dropped onto window glass from various heights in free fall conditions. The glass samples were visually inspected for damage. The following results were obtained:

Mass of Nugget:	Damage Initiated at Velocity:
14 ± 7 grams	0.99 ± .02 m/s
28 ± 7 grams	0.99 ± .02 m/s
85 ± 7 grams	0.63 ± .03 m/s
140 ± 7 grams	0.44 ± .04 m/s
185 ± 7 grams	0.44 ± .04 m/s

It is recommended that the speeds be kept under 50% of these values as a safety factor. If the impact velocities are restricted to less than 20 centimeters per second, there is very little chance of impact damage.

Appendix D

Wrist Control Circuitry

The electronic circuit which interfaces the control computer with the wrist portion of the gripper subsystem is depicted in Figure D-1 (Leier, 1998).

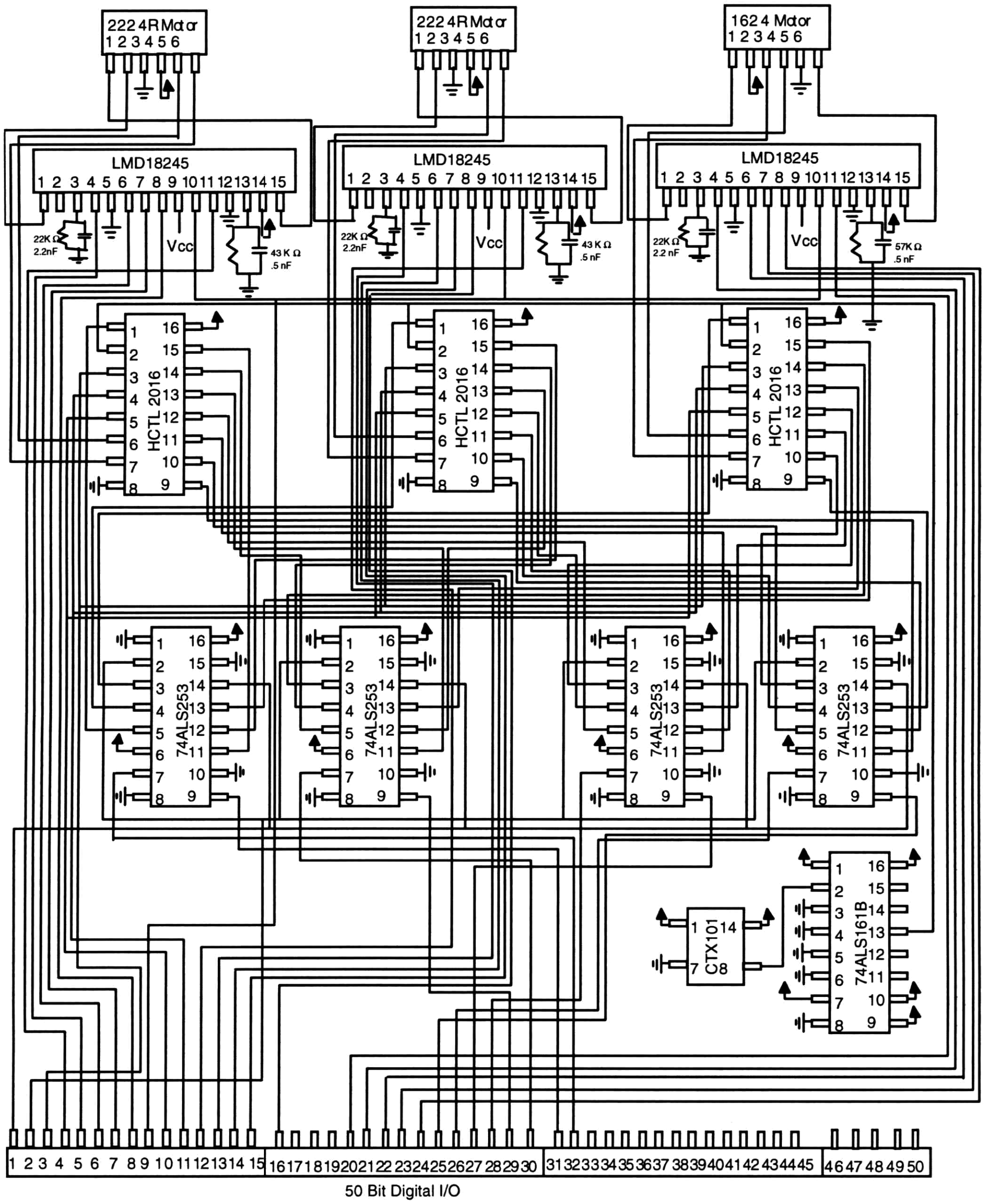


Figure D-1: Wrist Circuit Diagram

It was designed and implemented with the assistance of Anthony Leier, and is detailed along with the rest of the wrist design in (Leier, 1998).

This circuit consists of the following components: three wrist motors with encoders, three LMD18245 DMOS full bridge motor drivers, three HCTL2016 16-bit quadrature decoders, four 74ALS253 multiplexors, one CTX101 1-MHz crystal oscillator, one 74ALS161B binary counter, and one PDIO-74 72-bit programmable digital I/O board which communicates with the PC.

The crystal oscillator clocks the binary counter, which provides a slower clock signal to the HCTL2016 decoders. Quadrature signals from the wrist encoders are fed to the decoders, which store two-byte encoder positions but only output one byte at a time. The three encoder position bytes are then combined into one byte by the multiplexor chips, and this single byte is read by the digital I/O board. Three selector lines are output by the digital I/O board to choose which byte from which decoder to read.

The digital I/O board outputs three digital motor torque command values, which are fed to the LMD18245 motor drivers. The motor drivers convert these signals into high-power current commands which drive the three wrist motors.

This circuit therefore allows the control computer to output torque commands as a function of wrist joint positions. A simple controller was implemented, as mentioned in section 6.2.4.