Technical Report 912

Trajectory and Force Control of a Direct Drive Arm

Chae Hun An

MIT Artificial Intelligence Laboratory

This blank page was inserted to preserve pagination.

Trajectory and Force Control of a Direct Drive Arm

by

Chae Hun An

S.B. (1982) Electrical Engineering and Computer Science
Massachusetts Institute of Technology
S.M. (1983) Electrical Engineering and Computer Science
Massachusetts Institute of Technology

©Massachusetts Institute of Technology 1986

Trajectory and Force Control of a Direct Drive Arm

by

Chae H. An

Submitted to the Department of Electrical Engineering and Computer Science on August 18, 1986 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract. Using the MIT Serial Link Direct Drive Arm as the main experimental device, various issues in trajectory and force control of manipulators were studied in this thesis. Since accurate modelling is important for any controller, issues of estimating the dynamic model of a manipulator and its load were addressed first. Practical and effective algorithms were developed from the Newton-Euler equations to estimate the inertial parameters of manipulator rigid-body loads and links. Load estimation was implemented both on a PUMA 600 robot and on the MIT Serial Link Direct Drive Arm. With the link estimation algorithm, the inertial parameters of the direct drive arm were obtained. For both load and link estimation results, the estimated parameters are good models of the actual system for control purposes since torques and forces can be predicted accurately from these estimated parameters.

The estimated model of the direct drive arm was then used to evaluate trajectory following performance by feedforward and computed torque control algorithms. The experimental evaluations showed that the dynamic compensation can greatly improve trajectory following accuracy.

Various stability issues of force control were studied next. It was determined that there are two types of instability in force control. Dynamic instability, present in all of the previous force control algorithms discussed in this thesis, is caused by the interaction of a manipulator with a stiff environment. Kinematic instability is present only in the hybrid control algorithm of Raibert and Craig, and is caused by the interaction of the inertia matrix with the Jacobian inverse coordinate transformation in the feedback path. Several methods were suggested and demonstrated experimentally to solve these stability problems. The results of the stability analyses were then incorporated in implementing a stable force/position controller on the direct drive arm by the modified resolved acceleration method using both joint torque and wrist force sensor feedbacks.

Thesis Supervisor: Dr. John M. Hollerbach
Associate Professor of Brain and Cognitive Sciences

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Systems Development Foundation and the Defense Advanced Research Projects Agency under Office of Naval Research contracts N00014-80-C-050, N00014-82-K-0334, and N00014-85-K-0124. Support for the author is provided in part by a National Science Foundation graduate fellowship.

Acknowledgments

I thank my advisor John Hollerbach without whom this thesis would not have been possible. He first suggested to me to work on the direct drive arm, and then guided, encouraged, and supported me throughout my thesis work at the MIT Artificial Intelligence Laboratory. His emphasis on experimental results was a very strong and positive motivation for me and became a main theme of my thesis.

I also thank the other members of my committee: Jeff Lang, Tomas Lozano-Perez, and Ken Salisbury. Jeff has been helping and encouraging me in many ways from my undergraduate years at MIT. He especially gave me many insights in understanding motors and control of motors. His courses in control and estimation also helped me greatly in my research. I first learned robotics from Tomas through his excellent course in robot manipulation. During my thesis work, he gave me many helpful comments that often redirected my thinking and my research emphasis. Ken's knowledge and understanding of force control were of great help to me especially during the second half of my thesis work. I had many stimulating discussions with him on issues of force control and he gave me many helpful suggestions in solving some of the problems that I had faced.

All of the members of John's group have been tremendous. I benefitted greatly from working with Chris Atkeson on the load and link estimation projects. Load estimation was his original idea and his valuable insights generated neat solutions and also created many new projects. Chris also gave me much help when I first came to the AI-LAB. Ki Suh did most of the early work with the direct drive arm and got the arm moving when it first came to the AI-LAB. As an office-mate, he also helped me in a number of ways, ranging from programming to giving me rides when needed. Sundar Narasimhan and Dave Siegel are two of the best computer hackers that I know. They developed the MUSE multi-processor system that I used in controlling the direct drive arm. Without their expertise in computers,

my experiments would not have been possible. John Griffiths worked on the first version of the computed torque controller. His help in this is greatly appreciated. Dave Bennet has been putting up with me as an office-mate. Also, he gave me some valuable insights on parts of my research.

There are many other people that have contributed to my thesis. Ron Wiken, Noble Larson, John Purbrick, Scott Jones, Harry Voorhees, and many others at the AI-LAB provided me with various helps. I used Harry's plotting software extensively in my thesis. Thanks to all the people of the Artificial Intelligence Laboratory who made it such a great place to work at.

Members of the Gate Bible Study at MIT have supported me by their friendship and prayers. My parents have supported and encouraged me throughout the years. My parents-in-law have always expressed confidence in me and encouraged me.

I don't know how to thank my wife, Heikyung, who has been just WONDERFUL. She encouraged me, supported me both emotionally and financially, and took care of me in many many ways. Thanks, Love!

Contents

1 Introduction **10**

1.1 Direct Drive Arm 10

1.2 Objectives 12

1.3 Literature Survey 14

1.4 Overview of the Thesis 16

2 Estimation of Load Inertial Parameters **18**

2.1 Introduction 18

2.2 The Newton-Euler Approach To The Load Identification Problem . . 20

2.2.1 Deriving The Parameter Equation 20

2.2.2 Estimating The Parameters 25

2.2.3 Recovering Object And Grip Parameters 25

2.3 Experimental Results 26

2.3.1 Estimation on the PUMA Robot 26

2.3.2 The MIT Serial Link Direct Drive Arm 33

2.4 Discussion 35

2.4.1 Usefulness of the Algorithm 35

2.4.2 Sources of Error 36

2.4.3 Inaccurate Estimates of the Moments of Inertia 39

2.5 Conclusion 40

3 Estimation of Link Inertial Parameters **41**

3.1 Introduction 41

3.1.1 Previous Work 42

3.2 Estimation Procedure 44

3.2.1 Formulation of Newton-Euler Equations 44

3.2.2 Estimating the Link Parameters 47
3.3 Experimental Results 49
3.4 Discussion 53
3.4.1 Identifiability of Inertial Parameters 55

4 Feedforward and Computed Torque Control 57
4.1 Introduction 57
4.1.1 Control Algorithms 58
4.2 Robot Controller Experiment 60
4.2.1 Feedforward Controller 60
4.2.2 Computed Torque Controller 64
4.3 Conclusions 66

5 Dynamic Stability Issues in Force Control 67
5.1 Introduction 67
5.2 Stability Problems 69
5.2.1 General Stability Analysis 69
5.2.2 Example of Unmodelled dynamics 73
5.2.3 Experimental Verification of Instability 73
5.3 Compliant Coverings 78
5.4 Adaptation to the Environment Stiffness 78
5.4.1 Least Squares 80
5.4.2 Adaptive Observer 85
5.4.3 Feasibility 86
5.5 Joint Torque Control 87
5.5.1 Dominant Pole 88
5.5.2 One Link Force Control Experiments 91

6 Kinematic Stability Issues in Force Control 103
6.1 Introduction 103
6.2 Intuitive Stability Analysis for Revolute Manipulators 108
6.2.1 Hybrid Control of Raibert and Craig 108
6.2.2 Stiffness Control of Salisbury 111
6.2.3 Summary of the intuitive analytical results 112
6.3 Root Loci, Simulations, and Experiments for Revolute Manipulators 113

6.3.1 Hybrid Control 114
6.3.2 Resolved Acceleration Force Control 121
6.3.3 Stiffness Control 125
6.4 Stability for Polar Manipulators 127
6.5 Conclusion on Kinematic Stability 130

7 Force Control Experiments on the Direct Drive Arm 133
7.1 Experimental Setup 133
7.2 Experimental Results 135
7.2.1 Square Wave Force Input 135
7.2.2 Sine Wave Force Input 136
7.2.3 Following the Eccentric Cam 136
7.3 Some Results with Stiffness Control 139

8 Conclusion and Future Research 143
8.1 Conclusion 143
8.2 Future Research 144
8.2.1 Manipulator 144
8.2.2 Collision 145
8.2.3 Fundamental Concepts of Compliant Behavior 145

Appendices 148
Appendix 1: Closed Form Dynamics 148
Appendix 2: Stability Robustness 150
Appendix 3: Adaptive Observer 152
Appendix 4: Operational Space and Resolved Acceleration 153

References 155

Chapter 1

Introduction

The control of robot manipulators has become an important subject of study due to growing interests and uses of robot manipulators. Unfortunately, very few of the proposed control algorithms have ever been implemented on an actual manipulator, and it is not clear how good or practical these algorithms really are. This lack of experimental results is mainly due to the lack of a high quality manipulator that can benefit from sophisticated control algorithms. The motivation of this thesis is to evaluate some of these control algorithms (both trajectory and force control) using a high quality direct drive arm and to understand some of the problems in robot control that researchers have observed in the past. Then, some new methods of control are developed to overcome those problems.

1.1 Direct Drive Arm

A typical industrial manipulator such as the PUMA (Unimation, Inc.) has small actuators at the joints and utilizes very large gear ratios in order to be able to exert enough torque to the links. This arrangement of actuation introduces a large amount of undesirable nonlinearities such as friction and backlash at the joints. In fact, for the PUMA 600 manipulator at the MIT Artificial Intelligence Laboratory, it was measured that the friction terms account for as much as 50% of the motor torques. Ironically, since these nonlinear effects are difficult to deal with, most of the proposed control algorithms are based on the rigid body dynamic model of the

robot, neglecting the non-ideal characteristics. Therefore, these algorithms cannot be used on typical manipulators effectively. In fact, the effects of the rigid body dynamics of the links are very small for these highly geared manipulators since these effects are reduced by the square of the gear ratio. A typical gear ratio of 100 : 1 would then reduce the dynamic effects of the links by 10^{-4}. As a result, the dynamics of a conventional robot are dominated by the motor inertias and the joint frictions (Goor, 1985; Good, Sweet, and Strobel, 1985). This is the main reason that the most common form of controller for these manipulators are the independent joint PID controllers.

Conventional robots have other disadvantages. Because of their characteristics mentioned above, they are in general slow, and cannot be used in high speed applications such as laser cutting (Youcef-Toumi, 1985). Also, they are essentially positioning devices and are not suitable for controlling interaction forces at the tip of the manipulator since the actuators with high gear ratios cannot be used to command torques effectively. Typically the best way to control these manipulators is to implement tight position loops at the joints, thus making the actuators into position sources.

Recently, however, several high quality direct drive arms have been developed in order to overcome some of the performance limitations of the conventional robots (Asada and Kanade, 1981; Asada, Kanade and Takeyama, 1983; Asada and Youcef-Toumi, 1984; Curran and Mayer, 1985; Kuwahara, Ono, Nikaido, and Matsumoto, 1985) . Since the links are directly coupled to the motors, the backlash effects are eliminated and the joint frictional effects are reduced immensely. Therefore, the dynamics of these direct drive arms are modelled accurately by the rigid body dynamics of the links. This characteristic makes these manipulators not only more suitable to test the recent control algorithms based on link dynamics, but also makes it necessary to use such sophisticated control algorithms since the full coupled dynamics of the links are reflected directly to the actuators.

Without the high gearing and the undesirable nonlinear effects at the joints, the control of joint torques also becomes more feasible. Since the actuators can be treated as torque sources, they are more suitable for controlling forces and torques at the tip of the manipulator.

But, there are also some drawbacks with the direct drive arm technology. Since there is no gearing to amplify the motor torques, the motors have to be large to

be able to exert large torques. This makes the whole manipulator large and more difficult to control. Also, since the motors have to exert large torques, large currents flow through the windings, overheating the motors quickly. Another drawback is that the manipulator dynamics will be sensitive to the changes in loads at the tip of the manipulator, since the load inertial effects are fully reflected to the joints without any reduction through the gears. Despite these drawbacks, the ability to model the manipulator accurately by the ideal rigid body dynamics makes these manipulators very attractive for control studies and for high performance applications.

The main experimental device used in this thesis is the MIT Serial Link Direct Drive Arm (DDARM) (Fig. 1.1), developed by Haruhiko Asada while he was at MIT. It is a three link manipulator with a three phase rare-earth permanent magnet brushless DC motor placed at each joint. The serial link configuration of this manipulator differentiates this direct drive arm from another direct drive arm at MIT, which was also developed by Asada but has parallel linkages. The motor charateristics of the DDARM are listed in Table 1.1 (Youcef-Toumi, 1985).

	motor dia. (cm)	motor mass (kg)	peak torque (Nm)	rotor inertia ($kg \cdot m^2$)	# poles	max. current (Amp) instantaneous	max. current (Amp) continuous
Joint 1	35	20.39	660	0.181	30	50	15
Joints 2 & 3	25	16.5	230	0.0256	18	30	10

Table 1.1: Motor characterisitics for the Direct Drive Arm

1.2 Objectives

As mentioned in the first paragraph, a goal of this thesis is to study robot control using the DDARM. The first step in any control design is the accurate modelling of the plant to be controlled. In practice, especially with the availability of automatic control design tools, this modelling step may occupy greater than 90% of the control designer's efforts. Hence, for controlling a direct drive arm, accurate modelling of the manipulator is important. Since the actuators are inherently parts of the links for direct drive arms, separate modelling of the mechanical properties of the actuators are not necessary. The electrical dynamics of the actuators are often

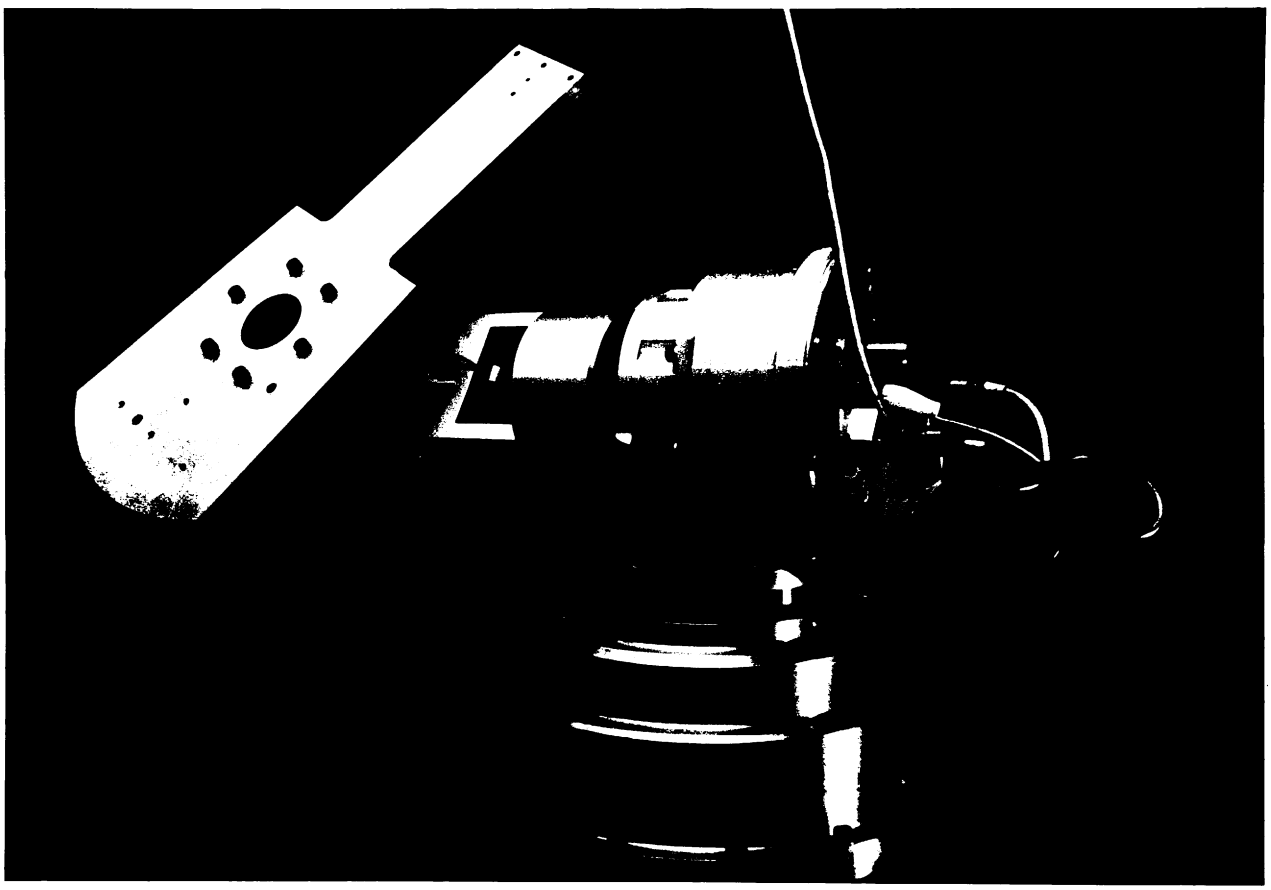


Figure 1.1: MIT Serial Link Direct Drive Arm

orders of magnitude faster than the inertial dynamics and it may not be necessary to include them in the robot model.

Kinematic parameters are usually well known or can be calibrated using methods developed by Whitney, Lozinski, and Rourke (1986) and others. However, the inertial parameters, i.e. the mass, the location of center of mass, and the moments of inertia of each rigid body link of a robot are usually not known even to the manufacturers of the robots. Also, even if the links were modelled accurately, the inertial parameters of the loads vary with different loads. Since a load is essentially a part of the last link, the knowledge of the inertial parameters of manipulator loads is also important for accurate control of manipulators. Therefore, one of the objectives of this thesis is to develop a practical algorithm to estimate the inertial parameters of links and loads accurately. The second objective is to use the estimated parameters in the design of trajectory and force control algorithms for the direct drive arm and evaluate experimental results using different controllers.

Between the area of trajectory control and the area of force control of manipulators, the issues of force control are much less understood. As mentioned before, conventional robots are not well suited for implementating force control algorithms since they are essentially positioning devices. Therefore, previous implementations seldom produced satisfactory results (Caine, 1985). In fact, researchers in the past have experienced significant instability problems associated with force controllers (Whitney, 1985). Therefore, another goal of this thesis is to understand some of the stability and performance problems associated with force control, and suggest and demonstrate some remedies to those problems using the direct drive arm.

As stressed several times already, the implementation aspects are important parts of my thesis work since very few experimental results are available in the field of robot control. Experiments are important not only because they can verify the theory but also some practical problems and insights may be discovered during implementations that may not have been obvious at first. This was true throughout my thesis work.

1.3 Literature Survey

This section is brief since each chapter of this thesis, except for Chapters 1, 7, and 8, contains introduction section which includes the survey of previous works.

The direct drive arm technology has been pioneered in the U.S. by Haruhiko Asada. His first version was developed at the Carnegie-Mellon University (Asada and Kanade, 1981; Asada, Kanade, and Takeyama, 1983). At MIT, Asada, Youcef-Toumi (1985), and Ro (Asada and Ro, 1985) developed the invariant inertia method for designing direct drive arms.

Since the importance of modelling the manipulator accurately has been increasingly apparent with the demand for high performance controller, several investigators have recently suggested various algorithms for estimating the kinematic and the dynamic models. Whitney (1986) has presented the most comprehensive work so far for calibrating the kinematic parameters using measurements with theodolites. His work includes both theoretical and experimental results. Other works on kinematic identification are by Wu (1983), Hayati (1983), Mooring (1983) , and Sugimoto and Okada (1984).

Paul (1981), Coiffet (1983), Olsen and Bekey (1985), and Mukerjee and Ballard (1985) have studied the rigid-body load identification problem, and Mayeda, Osuka, and Kangawa (1984), Olsen and Bekey (1985), Mukerjee and Ballard (1985), Newman and Khosla (1985) and Khalil, Gautier, and Kleinfinger (1986) have studied the rigid-body link identification problem. Some of the works by the above authors are similar to the algorithms presented in this thesis, but very few have been verified by implementation. In addition, Cannon and Schmitz (1984) and Book (1984) among other researchers have considered modelling flexible modes of links.

There have been many control algorithms based on the rigid body dynamic model of a manipulator. The computed torque control method for trajectory control has been studied by various researchers (Paul, 1972; Markiewicz, 1973; Bejczy, 1974; Luh, Walker, and Paul,1980b, Gilbert and Ha, 1984). However, the actual implementation of such controller has not been reported until very recently. The only experimental result that has been published is by Khosla and Kanade (1986), who at CMU also used a direct drive arm. The feedforward control for manipulators was suggested by Liegeois, Fournier, and Aldon (1980), and Asada, Kanade, and Takeyama (1983) reported some results of feedforward control implementation on their direct drive arm.

In the study of force control, there has been much effort since the late 1970's. The report by Whitney (1985) includes an extensive survey of force control for manipulators. Some of the studies have focused on passive compliance devices (Drake

and Simunovic, 1977). For active force control, there are mainly four continuous feedback methods that have been proposed and implemented with various degree of success by researchers. They include:

- damping control (Whitney, 1977),
- stiffness control (Salisbury, 1980),
- impedance control (Hogan, 1985a, 1985b, 1985c),
- hybrid force/position control (Raibert and Craig, 1981; Khatib, 1983; Khatib and Burdick, 1986).

The above force control implementations generally had difficulty in dealing with stiff environments. Recently, Whitney (1985), Roberts, Paul, and Hillberry (1985), Wlassich (1986), and Eppinger and Seering (1986) have addressed stability problems for force feedback algorithms using wrist force sensors. In some instances, the work reported in this thesis is similar to their work, but includes more complete analyses, verification by actual experiments, and some novel approaches in remedying the stability problems.

There has been much discussion on the singularities of the Jacobian inverse often used in force control algorithms (Whitney, 1972). But there has been no previous work in considering the instabilities caused by the coordinate transformations at places other than the kinematic singularity points. To the best of my knowledge, the results reported in this thesis on the kinematic instabilities of some force control methods are the first in this area.

1.4 Overview of the Thesis

In Chapter 2,[1] the estimation algorithm for a load is presented along with experimental results on the PUMA and the DDARM. The algorithm is based on the reformulation of the Newton-Euler rigid body dynamics of a load such that the resulting equations are linear in terms of the unknown inertial parameters. The load identification algorithm is extended in Chapter 3 to identify all of the inertial

[1]The work reported in Chapters 2, 3, and 4 results from joint effort with Chris Atkeson, John Hollerbach, and John Griffiths.

parameters of the links of any manipulator whose torque or force at each joint can be measured. The algorithm is implemented successfully to identify the link inertial parameters of the DDARM. The estimated parameters are used in the feedforward and the computed torque control algorithms and their performances are compared in Chapter 4 against the performance of a simple PD controller.

Chapters 5 and 6 deal with the stability problems of force control. The dynamic instabilities, observed in a force controlled manipulator in contact with a stiff environment, are studied in Chapter 5. Analytical results using a simple model of a manipulator are presented and verified by experiments. Then, some methods of solving the instability problems are suggested and demonstrated on the third link of the DDARM. In Chapter 6, another type of instability, associated with some force control methods, is studied. Three different force control methods are considered and it is shown both by analyses and by experiments that the hybrid control method of Raibert and Craig (1981) exhibit kinematically induced instabilities whereas the other methods do not.

In Chapter 7, the results of the estimation, the trajectory control, and the dynamic and kinematic instability analyses are combined in implementing a stable force/position controller on the two joints of the DDARM. Finally, conclusions and recommendation for future work are presented in Chapter 8.

Chapter 2

Estimation of Load Inertial Parameters

2.1 Introduction

This chapter presents a method of estimating all of the inertial parameters of a rigid body load using a wrist force/torque sensor: the mass, the moments of inertia, the location of its center of mass, and the object's orientation relative to a force sensing coordinate system. This procedure has three steps:

1. A Newton-Euler formulation for the rigid body load yields dynamics equations linear in the unknown inertial parameters, when the moment of inertia tensor is expressed about the wrist force sensing coordinate system origin.

2. These inertial parameters are then estimated using a least squares estimation algorithm.

3. The location of the load's center of mass, its orientation, and its principal moments of inertia can be recovered from the sensor referenced estimated parameters.

In principle, there are no restrictions on the movements used to do this load identification, except that if accurate estimation of all the parameters is desired the

This chapter is a revised version of (Atkeson, An, and Hollerbach, 1985b)

motion must be sufficiently rich (i.e. occupy more than one orientation with respect to gravity and contain angular accelerations in several different directions) and sometimes special test movements must be used to get accurate estimates of moment of inertia parameters.

There are several applications for this load identification procedure. The accuracy of path following and general control quality of manipulators moving external loads can be improved by incorporating a model of the load into the controller, as the effective inertial parameters of the last link of the manipulator change with the load. The mass, the center of mass, and the moments of inertia constitute a complete set of inertial parameters for an object; in most cases, these parameters form a good description of the object, although they do not uniquely define it. The object may be completely unknown at first and an inertial description of the object may be generated as the robot picks up and moves the object. The robot may also be used in a verification process, in which the desired specification of the object is known and the manipulator examines the object to verify if it is within the tolerances. Recognition, finding the best match of a manipulated object to one among a set of known objects, may also be desired. Finally, the estimated location of the center of mass and the orientation of the principal axis can be used to verify that the manipulator has grasped the object in the desired manner.

A key feature of the approach in this thesis is that it requires no special test or identification movements and therefore can continuously interpret wrist force and torque sensory data during any desired manipulation. Previous methods of load identification were restricted in their application. Paul (1981) described two methods of determining the mass of a load when the manipulator is at rest, one requiring the knowledge of joint torques and the other forces and torques at the wrist. The center of mass and the load moments of inertia were not identified.

Coiffet (1983) utilized joint torque sensing to estimate the mass and center of mass of a load for a robot at rest. Moments of inertia were estimated with special test motions, moving only one axis at a time or applying test torques. Because of the intervening link masses and domination of inertia by the mass moments, joint torque sensing is less accurate than wrist force-torque sensing.

Olsen and Bekey (1985) assumed full force-torque sensing at the wrist to identify the load without special test motions. Mukerjee's approach (1984, 1985), which also allows general motion during load identification, is similar to the approach taken in

this chapter. Nevertheless, neither paper simulated or experimentally implemented their procedures to verify the correctness of the equations or to determine the accuracy of estimation in the presence of noise and imperfect measurements.

Khalil, Gautier, and Kleinfinger (1986) have suggested a method of identifying the load inertial parameters using joint torque sensing as a part of their link identification procedure. As mentioned before, joint torque sensing is less accurate than wrist force-torque sensing. Also, they did not present any simulation or experimental results.

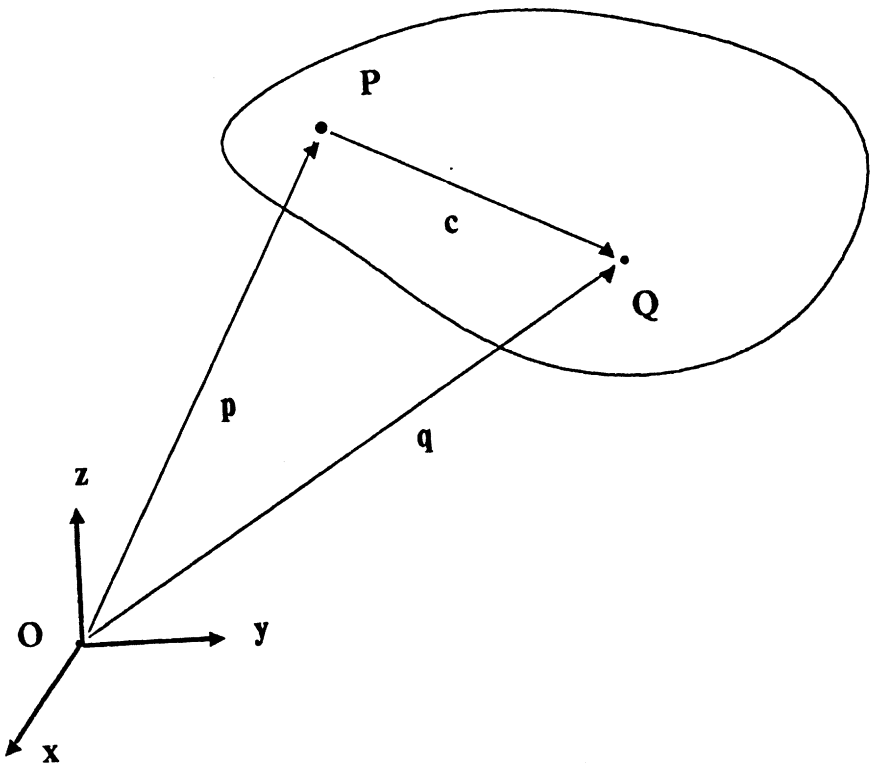
The algorithm to be presented requires measurements of the force and torque due to a load and measurements or estimates of the position, velocity, acceleration, orientation, angular velocity, and angular acceleration of the force sensing coordinate system. It can handle incomplete force and torque measurement by simply eliminating the equations containing missing measurements. The necessary kinematic data can be obtained from the joint angles and, if available, the joint velocities of the manipulator. Also, the inertial parameters of a robot hand can be identified using this algorithm and then the predicted forces and torques due to the hand can be subtracted from the sensed forces and torques, so that only the load is estimated.

This inertial parameter estimation algorithm was implemented using a PUMA 600 robot equipped with an RTI FS-B wrist force/torque sensor, and on the MIT Serial Link Direct Drive Arm (DDARM) equipped with a Barry Wright Company Astek FS6-120A-200 6-axis wrist force/torque sensor.

2.2 The Newton-Euler Approach To The Load Identification Problem

2.2.1 Deriving The Parameter Equation

To derive equations for estimating the unknown inertial parameters, the coordinate systems in Figure 2.1 are used to relate different coordinate frames and vectors. **O** is an inertial or base coordinate system, which is fixed in space with gravity pointing along the $-z$ axis. **P** is the force reference coordinate system of a wrist force/torque sensor rigidly attached to the load. **Q** represents the principal axis of the rigid body load located at the center of mass. The x axis of **Q** is along



p: position vector from the origin of the base coordinate frame to the origin of the wrist sensor coordinate frame.

q: position vector from the origin of the base coordinate frame to the center of the mass of the load.

c: position vector from the origin of the wrist sensor coordinate frame to the center of the mass of the load.

Figure 2.1: Coordinate Frames.

the largest principal moment of inertia, and the z axis along the smallest. **Q** is unique up to a reflection in bodies with 3 distinct principal moments of inertia. In the derivation that follows all vectors are initially expressed in the base coordinate system **O**.

The mass, moments of inertia, location of the center of mass, and orientation of the body (a rotation ${}_{QP}\mathbf{R}$ from the principal axes to the force reference system) are related to the motion of the load and the forces and torques exerted on it by the Newton-Euler equations. The net force ${}_{q}\mathbf{f}$ and the net torque ${}_{q}\mathbf{n}$ acting on the load at the center of mass are:

$$ {}_{q}\mathbf{f} = \mathbf{f} + m\mathbf{g} = m\ddot{\mathbf{q}} \tag{2.1} $$

$$ {}_{q}\mathbf{n} = \mathbf{n} - \mathbf{c} \times \mathbf{f} = {}_{q}\mathbf{I}\dot{\omega} + \omega \times ({}_{q}\mathbf{I}\omega) \tag{2.2} $$

where:

- $\mathbf{f}$ = the force exerted by the wrist sensor on the load at the point $\mathbf{p}$,
- m = the mass of the load,
- $\mathbf{g}$ = the gravity vector ($\mathbf{g} = [0, 0, -9.8\ \text{meters/sec}^2]$),
- $\ddot{\mathbf{q}}$ = the acceleration of the center of mass of the load,
- $\mathbf{n}$ = the torque exerted by the wrist sensor on the load at the point $\mathbf{p}$,
- $\mathbf{c}$ = the unknown location of the center of mass relative to the force sensing wrist origin $\mathbf{P}$,
- ${}_{q}\mathbf{I}$ = the moment of inertia tensor about the center of mass,
- ω = the angular velocity vector, and
- $\dot{\omega}$ = the angular acceleration vector.

To formulate an estimation algorithm, the force and torque measured by the wrist sensor must be expressed in terms of the product of known geometric parameters and the unknown inertial parameters. Although the location of the center of mass and hence its acceleration $\ddot{\mathbf{q}}$ are unknown, $\ddot{\mathbf{q}}$ is related to the the acceleration of the force reference frame $\ddot{\mathbf{p}}$ by

$$\ddot{\mathbf{q}} = \ddot{\mathbf{p}} + \dot{\boldsymbol{\omega}} \times \mathbf{c} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{c}) \qquad (2.3\ [7.40]^1)$$

Substituting (2.3) into (2.1),

$$\mathbf{f} = m\ddot{\mathbf{p}} - m\mathbf{g} + \dot{\boldsymbol{\omega}} \times m\mathbf{c} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times m\mathbf{c}) \qquad (2.4)$$

Substituting (2.4) into (2.2),

$$\mathbf{n} = {}_q\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times ({}_q\mathbf{I}\boldsymbol{\omega}) + m\mathbf{c} \times (\dot{\boldsymbol{\omega}} \times \mathbf{c}) + m\mathbf{c} \times (\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{c})) + m\mathbf{c} \times \ddot{\mathbf{p}} - m\mathbf{c} \times \mathbf{g} \qquad (2.5)$$

Although the terms $\mathbf{c} \times (\dot{\boldsymbol{\omega}} \times \mathbf{c})$ and $\mathbf{c} \times (\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{c}))$ are quadratic in the unknown location of the center of mass $\mathbf{c}$, these quadratic terms are eliminated by expressing the moment of inertia tensor about the force sensor coordinate origin $({}_p\mathbf{I})$ instead of about the center of mass $({}_q\mathbf{I})$. Rewriting (2.5) as:

$$\mathbf{n} = {}_q\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times ({}_q\mathbf{I}\boldsymbol{\omega}) + m[(\mathbf{c}^T\mathbf{c})\mathbf{1} - (\mathbf{c}\mathbf{c}^T)]\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (m[(\mathbf{c}^T\mathbf{c})\mathbf{1} - (\mathbf{c}\mathbf{c}^T)]\boldsymbol{\omega}) + m\mathbf{c} \times \ddot{\mathbf{p}} - m\mathbf{c} \times \mathbf{g} \qquad (2.6)$$

and using the three dimensional version of the parallel axis theorem

$${}_p\mathbf{I} = {}_q\mathbf{I} + m[(\mathbf{c}^T\mathbf{c})\mathbf{1} - (\mathbf{c}\mathbf{c}^T)] \qquad (2.7\ [10.147])$$

to simplify it results in:

$$\mathbf{n} = {}_p\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times ({}_p\mathbf{I}\boldsymbol{\omega}) + m\mathbf{c} \times \ddot{\mathbf{p}} - m\mathbf{c} \times \mathbf{g} \qquad (2.8)$$

($\mathbf{1}$ is the 3 dimensional identity matrix). All the vectors are expressed in the wrist sensor coordinate system $\mathbf{P}$, so that the quantities $\mathbf{c}$ and ${}_p\mathbf{I}$ are constant. Moreover, the wrist force/torque sensor measures forces and torques directly in the $\mathbf{P}$ coordinate frame.

[1] Equation numbers in brackets refer to equations in Symon, 1971.

In order to formulate the above equations as a system of linear equations, the following equalities and notations are used:

$$\boldsymbol{\omega} \times \mathbf{c} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \triangleq [\boldsymbol{\omega}\times]\, \mathbf{c} \tag{2.9}$$

$$\mathbf{I}\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix} \begin{bmatrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{22} \\ I_{23} \\ I_{33} \end{bmatrix} \triangleq [\bullet\boldsymbol{\omega}] \begin{bmatrix} I_{11} \\ I_{12} \\ I_{13} \\ I_{22} \\ I_{23} \\ I_{33} \end{bmatrix} \tag{2.10}$$

where

$$\mathbf{I} = \mathbf{I}^T = \begin{bmatrix} I_{11} & I_{12} & I_{13} \\ I_{12} & I_{22} & I_{23} \\ I_{13} & I_{23} & I_{33} \end{bmatrix} \tag{2.11}$$

Using these expressions, Eqs. (2.4) and (2.8) can be written as a single matrix equation in the wrist sensor coordinate frame:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{p}} - \mathbf{g} & [\dot{\boldsymbol{\omega}}\times] + [\boldsymbol{\omega}\times][\boldsymbol{\omega}\times] & \mathbf{0} \\ \mathbf{0} & [(\mathbf{g} - \ddot{\mathbf{p}})\times] & [\bullet\dot{\boldsymbol{\omega}}] + [\boldsymbol{\omega}\times][\bullet\boldsymbol{\omega}] \end{bmatrix} \begin{bmatrix} m \\ mc_x \\ mc_y \\ mc_z \\ I_{11} \\ I_{12} \\ I_{13} \\ I_{22} \\ I_{23} \\ I_{33} \end{bmatrix} \tag{2.12}$$

or more compactly,

$$\boldsymbol{w} = A\boldsymbol{\phi} \tag{2.13}$$

where $\boldsymbol{w}$ is a 6 element wrench vector combining both the force and torque vectors, $\boldsymbol{A}$ is a 6×10 matrix and $\boldsymbol{\phi}$ is the vector of the 10 unknown inertial parameters. Note that the center of mass cannot be determined directly, but only as the mass

moment $m\mathbf{c}$. But since the mass m is separately determined, its contribution can be factored from the mass moment later.

2.2.2 Estimating The Parameters

The quantities inside the $\boldsymbol{A}$ matrix are computed by direct kinematics computation (Luh, Walker, and Paul, 1980a) from the measured joint angles. The elements of the $\boldsymbol{w}$ vector are measured directly by the wrist force sensor. Since (2.13) represents 6 equations and 10 unknowns, at least two data points are necessary to solve for the $\boldsymbol{\phi}$ vector, i.e. the force and the position data sampled at two different configurations of the manipulator. For robust estimates in the presence of noise, a larger number of data points must be used. Each data point adds 6 more equations, whereas the number of unknowns, the elements of $\boldsymbol{\phi}$, remain constant. This can be represented by augmenting $\boldsymbol{w}$ and $\boldsymbol{A}$ as:

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{A}[1] \\ \cdot \\ \cdot \\ \boldsymbol{A}[n] \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} \boldsymbol{w}[1] \\ \cdot \\ \cdot \\ \boldsymbol{w}[n] \end{bmatrix}, \qquad n = \text{number of data points} \tag{2.14}$$

where each $\boldsymbol{A}[i]$ and $\boldsymbol{w}[i]$ are matrix and vector quantities described in (2.12). Formulated this way, any linear estimation algorithm can be used to identify the $\boldsymbol{\phi}$ vector. A simple and popular method is the least squares method. The estimate for $\boldsymbol{\phi}$ is given by:

$$\hat{\boldsymbol{\phi}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{w} \tag{2.15}$$

Equation (2.15) can also be formulated in a recursive form (Ljung and Soderstrom, 1983) for on-line estimation.

2.2.3 Recovering Object And Grip Parameters

The estimated inertial parameters (m, $m\mathbf{c}$, ${}_p\mathbf{I}$) are adequate for control, but for object recognition and verification it is also necessary to obtain the principal moments of inertia I_1, I_2, I_3, the location of the center of mass $\mathbf{c}$, and the orientation ${}_{QP}\mathbf{R}$ of $\mathbf{Q}$ with respect to $\mathbf{P}$.

The parallel axis theorem is used to compute the inertia terms translated to the center of mass of the load.

$$\hat{\mathbf{c}} = \frac{\widehat{m\mathbf{c}}}{\hat{m}} \tag{2.16}$$

$${}_q\hat{\mathbf{I}} = {}_p\hat{\mathbf{I}} - \hat{m}[(\hat{\mathbf{c}}^T\hat{\mathbf{c}})\mathbf{1} - (\hat{\mathbf{c}}\hat{\mathbf{c}}^T)] \tag{2.17}$$

Then, the principal moments are obtained by diagonalizing ${}_q\hat{\mathbf{I}}$.

$${}_q\hat{\mathbf{I}} = {}_{QP}\hat{\mathbf{R}} \begin{bmatrix} \hat{I}_1 & 0 & 0 \\ 0 & \hat{I}_2 & 0 \\ 0 & 0 & \hat{I}_3 \end{bmatrix} {}_{QP}\hat{\mathbf{R}}^T \tag{2.18}$$

This diagonalization can always be achieved since ${}_q\hat{\mathbf{I}}$ is symmetric; but when two or more principal moments are equal, the rotation matrix ${}_{QP}\mathbf{R}$ is no longer unique.

2.3 Experimental Results

2.3.1 Estimation on the PUMA Robot

The inertial parameter estimation algorithm was originally implemented on a PUMA 600 robot equipped with an RTI FS-B wrist force/torque sensor (Figure 2.2), which measures all six forces and torques. The PUMA 600 has encoders at each joint to measure joint angles, but no tachometers. Thus, to obtain the joint velocities and accelerations, the joint angles are differentiated and double-differentiated, respectively, by a digital differentiating filter (Figure 2.3). The cutoff frequency of 33 Hz for the filter was determined empirically to produce the best results. Both the encoder data the wrist sensor data were initially sampled at 1000 Hz. It was later determined that a sampling rate of 200 Hz was sufficient, and the data were resampled at the lower rate to reduce processing time. A least squares identification algorithm was implemented as an off-line computation, but an on-line implementation would have been straightforward.

Static Estimation Using The PUMA

To test the calibration of the force sensor and the kinematics of the PUMA arm a static identification was performed. The forces and torques are now due only to the

Figure 2.2: Puma with a test load.

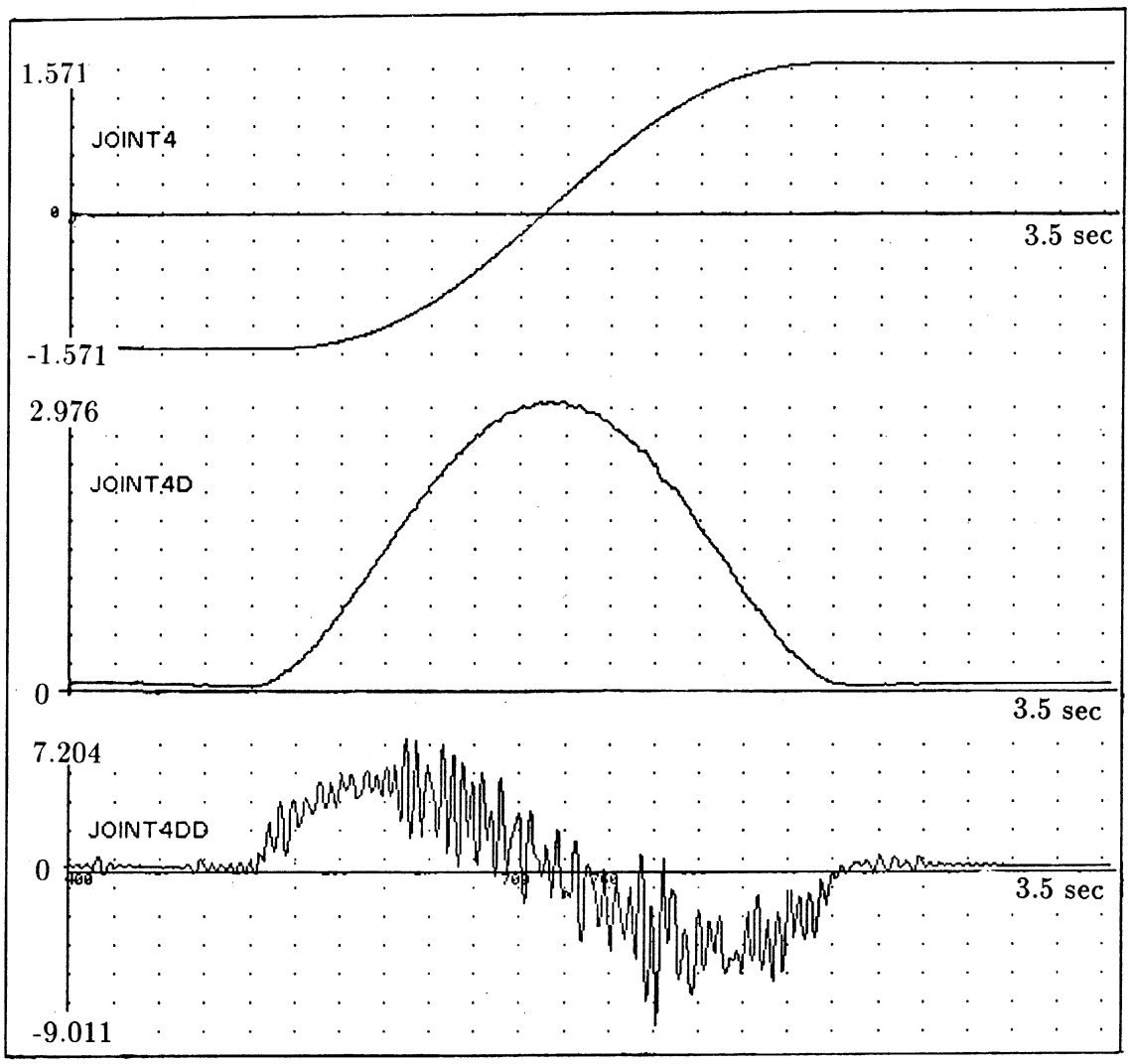


Figure 2.3: Measured angle θ, calculated angular velocity $\dot{\theta}$, and calculated angular acceleration $\ddot{\theta}$ for joint 4.

Parameters	*Actual Values*	*Static Estimates*	*Dynamic Estimates*
Mass (kg)	1.106	1.103	1.067
Change in $c_y(m)$	0.037	0.037	0.039
Mass (kg)	1.106	1.107	1.084
Change in $c_y(m)$	-0.043	-0.043	-0.042
Mass (kg)	1.106	1.100	1.073
Change in $c_y(m)$	-0.021	-0.020	-0.021
Mass (kg)	1.106	1.099	1.074
Change in $c_y(m)$	0.018	0.018	0.020

Table 2.1: Mass and the Center of Mass Estimates

gravity acting on the load, and equations (2.4) and (2.8) simplify to

$$\mathbf{f} = -m\mathbf{g} \tag{2.19}$$

$$\mathbf{n} = -m\mathbf{c} \times \mathbf{g} \tag{2.20}$$

As seen in (2.19) and (2.20), only the mass and the center of mass can be identified while the manipulator is stationary.

To avoid needing to determine the gripper geometric parameters, the center of mass estimates are evaluated by the estimates of the changes in the center of mass as the load is moved along the y-axis from the reference position by known amounts. The results of estimation are shown in the third column of Table 2.1 for an aluminum block $(2 \times 2 \times 6\,in.)$ with a mass of $1.106\,Kg$. Only the changes in c_y are shown in Table 2.1; the estimates of c_x and c_z remained within $1\,mm$ of the reference values ($c_x = 1\,mm$ and $c_z = 47\,mm$). Each set of estimates were computed from 6 sets of data, i.e. data taken at 6 different positions and orientations of the manipulator, where each data point is averaged over 1000 samples to minimize the effects of noise. The results show that in the static case the mass of the load can be estimated to within $10g$ of the actual mass. The center of mass can be estimated to within $1mm$ of the actual values for this load.

Static load estimation only tests the force sensor calibration and the position measurement capabilities of the robot on which the sensor is mounted. In order to

assess the effects of the dynamic capabilities of the robot on load estimation and to be able to estimate the moments of inertia of the load, it is necessary to assess parameter estimation during general movement.

Dynamic Estimation Using The PUMA

In the dynamic case, the joint position encoder and the wrist sensor data are sampled while the manipulator is in motion. A fifth order polynomial trajectory in joint space was used to minimize the mechanical vibrations at the beginning and the end of the movement, and to improve the signal to noise ratio (SNR) in the acceleration data (Figure 2.3). For more popular bang-coast-bang type trajectories, the joint accelerations are zero except at the beginning and the end of the movements, resulting in poor SNR in the acceleration data for most of the movement.

The PUMA robot lacked the acceleration capacity necessary to estimate the moments of inertia of the load. It also lacked true velocity sensors at the joints, which made estimation of the acceleration of the load difficult. The dynamic estimates of mass and center of mass for the previous load are shown in the last column of Table 2.1. The data used in these estimates were sampled while the manipulator was moving from $[0, 0, 0, -90, 0, 0]$ to $[90, -60, 90, 90, 90, 90]$ degrees on a straight line in joint space in 2 seconds. Joint 4 of the PUMA has a higher maximum acceleration than the other joints, and thus, a longer path was given for it. This movement was the fastest the PUMA can execute using the fifth order trajectory without reaching the maximum acceleration for any of its joints. The estimates used all 400 data points sampled during the 2 second movement. The results show slight deterioration in these estimates when compared to the static estimates; but they are still within $40g$ and $2mm$ of the actual mass and displacement, respectively. However, the SNR in the acceleration and the force/torque data were too low for accurate estimates of the moments of inertia for this load ($0.00238Kg \cdot m^2$ in the largest principal moment). In this case, the torque due to gravity is approximately 40 times greater than the torque due to the maximum angular acceleration of the load. Thus, even slight noise in the data would result in poor estimates of **I**.

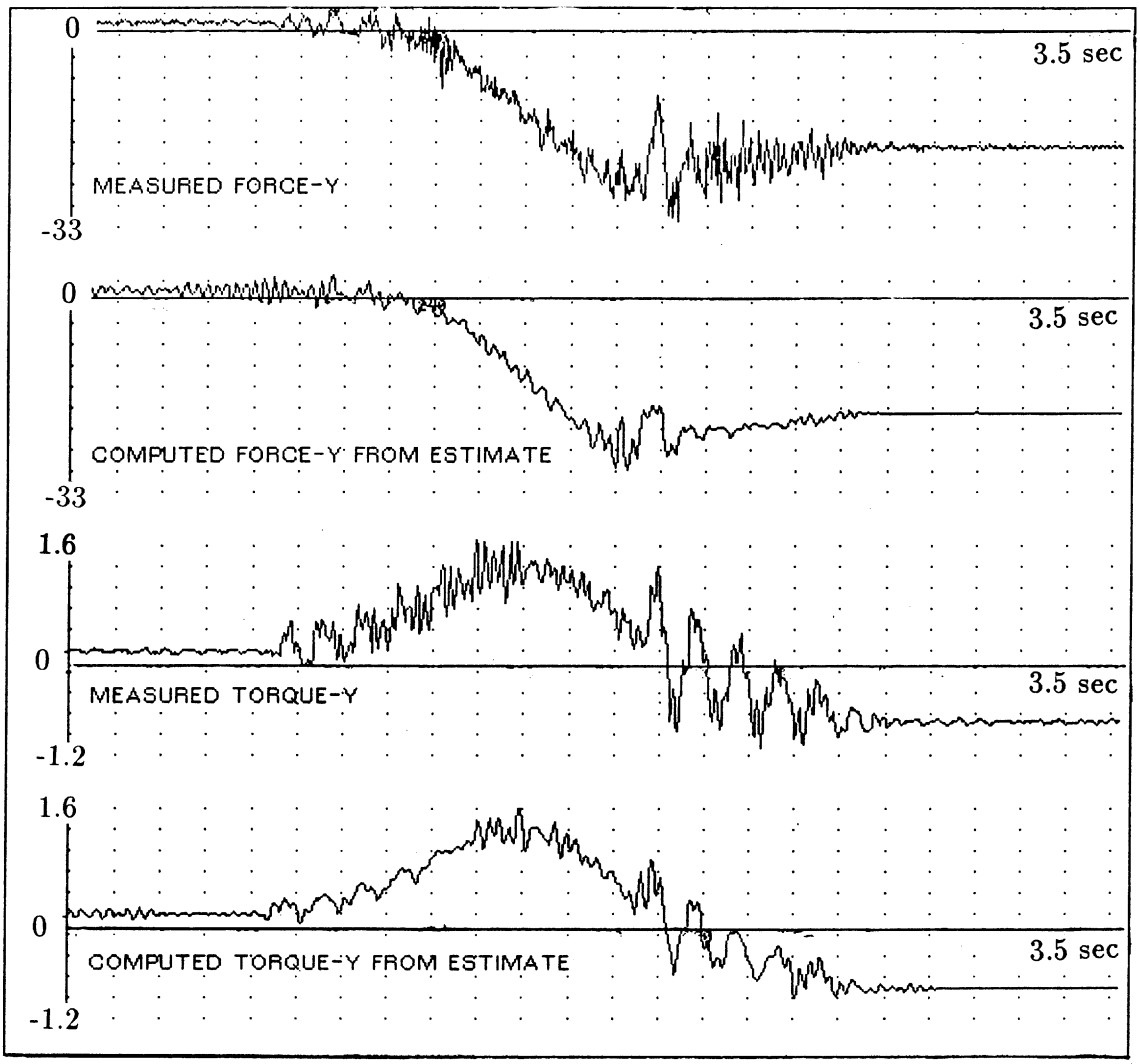


Figure 2.4: Measured force/torque data and computed force/torque data from the estimates using the PUMA.

Parameters $(kg \cdot m^2)$	*Actual Values*	PUMA[1] *Estimates*	PUMA[2] *Estimates*	DDARM[1] *Estimates*
I_{11}	0.0244	0.0192	0.0246	0.0233
I_{12}	0	-0.0048	0.0006	0.0003
I_{13}	0	0.0019	0.0008	0.0007
I_{22}	0.0007	0.0021	0.0036	0.0001
I_{23}	0	-0.0016	-0.0004	-0.0002
I_{33}	0.0242	0.0176	0.0199	0.0236

[1] (all joints moving)

[2] (3 special test movements combined)

Table 2.2: Estimates of the moments of inertia

Special Test Movements Using The PUMA

Therefore, experiments with a different load were performed for the estimates of the moments of inertia. The new experimental load is shown in Figure 2.2. This load has large masses at the two ends of the aluminum bar, resulting in large moments of inertia in two directions ($\sim 0.024kg \cdot m^2$) and a small moment in the other. A typical set of estimates of the moments of inertia at the center of mass frame for the load with the gripper subtracted out are shown in Table 2.3.1 for the above all-joints-moving trajectory. They contain some errors but are fairly close to the actual values.

In order to improve the estimates, the data were sampled while the robot was following three separate 2-second rotational trajectories around the principal axes of the load. Such trajectories used joint 4 and joint 6 only, and resulted in higher acceleration data than the previous trajectory, thus improving the SNR in both the acceleration and the force/torque data. Typical estimates for these special movements show improvements over the estimates with the previous trajectory (Table 2.3.1). Although the estimate of I_{22} is slightly worse than before, all the other terms have improved; the cross terms, especially, are much smaller than before. However, these estimates of **I** are not as accurate as the estimates of the mass and the center of mass shown in Table 2.1. Most of the error is probably due to the large amount of noise present in the acceleration data caused by differentiating the joint angle

data twice. Part of the error may be due to inaccuracies in the current kinematic parameters of the manipulator. Experiments have shown that the actual orientation of the robot can be up to 4° off from the orientation computed from the encoder data.

Figure 2.4 shows the comparison of the measured forces and torques, and the computed forces and torques generated by a simulator from the estimated parameters and the measured joint data. The two sets of figures match very well even in the mechanical vibrations, verifying qualitatively the accuracy of the estimates. This suggests that for control purposes even poor estimation of moment of inertia parameters will allow good estimates of the total force and torque necessary to achieve a trajectory. This makes good sense in that the load forces with the PUMA are dominated by gravitational componenents, and angular accelerations experienced by the load are small relative to those components.

2.3.2 The MIT Serial Link Direct Drive Arm

The effect of the errors causing poor estimates of moment of inertia parameters could be alleviated by increasing the angular acceleration experienced by the load. Since the PUMA robot was already operating at its limits for the experimental results presented above, the algorithm was next implemented on the MIT Serial Link Direct Drive Arm, which has much higher acceleration and velocity capability. The DDARM also has a tachometer on each of its three joints so that numerical differentiation of positions is unnecessary; but the accelerations were obtained by digitally differentiating the velocities using a cutoff frequency of 30Hz. A Barry Wright Company Astek FS6-120A-200 6-axis force/torque sensor was used to measure all three forces and three torques about a point. The positions and velocities of the robot were initially sampled at 1kHz but were later down-sampled to match the sampling frequency of the force/torque sensor of 240 Hz. The identification procedure was again implemented off-line.

Dynamic Estimation Using The Direct Drive Arm

The data used for estimating the inertial parameters of the load were sampled while the manipulator was moving from $(280, 269.1, -30)$ to $(80, 19.1, 220)$ in one second. Again a fifth order polynomial straight line trajectory in joint space was used. The

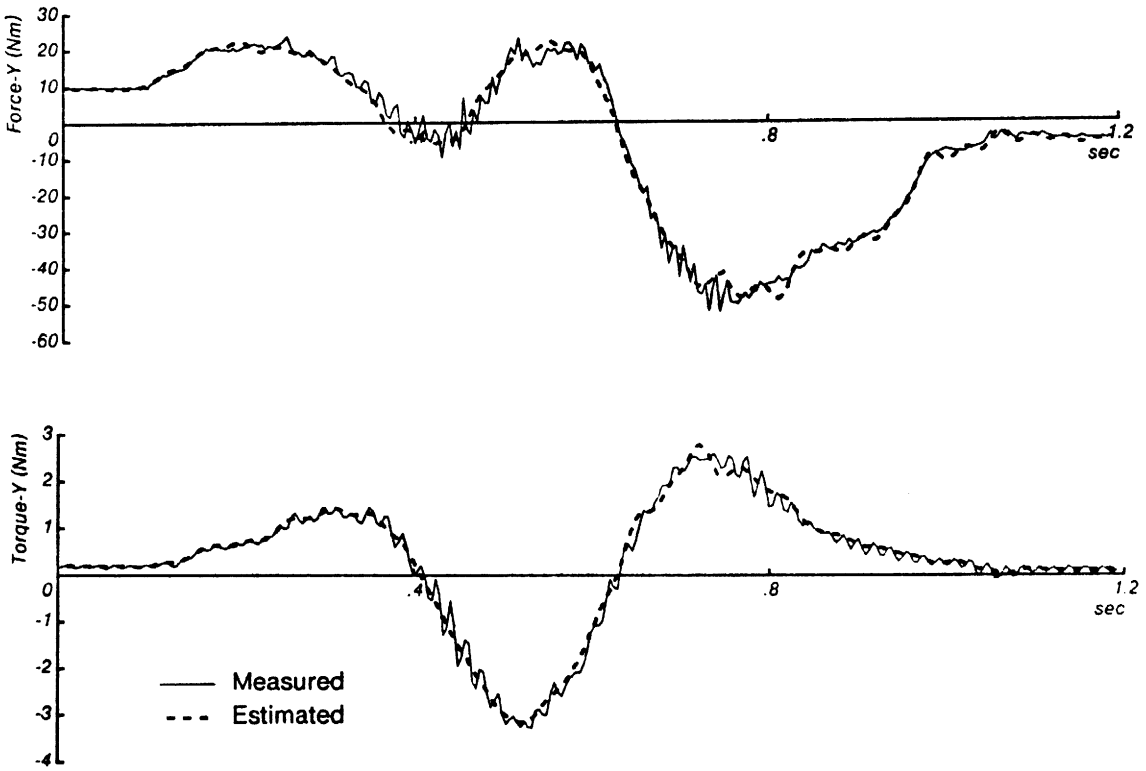


Figure 2.5: Measured force/torque data and computed force/torque data from the estimates using the Direct Drive Arm.

resulting estimates for the moment of inertia parameters are shown in the last column of Table 2.3.1. The estimates for the mass and the location of the center of mass were as good as the PUMA results and are not shown. The estimated moment of inertia parameters are on the whole better than the PUMA results.

Parameters estimated for a set of special test movements using the direct drive arm were not substantially different. The special test movements for the DDARM were not substantially faster than the movement of all joints, and thus probably contained the same amount of information.

Finally, Figure 2.5 shows the comparison of typical measured forces and torques with computed forces and torques generated by a simulator from the estimated parameters and the measured joint data. Once again there is a very good match between the measured and the predicted forces and torques. Thus, as expected, the combination of higher angular accelerations and good velocity sensing results in better parameter estimates,

2.4 Discussion

2.4.1 Usefulness of the Algorithm

It is important to realize that there are two distinct uses of an identified model. For control what matters is matching the input-output behavior of the model (in this case the relationship of load trajectory to load forces and torques) to reality, while for recognition/verification what matters is matching estimated parameters to a set of parameters postulated for reality. Both implementations of load inertial parameter estimation successfully match the input-output behavior of the load (Figures 2.4 and 2.5), However, the limited acceleration capacity of the PUMA robot and its limited sensing result in relatively poor estimates of the moments of inertia of the load without the use of special test motions. In all cases the mass and the location of the center of mass could be accurately estimated from both series of static measurements, and dynamic measurements. Hence, identifying parameters well enough for recognition of the object may require large accelerations or special test movements in order to obtain the moment of inertia parameters accurately.

2.4.2 Sources of Error

This work is preliminary in that an adequate statistical characterization of the errors of the estimated parameters of the predicted forces has not been attempted. Nevertheless, some insights were gained into the sources of such errors.

The ultimate source of error is the random noise inherent in the sensing process itself. The noise levels on the position and velocity sensing are probably negligible, and could be further reduced by appropriate filtering using a model based filter such as the Extended Kalman Filter. The force and torque measurement process involve measuring strain of structure members in the sensor with semiconductor strain gages. The random noise involved in such measurements is also probably negligible.

Bias Errors

However, semiconductor strain gages are notoriously prone to drift. Periodic recalibration of the offsets (very often) and the strain-to-force calibration matrix (often) may be necessary to reduce load parameter estimation errors further. During the experiments presented in this chapter, in order to minimize the bias errors, the data were taken after the force sensors had been left on for a while to warm up and the offsets were recalibrated before each data collection session.

Unmodelled Dynamics

A further source of noise is unmodelled structural dynamics. Neither the robot links nor the load itself are perfectly rigid bodies. A greater source of concern is the compliance of the force sensor itself. In order to generate structural strains large enough to be reliably measured with even semiconductor strain gage technology, a good deal of compliance is introduced into the force sensor. The load rigidly attached to the force sensor becomes a relatively undamped spring mass system. The response of the Astek force sensor to a tap on an attached load is shown in the "undamped impulse response" record of Figure 2.6. The effect of robot movement on this spring mass system is shown in the "undamped movement response" record.

There are several approaches to take to deal with this problem of unmodelled dynamics. One approach is to attempt to identify the additional dynamics. However,

this greatly increases the complexity of the identification process and the amount of data that needs to be collected to get reliable estimates of any parameter.

Another approach is to try to avoid exciting the unmodelled dynamics by choosing robot trajectories that were as smooth as possible. Therefore, the 5th order polynomial trajectories were chosen in the experiments so that the velocities and accelerations are always continuous. Using higher order polynomials would result in even greater smoothness. However, with the PUMA a smooth commanded trajectory did not result in a smooth actual trajectory, because the control methods used and the actual hardware of the robot still introduced substantial vibration. One way to tell if the PUMA is turned on is to touch it and feel if it is vibrating. Vibrations were less of a problem with the direct drive arm, although still present.

The most successful approach is to mechanically damp out the vibrations by introducing some form of energy dissipation into the structure. Hard rubber washers were added between the force sensor and the load. The "damped impulse response" of Figure 2.6 illustrates the response of the force sensor to a tap on the load. The oscillations decay much faster with the added damping. The "damped movement response" indicates that this mechanical damping also greatly reduces the effect of movement on the resonant modes of the force sensor plus load.

A better method would have been to design appropriate damping into the force sensors, just as accelerometers are filled with oil to provide a critically damped response for a specified measurement bandwidth. Failing that, energy dissipation must be introduced either into the structural components of the robot or into the gripper either structurally or as a viscous skin. As will be discussed later, appropriate mechanical damping may also be useful when using a force sensor in closed-loop force control.

Optimal Filtering

Finally, the need to numerically differentiate the velocity to find the acceleration greatly amplifies whatever noise is present. One can avoid the need to explicitly calculate acceleration by integrating equations (2.4) and (2.8). However, since an integrator is an infinite gain filter at the 0 frequency, large errors can result from small low frequency errors such as offsets. The best performance will be achieved from applying some "optimal" filter, whose shape is probably an integrator at high

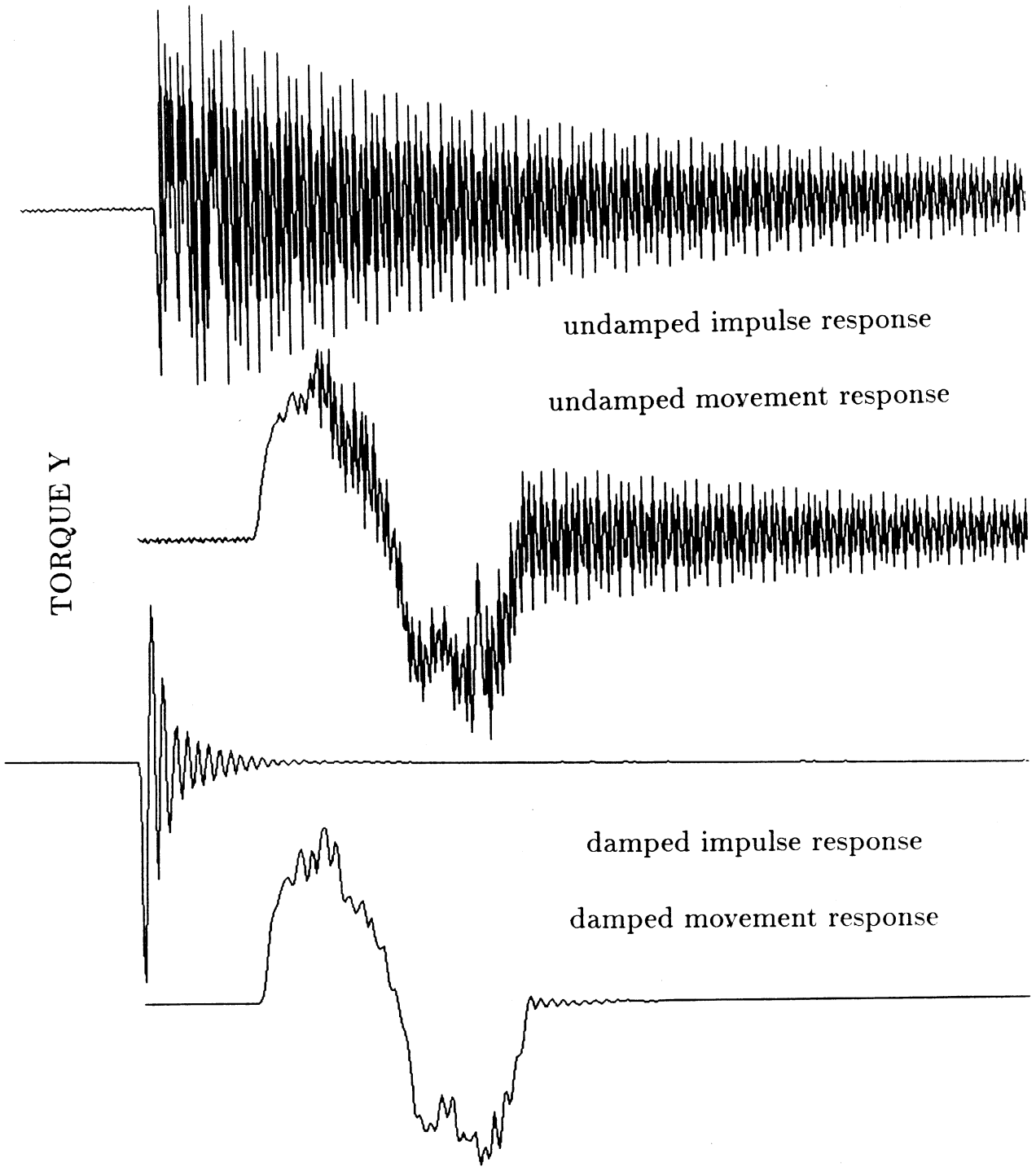


Figure 2.6: Vibration of load on force sensor.

frequencies but a differentiator at lower frequencies (Atkeson, 1986).

2.4.3 Inaccurate Estimates of the Moments of Inertia

One of the factors that make it difficult to identify moments of inertia accurately is the typically large contribution of gravitational torque, which depends only on the mass and the relative location of the center of mass to the force sensing coordinate origin. A point mass rotated at a radius of $5cm$ from a horizontal axis must complete a full 360° rotation in 425 milliseconds for the torque due to angular acceleration to be equal to the gravitational torque, if a 5th order polynomial trajectory is used. A way to avoid gravitational torques is to rotate the load about the vertical axis, or to have the point of force/torque sensing close to the center of mass.

A simple example will illustrate the difficulty of recovering principal moments of inertia, given the moment of inertia tensor about the force sensing origin. The principal moment of inertia of a uniform sphere surface is only 2/7 of the total moment of inertia when it is rotated about an axis tangent to its surface, so that the effects of any errors in estimating the mass, the location of the center of mass, and the grip moments of inertia are amplified when the principal moment of inertia is calculated. This problem can be reduced by having the point of force sensing as close to the center of mass as possible

It still may be difficult to find the orientation of the principal moments of inertia even when the moment of inertia tensor about the center of mass has been estimated fairly accurately. This occurs when two or more principal moments of inertia are approximately equal. Finding the orientation of the principal axis is equivalent to diagonalizing a symmetric matrix, which becomes ill-conditioned when some of the eigenvalues are almost equal. A two dimensional example illustrates the problem. Consider the diagonalized matrix

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{2.21}$$

with eigenvalues $\{\lambda_1, \lambda_2\}$ and whose first principal axis is oriented at an angle θ with respect to the x axis. When the two eigenvalues are almost equal, the terms of the matrix dependent on the angle θ become very small. By substituting $\lambda_1 - \lambda_2 = \epsilon$

into the matrix (2.21),

$$\begin{bmatrix} \lambda_2 + \epsilon \cos^2(\theta) & \epsilon \cos(\theta) \sin(\theta) \\ \epsilon \cos(\theta) \sin(\theta) & \lambda_2 + \epsilon \sin^2(\theta) \end{bmatrix} \tag{2.22}$$

All terms that contain angle information are multiplied by the difference (ϵ) of the principal moments of inertia. With a fixed amount of noise in each of the entries of the identified moment of inertia matrix, the orientation of the principal axis (θ) will become more and more difficult to recover.

2.5 Conclusion

In summary, it was demonstrated that the inertial parameters of a manipulator load can be estimated accurately enough for purposes of control. The estimation algorithm was derived from the reformulation of the Newton-Euler equations for the rigid-body dynamics of a load so that the equations are linear in terms of the unknown inertial parameters. The estimation procedure then involved a simple least squares solution to a set of linear equations. In Chapter 3, this estimation method for a load will be extended to estimate the inertial parameters of all the links of a manipulator.

Chapter 3

Estimation of Link Inertial Parameters

3.1 Introduction

This chapter presents a method of estimating all of the inertial parameters, the mass, the center of mass, and the moments of inertia of each rigid body link of a robot manipulator using joint torque sensing. Determining these parameters from measurements or computer models is generally difficult and involves some approximations to handle the complex shapes of the arm components. Typically, even the manufacturers of manipulators do not know accurate values of these parameters.

The degree of uncertainty in inertial parameters is an important factor in judging the robustness of model-based control strategies. A common objection to the computed torque methods, which involve full dynamics computation (e.g., Luh, Walker, and Paul, 1980b), is their sensitivity to modelling errors, and a variety of alternative robust controllers have been suggested (Samson, 1983; Slotine, 1985; Spong, Thorp, and Kleinwaks; 1984, Gilbert and Ha, 1984). Typically these robust controllers express modelling errors as a differential inertia matrix and coriolis and gravity vectors, but in so doing, no rational basis is provided for the source of errors or the bounds on errors. The error matrices and vectors combine kinematic and dy-

This chapter is a revised version of (An, Atkeson, and Hollerbach, 1985)

namic parameter errors, but kinematic calibration is sufficiently developed so that very little error can be expected in the kinematic parameters (Whitney, Lozinski, and Rourke , 1986).

One aim of this work is to place similar bounds on inertial parameter errors by explicitly identifying the inertial parameters of each link that go into the making of the inertia matrix and coriolis and gravity vectors. The load identification results of Chapter 2 suggests, for example, that mass can be accurately identified to within 1%. Therefore, an assumption of 50% error in link mass in verifying a robust control formulation (Spong, Thorp, and Kleinwaks, 1984) is an unreasonable basis for argument. Slotine (1985) suggests that errors of only a few percent in inertial parameters make his robust controller superior to the computed torque method, but it may well be that these parameters can be identified more accurately than his assumptions.

In this thesis, as an alternative approach, the inertial parameters will be estimated on the basis of direct dynamic measurements. The algorithm of Chapter 2 (Atkeson, An, and Hollerbach, 1985b), used to identify load inertial parameters, can be modified to find link inertial parameters of a robot arm made up of rigid parts. The Newton-Euler dynamic equations are used to express the measured forces and torques at each joint in terms of the product of the measured movements of the rigid body links and the unknown link inertial parameters. These equations are linear in the inertial parameters. However, unlike load estimation, the only sensing is one component of joint torque, inferred from motor current. Coupled with restricted movement near the base, it is, therefore, not possible to find all the inertial parameters of the proximal links. As will be seen, these missing parameters have no effect on the control of the arm.

In this chapter, manipulators with only revolute joints are discussed since handling prismatic joints requires only trivial modifications to the algorithm. The proposed algorithm was verified by implementation on the MIT Serial Link Direct Drive Arm.

3.1.1 Previous Work

Mayeda, Osuka, and Kangawa (1984) required three sets of special test motions to estimate the coefficients of a closed-form Lagrangian dynamics formulation. The 10

inertial parameters of each link are lumped into these numerous coefficients, which are redundant and susceptible to numerical problems in estimation. On the other hand, every coefficient is identifiable since these coefficients represent the actual degrees of freedom of the robot. By sensing torque from only one joint at a time, their algorithm is more susceptible to noise from transmission of dynamic effects of distant links to the proximal measuring joints. For efficient dynamics computation, the recursive dynamics algorithms require the link parameters explicitly. Though recoverable from the Lagrangian coefficients, it is better to estimate the inertial parameters directly. Though this algorithm was implemented on a PUMA robot, it is difficult to interpret the results because of dominance of the dynamics by the rotor inertia and friction.

Mukerjee (1984; Mukerjee and Ballard, 1985) directly applied his load identification method to link identification, by requiring full force-torque sensing at each joint. Instrumenting each robot link with full force-torque sensing seems impractical, and is actually unnecessary given joint torque sensing about the rotation axis. Partially as a result, he does not address the issue of unidentifiability of some inertial parameters. Also, he did not verify his algorithm by simulation or by implementation.

Olsen and Bekey (1985) presented a link identification procedure using joint torque sensing and special test motions with single joints. The unidentifiability of certain inertial parameters was not resolved, and the least squares estimation procedure written as a generalized inverse would fail because of linear dependence of some of the inertial parameters. Again, their procedure was not tested by simulation or by actual implementation on a robot arm.

Neuman and Khosla (1985) developed a hybrid estimation procedure combining a Newton-Euler and a Lagrange-Euler formulation of dynamics. Simulation results for a three degree-of-freedom cylindrical robot were presented, and the unidentifiability of certain inertial components was addressed. For some reason they state link mass must be known for a linear estimation procedure, but such a restriction does not exist with the method of this Chapter. Though planning to work with the CMU DDArm II, they have not yet presented experimental results.

Khalil, Gautier, and Kleinfinger (1986) used a Lagrange formulation in presenting an identification model for link inertial parameters. They addressed the unidentifiability of some parameters, and used it to regroup the dynamic param-

eters and simplify computation. However, they did not try any estimation using their model.

Armstrong, Khatib, and Burdick (1986) measured the inertial properties of a PUMA 560 robot by counter-balancing the disassembled parts. This is an alternative approach to estimation, but is very tedious. Also the cross terms of the inertia matrix cannot be obtained in this way.

3.2 Estimation Procedure

3.2.1 Formulation of Newton-Euler Equations

In Chapter 2, the Newton-Euler equations for a rigid body load were formulated to be linear in the unknown inertial parameters. Then simple linear least squares method was used to estimate those parameters. By treating each link of a manipulator as a load, this formulation can be extended to the link estimation problem. The differences in the equations are that only one component of force or torque is sensed and that the forces and torques from distal links are summed and transmitted to the proximal joints.

Consider a manipulator with n joints (Figure 3.1). Each link i has its own local coordinate system $\mathbf{P}_i$ fixed in the link with its origin at joint i. The joint force and torque due to the movement of its own link can be expressed by simply treating the link as a load and applying the equations from Chapter 2 for load identification:

$$\begin{bmatrix} \mathbf{f}_{ii} \\ \mathbf{n}_{ii} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{p}}_i - \mathbf{g} & [\dot{\omega}\times] + [\omega_i\times][\omega_i\times] & \mathbf{0} \\ \mathbf{0} & [(\mathbf{g} - \ddot{\mathbf{p}}_i)\times] & [\bullet\dot{\omega}] + [\omega_i\times][\bullet\omega_i] \end{bmatrix} \begin{bmatrix} m_i \\ m_i c_{x_i} \\ m_i c_{y_i} \\ m_i c_{z_i} \\ I_{xx_i} \\ I_{xy_i} \\ I_{xz_i} \\ I_{yy_i} \\ I_{yz_i} \\ I_{zz_i} \end{bmatrix}$$

or more compactly,

$$\boldsymbol{w}_{ii} = \boldsymbol{A}_i \boldsymbol{\phi}_i \tag{3.1}$$

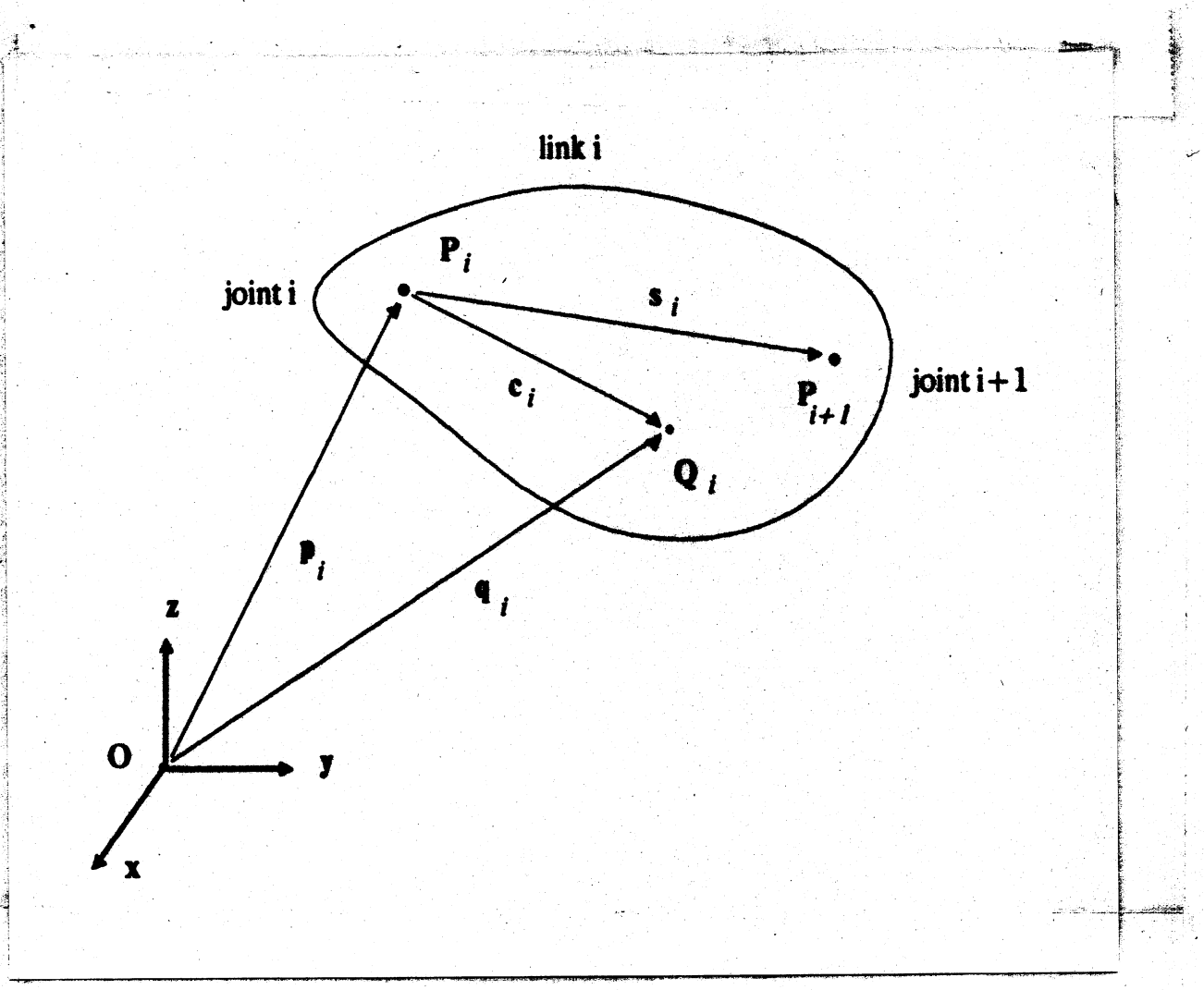


Figure 3.1: Coordinate origins and location vectors for link identification.

where $\boldsymbol{w}_{ij}$ is the wrench (vector of forces and torques) at joint i due to movement of link j alone. $\boldsymbol{A}_i$ is the kinematic matrix that describes the motion of link i and $\boldsymbol{\phi}_i$ is the vector of unknown link inertial parameters. All of the quantities are expressed in the local joint i coordinate system.

The total wrench $\mathbf{w}_i$ at joint i is the sum of the wrenches $\boldsymbol{w}_{ij}$ for all links j distal to joint i:

$$\mathbf{w}_i = \sum_{j=i}^{N} \boldsymbol{w}_{ij} \tag{3.2}$$

Each wrench $\boldsymbol{w}_{ij}$ at joint i is determined by transmitting the distal wrench $\boldsymbol{w}_{jj}$ across intermediate joints. This is a function of the geometry of the linkage only. The forces and torques at neighboring joints are related by

$$\begin{bmatrix} \mathbf{f}_{i,i+1} \\ \mathbf{n}_{i,i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & \mathbf{0} \\ [\mathbf{s}_i\times]\cdot\mathbf{R}_i & \mathbf{R}_i \end{bmatrix} \begin{bmatrix} \mathbf{f}_{i+1,i+1} \\ \mathbf{n}_{i+1,i+1} \end{bmatrix} \tag{3.3}$$

or more compactly

$$\boldsymbol{w}_{i,i+1} = \mathbf{T}_i \, \boldsymbol{w}_{i+1,i+1} \tag{3.4}$$

where

$\mathbf{R}_i$= the rotation matrix rotating the link $i+1$ coordinate system to the link i coordinate system,

$\mathbf{s}_i$= a vector from the origin of the link i coordinate system to the link $i+1$ coordinate system, and

$\mathbf{T}_i$= a wrench transmission matrix.

To obtain the forces and torques at the ith joint due to the movements of the jth link, these matrices can be cascaded:

$$\begin{aligned} \boldsymbol{w}_{ij} &= \mathbf{T}_i\mathbf{T}_{i+1}\cdots\mathbf{T}_j\boldsymbol{w}_{jj} \\ &= \mathbf{U}_{ij}\boldsymbol{\phi}_j \end{aligned} \tag{3.5}$$

where $\mathbf{U}_{ij} = \mathbf{T}_i\mathbf{T}_{i+1}\cdots\mathbf{T}_j\boldsymbol{A}_i$ and $\mathbf{U}_{ii} = \boldsymbol{A}_i$. A simple matrix expression for a serial

kinematic chain (in this case a six joint arm) can be derived from (3.2) and (3.5):

$$\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \mathbf{w}_4 \\ \mathbf{w}_5 \\ \mathbf{w}_6 \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \mathbf{U}_{13} & \mathbf{U}_{14} & \mathbf{U}_{15} & \mathbf{U}_{16} \\ \mathbf{0} & \mathbf{U}_{22} & \mathbf{U}_{23} & \mathbf{U}_{24} & \mathbf{U}_{25} & \mathbf{U}_{26} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_{33} & \mathbf{U}_{34} & \mathbf{U}_{35} & \mathbf{U}_{36} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{44} & \mathbf{U}_{45} & \mathbf{U}_{46} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{55} & \mathbf{U}_{56} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{U}_{66} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \boldsymbol{\phi}_3 \\ \boldsymbol{\phi}_4 \\ \boldsymbol{\phi}_5 \\ \boldsymbol{\phi}_6 \end{bmatrix} \tag{3.6}$$

This equation is linear in the unknown parameters, but the left side is composed of a full force-torque vector at each joint. Since only the torque about the joint axis can usually be measured, each joint wrench must be projected onto the joint rotation axis (typically $[0, 0, 1]$ in internal coordinates), reducing (3.6) to

$$\boldsymbol{\tau} = \mathbf{K}\boldsymbol{\psi} \tag{3.7}$$

where $\tau_i = [0, 0, 0, 0, 0, 1] \cdot \mathbf{w}_i$ is the joint torque of the ith link, $\boldsymbol{\psi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3, \boldsymbol{\phi}_4, \boldsymbol{\phi}_5, \boldsymbol{\phi}_6]^T$, and $\mathbf{K}_{ij} = [0, 0, 0, 0, 0, 1] \cdot \mathbf{U}_{ij}$ when the corresponding entry in (3.6) is nonzero. For an n-link manipulator, $\boldsymbol{\tau}$ is a $n \times 1$ vector, $\boldsymbol{\psi}$ is a $10n \times 1$ vector, and $\mathbf{K}$ is a $n \times 10n$ matrix.

3.2.2 Estimating the Link Parameters

Equation (3.7) represents the dynamics of the manipulator for one sample point. As with load identification, (3.7) is augmented using N data points:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}(1) \\ \cdot \\ \cdot \\ \mathbf{K}(N) \end{bmatrix} \quad \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}(1) \\ \cdot \\ \cdot \\ \boldsymbol{\tau}(N) \end{bmatrix}$$

Unfortunately, one cannot apply simple least squares estimate:

$$\boldsymbol{\psi}_{estimate} = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\boldsymbol{\tau} \tag{3.8}$$

because $\mathbf{K}^T\mathbf{K}$ is not invertible due to loss of rank from restricted degrees of freedom at the proximal links and the lack of full force-torque sensing. Some inertial parameters are completely unidentifiable, while some others can only be identified in linear combinations.

Two different approaches were used to solve the above rank deficient problem. The simplest is ridge regression (Marquardt and Snee, 1975), which makes $\mathbf{K}^T\mathbf{K}$ invertible by adding a small number d to the diagonal elements:

$$\hat{\boldsymbol{\psi}} = (\mathbf{K}^T\mathbf{K} + d\mathbf{I}_{10n})^{-1}\mathbf{K}^T\boldsymbol{\tau} \tag{3.9}$$

The estimates are nearly optimal if $d << \lambda_{min}(\mathbf{K}^T\mathbf{K})$, where λ_{min} is the smallest non-zero eigenvalue of $\mathbf{K}^T\mathbf{K}$.

Another approach expresses the dynamics in terms of a reduced set of inertial parameters that are independently identifiable and that allow the application of a straight least squares estimate. This reduced set can be generated either by examination of the closed form dynamic equations for linear combinations of parameters, or by application of singular value decomposition. Both methods were applied and the results checked against each other. The closed form equations were derived with the aid of MACSYMA (Mathlab Group, 1983) for the MIT Serial Link Direct Drive Arm, since for 3 degrees of freedom the dynamic equations in closed form are already quite complicated. The results are summarized in Appendix 1 in terms of 15 essential variables; made explicit are both the unidentifiable parameters and the parameters identifiable only in linear combinations.

A far less complicated method that can be applied rather automatically to any manipulator kinematic structure is singular value decomposition of $\mathbf{K}$ in (3.8), yielding (Golub and Van Loan, 1983)

$$\mathbf{K} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

where $\boldsymbol{\Sigma} = \text{diag}\{\sigma_i\}$ and $\mathbf{U}$ and $\mathbf{V}^T$ are orthogonal matrices. For each column of $\mathbf{V}$ there corresponds a singular value σ_i which if not zero indicates that the linear combination of parameters, $\mathbf{v}_i^T\boldsymbol{\psi}$, is identifiable. The unidentifiable parameters will have zero singular values associated with them. Since $\mathbf{K}$ is a function only of the geometry of the arm and the commanded movement, it can be generated exactly by simulation rather than by actually moving the real arm and recording data with the concomitant and inevitable noise. For completely unidentifiable parameters, the corresponding columns of $\mathbf{K}$ can be deleted without affecting $\boldsymbol{\tau}$. For parameters identifiable in linear combinations, all columns except one in a linear combination can also be deleted. The resulting smaller $\mathbf{K}^T\mathbf{K}$ matrix will be invertible, and (3.8)

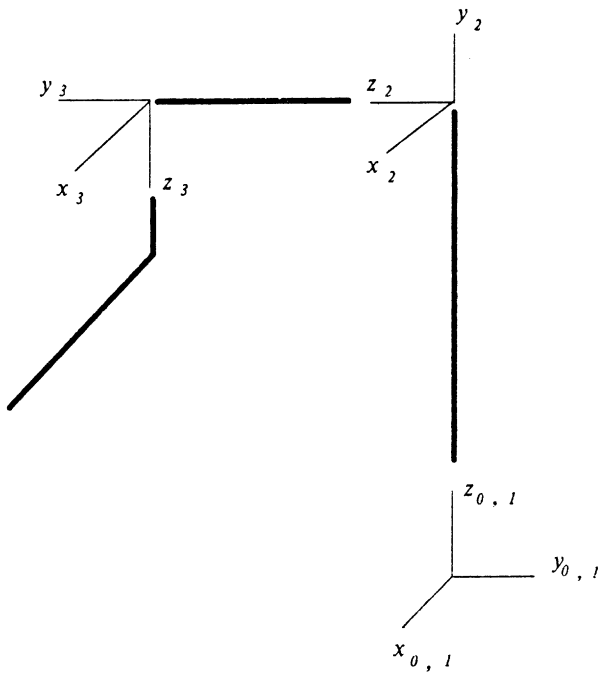


Figure 3.2: The link coordinate system for DDARM

can be used to estimate the reduced set of parameters.

3.3 Experimental Results

Link estimation was implemented on the MIT Serial Link Direct Drive Arm As discussed in Chapter 1, the ideal rigid body dynamics is a good model for this arm, uncomplicated by joint friction or backlash typical of other manipulators. Hence the fidelity of this manipulator's dynamic model suits estimation well. The coordinate system for this arm is defined in Figure 3.2. A set of inertial parameters is available for the arm (Table 3.1)), determined by modeling with a CAD/CAM database (Lee, 1983). These values may not be accurate because of inevitable modeling errors, but they can serve as a point of comparison for the estimation results.

Joint 1 is presently capable of an angular acceleration of 1150 deg/sec^2, joints 2 and 3 in excess of 6000 deg/sec^2. In comparison, joint 1 of the PUMA 600 has a peak acceleration of 130 deg/sec^2 and joint 4 at the wrist 260 deg/sec^2. Joint position is measured by a resolver and joint velocity by a tachometer. The tachometer output is of high quality and leads to good acceleration estimates after differentiation. The accuracy of the acceleration estimates plus high angular accelerations greatly

improves inertia estimation.

The joint torques are computed by measuring the currents of the 3 phase windings of each motor (Asada, Youcef-Toumi, and Lim, 1984). For the three phase brushless permanent magnet motors of the direct drive arm, the output torque is related to the currents in the windings by:

$$\tau = K_T(I_a \sin(n\theta + offset) + I_b \sin(n\theta + 120 + offset) + I_c \sin(n\theta + 240 + offset)) \tag{3.10}$$

where

$$n = \begin{cases} 15 & \text{for Joint 1} \\ 9 & \text{for Joints 2 and 3} \end{cases} \qquad offset^1 = \begin{cases} 60° & \text{for Joint 1} \\ -33.3 & \text{for Joint 2} \\ -108 & \text{for Joint 3} \end{cases}$$

The torque constant K_T for each motor is calibrated statically by measuring the force produced by the motor torque at the end of a known lever arm. As in load identification experiments, the force is measured using a Barry Wright Company Astek FS6-120A-200 6-axis force/torque sensor. Asada, Youcef-Toumi, and Lim (1984) have found that for a motor similar to the motors of the DDARM, the torque versus current relationship was non-linear, especially for small magnitudes of torques, and also varied as a function of the rotor position. However, for the results presented in this paper, the nonlinear effects were ignored since substantial portions of the movements in the experiments required large magnitudes of torques. Since the least squares algorithm minimizes the square of the error, torque errors for torques of small magnitudes do not affect the estimates very much.

For the estimation results presented, 600 data points were sampled while the manipulator was executing 3 sets of fifth order polynomial trajectories in joint space. The specifications of the trajectories were:

1. (330, 289.1, 230) to (80, 39.1, -10) degrees in 1.3s,
2. (330, 269.1, -30) to (80, 19.1, 220) degrees in 1.3s,
3. (80, 269.1, -30) to (330, 19.1, 220) degrees in 1.3s,

Since $\mathbf{K}^T\mathbf{K}$ in (3.9) is singular, estimates for the 30 unknowns are computed by adding a small number ($d = 10.0 << \lambda_{min}(\mathbf{K}^T\mathbf{K}) = 3395.0$) to the diagonal elements of $\mathbf{K}^T\mathbf{K}$.

Parameters	Link 1	Link 2	Link 3
$m(Kg)$	67.13	53.01	19.67
$mc_x(Kg \cdot m)$	0.0	0.0	0.3108
mc_y	2.432	3.4081	0.0
mc_z	35.8257	16.6505	0.3268
$I_{xx}(Kg \cdot m^2)$	23.1568	7.9088	0.1825
I_{xy}	0.0	0.0	0.0
I_{xz}	-0.3145	0.0	-0.0166
I_{yy}	20.4472	7.6766	0.4560
I_{yz}	-1.2948	-1.5036	0.0
I_{zz}	0.7418	0.6807	0.3900

Table 3.1: CAD-modeled inertial parameters.

Parameters	Link 1	Link 2	Link 3
$m(Kg)$	0.0*	0.0*	1.8920†
$mc_x(Kg \cdot m)$	0.0*	-0.1591	0.4676
mc_y	0.0*	0.6776†	0.0315
mc_z	0.0*	0.0*	-1.0087†
$I_{xx}(Kg \cdot m^2)$	0.0*	4.1562†	1.5276†
I_{xy}	0.0*	0.3894	-0.0256
I_{xz}	0.0*	0.0118	0.0143
I_{yy}	0.0*	5.2129†	1.8967†
I_{yz}	0.0*	-0.6050†	-0.0160
I_{zz}	9.33598†	-0.8194†	0.3568

Table 3.2: Estimated inertial parameters.

Linear Combinations	Estimated	CAD-Modeled
$m_3 c_{z_3} l_2 + I_{yz_2}$	-1.0710	-1.3526
$I_{xx_3} - I_{yy_3}$	-0.3691	-0.2702
$I_{zz_2} + I_{xx_3}$	0.7082	0.8632
$I_{zz_1} + I_{xx_2} + I_{xx_3} + m_3 l_2^2$	15.4236	13.0315
$I_{xx_2} + I_{xx_3} - I_{yy_2}$	0.4709	0.4147
$m_3 c_{z_3} - m_2 c_{y_2}$	-1.6863	-3.0814

Table 3.3: Parameters in linear combinations ($l_2 = 0.462\,m$.)

Typical results, obtained using ridge regression method, are shown in Table 3.2. Parameters that cannot be identified because of constrained motion near the base are denoted by 0.0*. The first nine parameters of the first link are not identifiable because this link has only one degree of freedom about its z axis. These nine parameters do not matter at all for the movement of the manipulator and thus can be arbitrarily set to 0.0.

Other parameters marked by (†) can only be identified in linear combinations, indicated explicitly in Table 3.3. The ridge regression automatically resolves the linear combinations in a least squares sense. It can be seen that the estimated sums roughly match the corresponding sums inferred from the CAD-modeled parameters, but the sizeable discrepancy indicates that one parameter set may be more accurate than the other.

To verify the accuracy of the estimated and the modeled parameters, the measured joint torques are compared to the torques computed from the above two sets of parameters using the measured joint kinematic data. As shown in Figure 3.3, the estimated torques match the measured torques very closely. The torques computed from the CAD/CAM modeled parameters do not match the measured torques as closely. This comparison verifies qualitatively that for control purposes the estimated parameters are in fact more accurate than the modeled parameters. However, as in Chapter 2, one cannot make conclusion on the absolute accuracy of the estimates on the basis of the plots in Figure 3.3. The plots only tell us that one can predict the joint torques well.

[1]The offsets are due to abnormalities in the commutation circuitry as well as offsets in resolvers.

Since the purpose of obtaining the estimates was to improve the control performance, the best test for verifying the goodness of the estimates is to use them in robot controllers. In Chapter 4, the estimated inertial parameters will be used in studying the effectiveness of different control algorithms.

3.4 Discussion

Good estimates of the link inertial parameters were obtained as determined from the match of predicted torques to measured torques. The potential advantage of this movement-based estimation procedure for increased accuracy as well as convenience was demonstrated by the less accurately predicted torques based on the CAD-modeled inertial parameters.

The inaccuracy of the CAD-modeled parameters is due to several sources. The links and the motors are complicated and computing the inertial parameters from the schematic drawing of the manipulator is bound to contain modelling errors. For the MIT Serial Link Direct Drive Arm, the masses and moments of inertia are dominated by the large motors at the joints. The modelling of the inertial properties of these motors are difficult since they are made of complicated parts such as the stator windings. Also, the links can be attached to the rotor axes at arbitrary positions by the assembler, introducing uncertainty in the CAD-modeled parameters.

It is possible that the inaccuracy of the CAD-modeled parameters is exaggerated, since the same sensors that were used in the estimation are being used to compare the CAD-modeled parameters to the dynamically estimated parameters. Presumably a systematic error in the sensors, such as a mis-calibration of motor torque constants K_T, would be reflected in the dynamically estimated parameters. This would lead to a judgment of better match with these estimated parameters, even though the CAD-modeled parameters could conceivably be the more accurate. Ideally an independent measuring procedure such as weighing and counterbalancing should be used to resolve this point, but this was not tried.

With regard to errors in the motor torque constant, the motors were calibrated with a commercial force/torque sensor, and it is expected that errors in this calibrating device are very small. Problems of a dead zone near zero torque and torque ripple (Asada, Youcef-Toumi, and Lim, 1984) are not considered to be significant

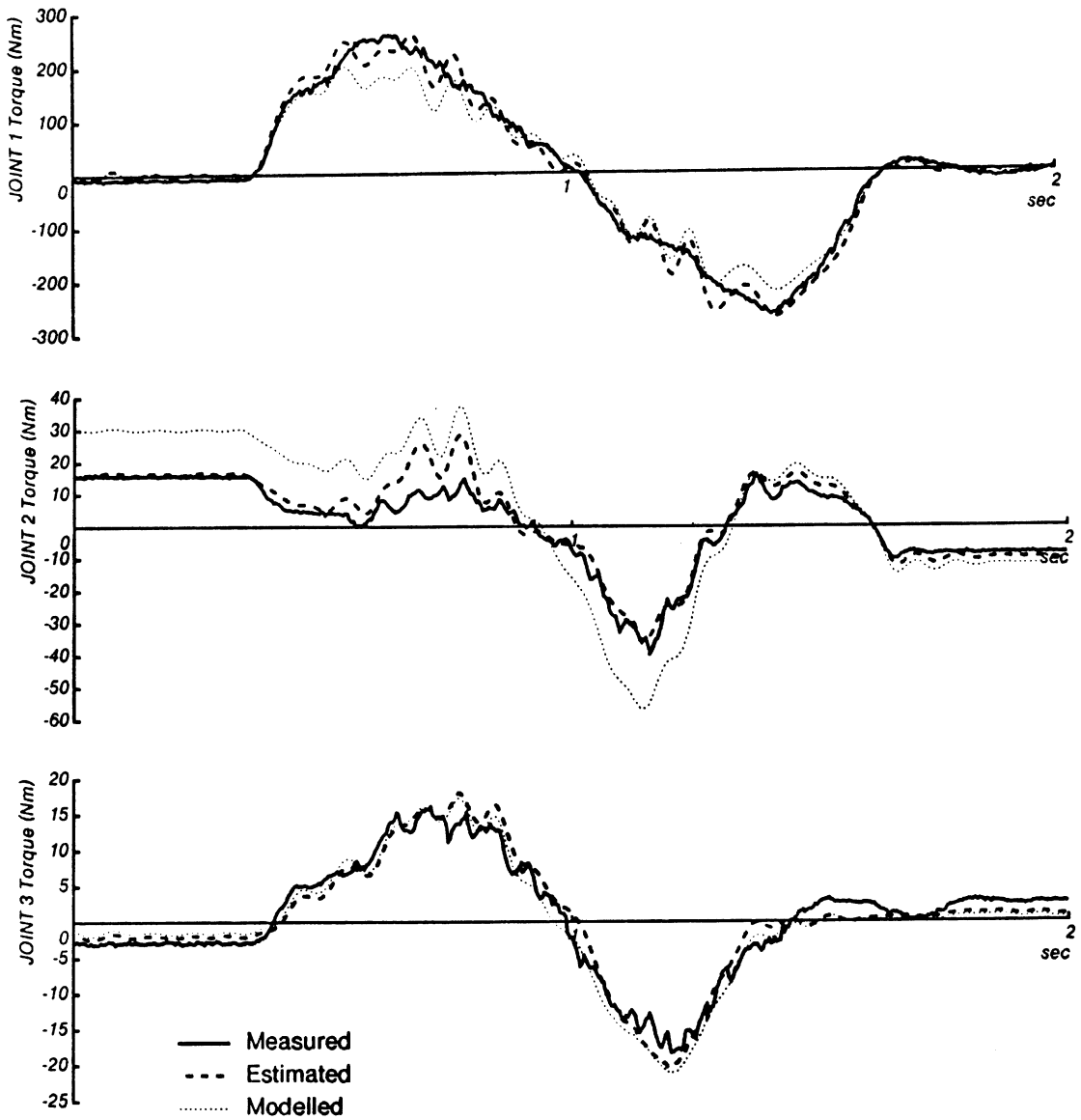


Figure 3.3: The measured, CAD-modelled, and the estimated joint torques

because of the large torques used in this study. Other sources of error are the same as discussed in the previous chapter, and are not repeated here.

Even supposing that there are possible errors in the sensors or kinematic variations due to assembly, the importance of the dynamic estimation of the link inertial parameters is actually emphasized. The controller must deal with the robot kinematics and sensor calibration as they exist, and the estimated model will accommodate kinematic variations and cancel sensor calibration error.

3.4.1 Identifiability of Inertial Parameters

There are three groups of inertial parameters: fully identifiable, identifiable in linear combinations, and completely unidentifiable. Membership of a parameter in a group depends on the manipulator's particular geometry. As shown in Table 3.2 and Appendix 1 for the MIT Serial Link Direct Drive Arm, the 30 inertial parameters are grouped into the following categories:

1. fully identifiable: $m_2c_{x_2}$, I_{xy_2}, I_{xz_2}, $m_3c_{x_3}$, $m_3c_{y_3}$, I_{xy_3}, I_{xz_3}, I_{yz_3}, I_{zz_3}
2. identifiable in linear combinations: I_{zz_1}, $m_2c_{y_2}$, I_{xx_2}, I_{yy_2}, I_{yz_2}, I_{zz_2}, m_3, $m_3c_{z_3}$, I_{xx_3}, I_{yy_3}
3. completely unidentifiable: m_1, $m_1c_{x_1}$, $m_1c_{y_1}$, $m_1c_{z_1}$, I_{xx_1}, I_{xy_1}, I_{xz_1}, I_{yy_1}, I_{yz_1}, m_2, $m_2c_{z_2}$

Some link inertial parameters are unidentifiable because of restricted motion near the base and the lack of full force-torque sensing at each joint. For the first link, rotation is only possible about its z axis. Suppose full force-torque sensing is available at joint 1. It can be seen from (3.1) that I_{xx_1}, I_{xy_1}, and I_{yy_1} are unidentifiable because they have no effect on joint torque. Since the gravity vector is parallel to the z axis, c_{z_1} is also unidentifiable. If it is now supposed that only torque about the z axis can be sensed, then all inertial parameters for link 1 become unidentifiable except I_{zz_1}.

In a multi-link robot a new phenomenon arises. Some parameters can only be identified in linear combinations, because proximal joints must provide the torque sensing to identify fully the parameters of each link. Certain parameters from distal

links are carried down to proximal links until a link appears with a rotation axis oriented appropriately for completing the identification. In between, these parameters appear in linear combinations with other parameters. This partial identifiability and the difficulty of analysis become worse as the number of links are increased.

The ridge regression automatically resolves the linear combinations in a least squares sense, which makes these inertial parameters appear superficially different from those derived by CAD modeling. This is an approximation to computing the pseudo inverse to solve the rank deficient least squares problem.

Although not as simple as ridge regression, singular value decomposition of $\mathbf{K}$ in (3.8) to determine the minimal number of inertial parameters is attractive since it allows reformulating the dynamics with identifiable parameters only. The procedure isolates several sets of parameters whose linear combinations within each set are identifiable. The linear combinations can be reduced by consistently setting certain parameters in these sets to zero, leaving only one non-zero parameter in each set; for example, zeroing m_3, $m_3 c_{z_3}$, I_{xz_3}, and I_{xx_2} leaves I_{yz_2}, I_{yy_3}, I_{zz_2}, I_{zz_1}, I_{yy_2}, and $m_2 c_{y_2}$ as identifiable parameters. The unidentifiable parameters can also be set to zero. Finally, what remains is a reduced full-rank $\mathbf{K}^T\mathbf{K}$ matrix of dimension 15×15.

Chapter 4

Feedforward and Computed Torque Control

4.1 Introduction

The accuracy of the manipulator dynamic model impinges on the performance of a dynamic controller using that model. Since friction is negligible for the direct drive arms, and presuming that one has good control of joint torques, the issue of accuracy reduces to how well the inertial parameters of the rigid links are known. In Chapters 2 and 3, an algorithm was developed for estimating these inertial parameters for any multi-link robot as a result of movement, and the inertial parameters of the MIT Serial Link Direct Drive Arm were estimated. This chapter presents results of utilizing the estimated model to control the robot by both off-line (feedforward) and on-line (computed torque) computation of the joint torques.

Two sets of experiments were performed with the MIT Serial Link Direct Drive Arm (DDARM) involving a subset of proposed control strategies. The first set of experiments is based on a hybrid control system. There is an independent analog servo for each joint with the position and velocity references, and feedforward commands generated by a microprocessor. Since most commercial arms are controlled by simple independent PID controller for each joint, an independent PD controller

This chapter is a revised version of (An, Atkeson, Griffiths, and Hollerbach; 1986)

was tested on this arm to provide a baseline for comparison. The PD controller was augmented by feeding forward first gravity compensation and then the complete rigid body dynamics to ascertain any trajectory following improvements attained by taking the nonlinear dynamics more fully into account. The second set of experiments shows the preliminary results of digital servo implementation, using one Motorola 68000 based microprocessor to control all the joints of the DDARM. The on-line computed torque approach is compared to the PD and to the feedforward approaches using the digital servo.

4.1.1 Control Algorithms

The full dynamics of an n degree-of-freedom manipulator are described by

$$\mathbf{n} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{f}_{fr}(\mathbf{q}, \dot{\mathbf{q}}) \tag{4.1}$$

where $\mathbf{n}$ is the vector of joint torques (for rotational joints), $\mathbf{q}$ is the vector of joint angles, $\mathbf{J}$ is the inertia matrix, $\mathbf{b}$ is the vector of coriolis and centripetal terms, $\mathbf{g}$ is the gravity vector, and $\mathbf{f}_{fr}$ is the vector of friction terms. The simplest and most common form of robot control is independent joint PD control, described by

$$\mathbf{n} = \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) \tag{4.2}$$

where $\dot{\mathbf{q}}_d$ and $\mathbf{q}_d$ are the desired joint velocities and positions, and $\mathbf{K}_p$ and $\mathbf{K}_v$ are n×n diagonal matrices of position and velocity gains.

Feedforward control (Fig. 4.1) augments the basic PD controller by providing a set of nominal torques $\mathbf{n}_{ff}$:

$$\mathbf{n}_{ff}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) = \hat{\mathbf{J}}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \hat{\mathbf{b}}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \hat{\mathbf{g}}(\mathbf{q}_d) + \hat{\mathbf{f}}_{fr}(\mathbf{q}_d, \dot{\mathbf{q}}_d) \tag{4.3}$$

where the hat ($\hat{}$) refers to the modelled values. When this equation is combined with (4.2), the feedforward controller results:

$$\mathbf{n} = \mathbf{n}_{ff}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d) + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) \tag{4.4}$$

The feedforward term $\mathbf{n}_{ff}$ can be thought of as a set of nominal torques which linearize the dynamics (4.1) about the operating points $\mathbf{q}_d, \dot{\mathbf{q}}_d$, and $\ddot{\mathbf{q}}_d$. Therefore, it is reasonable to add the linear feedback terms $\mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$ as the

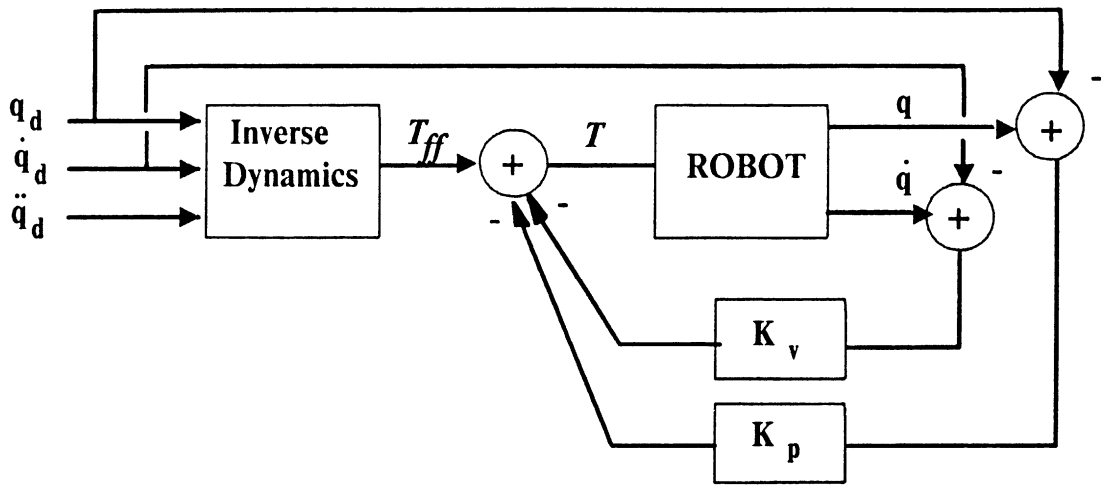


Figure 4.1: Feedforward controller

control for the linearized system. These feedforward terms can be computed off-line, since they are function of the parameters of the desired trajectory only.

On the other hand, the computed torque controller computes the dynamics on-line, using the sampled joint position and velocity data. The control equation is:

$$\mathbf{n}_{ct}(\mathbf{q}_d, \mathbf{q}, \dot{\mathbf{q}}_d, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_d) = \hat{\mathbf{J}}(\mathbf{q})\ddot{\mathbf{q}}^* + \hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q}) + \hat{\mathbf{f}}_{fr}(\mathbf{q}, \dot{\mathbf{q}}) \tag{4.5}$$

where $\ddot{\mathbf{q}}^*$ is given by,

$$\ddot{\mathbf{q}}^* = \ddot{\mathbf{q}}_d + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}). \tag{4.6}$$

If the robot model were exact, then each link of the robot is decoupled, and the trajectory error goes to zero. Gilbert and Ha (1984) have shown that the computed torque control method is robust to small modelling errors.

Previously, Liegeois, Fournier, and Aldon (1980) suggested feedforward control as an alternative to the on-line computation requirements of computed torque control, although they did not present any experimental results. Golla, Garg, and Hughes (1981) discussed different linear state-feedback controller using a linearized model of a manipulator. Asada, Kanade, and Takeyama (1983) presented some results of applying a feedforward control to the early version of a direct drive arm at the Robotics Institute of CMU, though for quite slow movements and for inertial parameters derived by CAD modeling. The computed torque method has been considered by several researchers (Paul, 1972; Markiewicz, 1973; Bejczy, 1974;

Luh, Walker, and Paul, 1980b; Gilbert and Ha, 1984). Although simulation results have been presented, there has been very few published report on the actual implementation of this controller, mainly due to the lack of an appropriate manipulator or on-line computational facility. Working with a direct drive arm, Khosla and Kanade (1986) presented performance comparisons of of the computed torque controller to the PD controller. The conclusion from their experiments is similar to the conclusion of this chapter.

In this paper, the feedforward control is first used as a method of evaluating the accuracy of the estimates of the inertial parameters of the links, and compare performance of the feedforward controller to several other simpler control methods for high speed movements. Then, I present some preliminary results on the implementation of a computed torque controller, again using the estimated inertial parameters of the links.

4.2 Robot Controller Experiment

4.2.1 Feedforward Controller

In this section, performances of several different controllers for full motion of the DDARM is evaluated using a hybrid controller (An, Atkeson, and Hollerbach, 1986). The reference positions and velocities, and the feedforward torques are generated by a microprocessor and input to three independent joint analog servos. Evaluations of the following five control methods are performed with high speed movements of all three joints of the manipulator:

1. PD controller with position reference only:

$$\mathbf{n} = -\mathbf{K}_v\dot{\mathbf{q}} + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

2. PD controller with position reference and feedforward of gravity torques:

$$\mathbf{n} = \hat{\mathbf{g}}(\mathbf{q}_d) - \mathbf{K}_v\dot{\mathbf{q}} + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

3. PD controller with position and velocity references:

$$\mathbf{n} = \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

4. PD controller with position and velocity references plus feedforward of gravity torques:

$$\mathbf{n} = \hat{\mathbf{g}}(\mathbf{q}_d) + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

5. PD controller with position and velocity references plus feedforward of full dynamics:

$$\mathbf{n} = \hat{\mathbf{J}}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \hat{\mathbf{b}}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \hat{\mathbf{g}}(\mathbf{q}_d) + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q})$$

In these experiments, friction was neglected. The nominal position and velocity gains were adjusted experimentally to achieve high stiffness and overdamped charateristics without the feedforward terms.

A fifth order polynomial in joint space was used to generate the reference trajectory. The joints moved from $(80°, 269.1°, -30°)$ to $(330°, 19.1°, 220°)$ in $1.3s$, with peak velocities of 360 deg/sec and the peak accelerations of 854 deg/sec^2 for each joint. The reference trajectory for Joint 1 is shown in Figure 4.2. For control methods (2), (4), and (5), the estimates of the link inertial parameters given in Chapter 3 (An, Atkeson, and Hollerbach; 1985) were used to compute the feedforward torques.

The trajectory errors for the above 5 controllers are shown on Figure 4.3. The errors for the first controller are very large and are out of the graph range. Adding a gravity feedforward term does not help very much, and the trajectory errors for Controller 2 are also very large. This was expected since gravity feedforward is a static correction to Controller 1, and the dynamic effects dominate the response for high speed movements. Modifying the first controller by adding a velocity reference signal improved the response greatly. As with Controller 2, adding a gravity feedforward term did not reduce the trajectory errors very much, and influenced mainly the steady state errors for joints 2 and 3.

The full feedforward controller reduced the trajectory errors significantly for Joints 1 and 2, with peak errors of only 0.33° and 0.64°, respectively. For Joint 3, the feedforward torques did not help because of the light inertia and the dominance

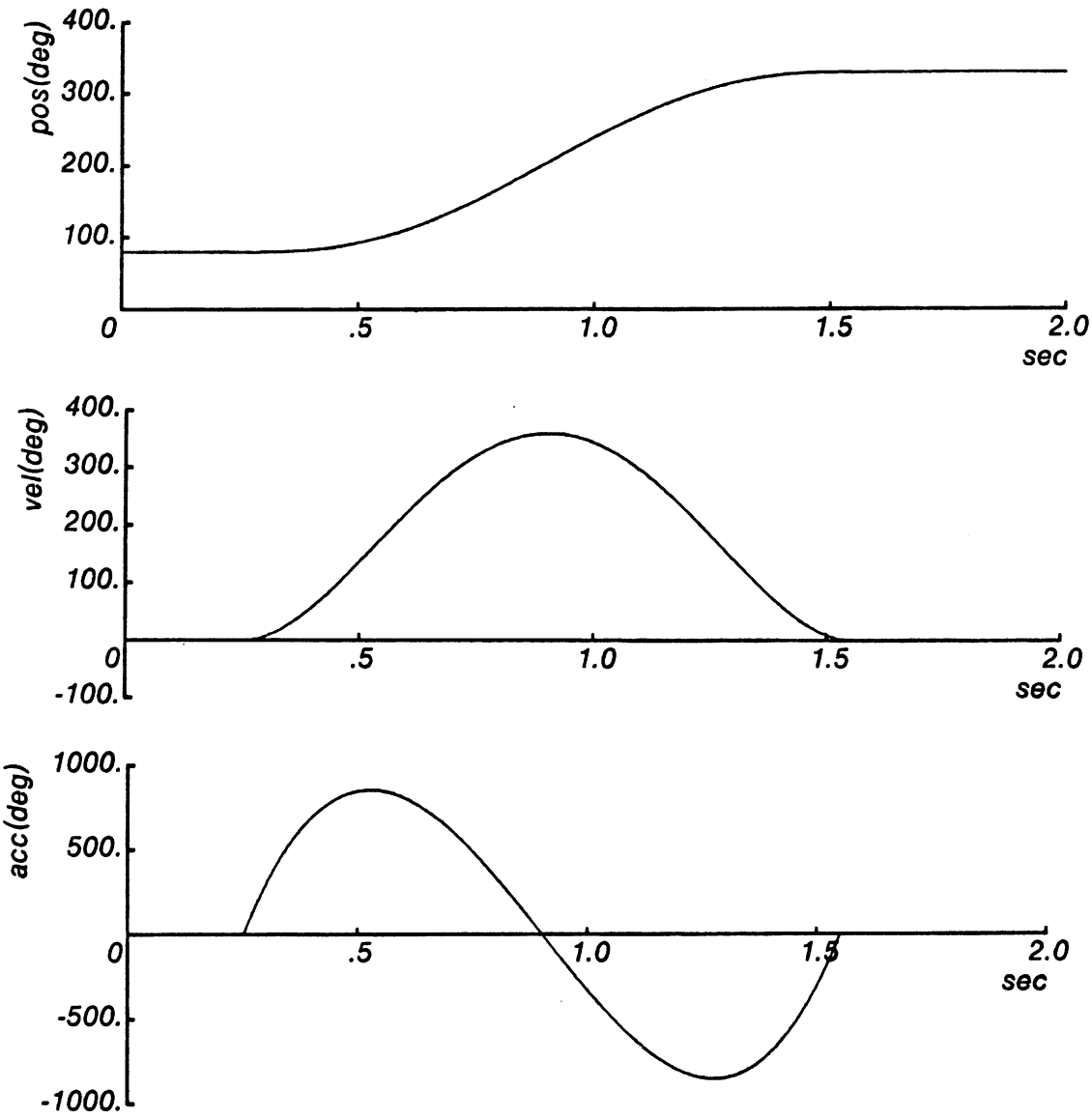


Figure 4.2: Reference trajectory for Joint 1

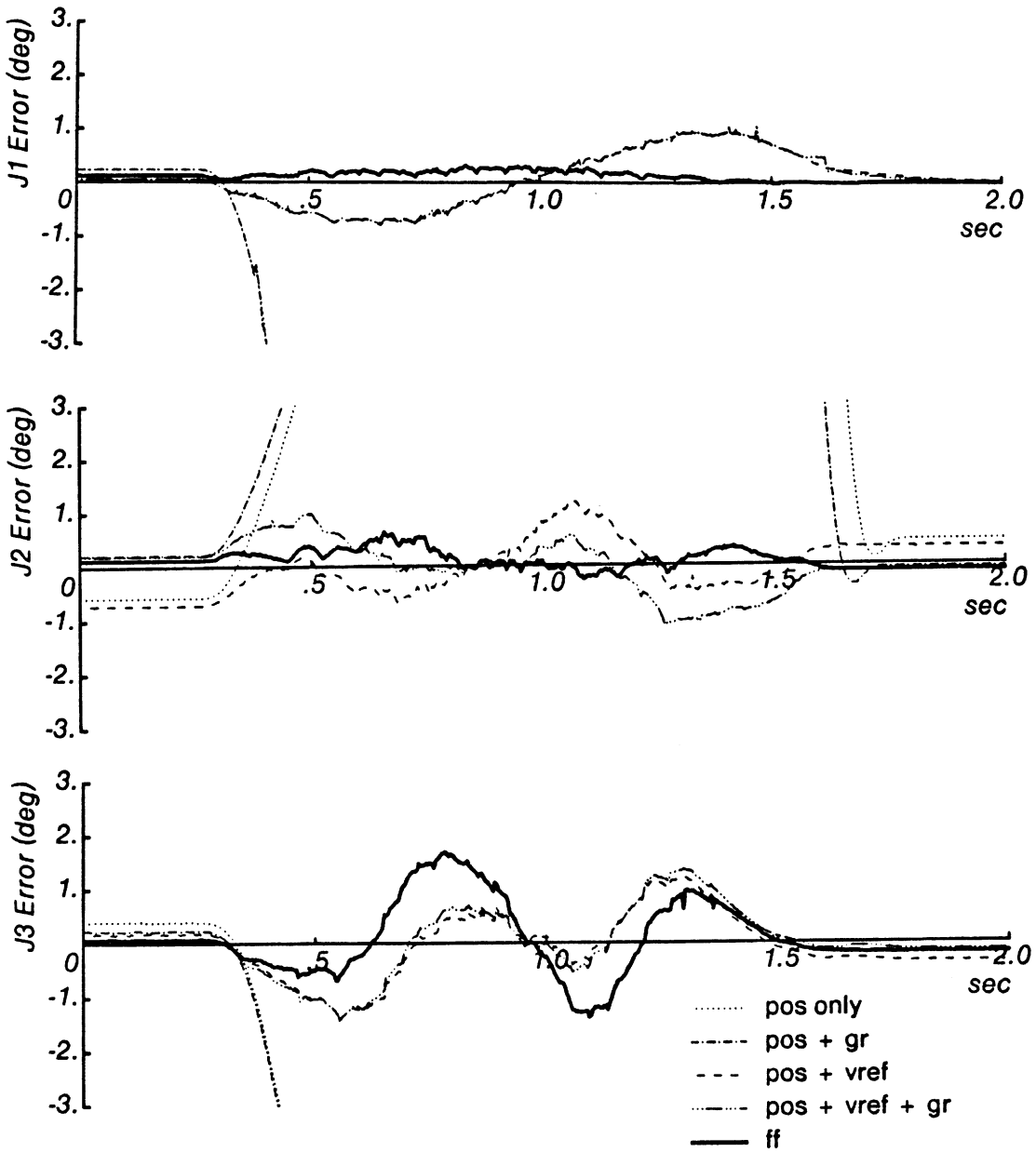


Figure 4.3: Trajectory errors of the 5 controllers for full 1.3s motion.

of unmodelled dynamics in the motor and in bearing friction. The high feedback gains make this joint somewhat robust to these unmodelled dynamics; yet, the trajectory errors could not be reduced below 1.4° with the feedforward torques based on the ideal rigid body model of the link.

4.2.2 Computed Torque Controller

In this section, some preliminary results are presented for the computed torque method implemented on the DDARM. In this implementation, the analog servos are disabled, and the feedback computation is done digitally by one Motorola 68000 based microprocessor using scaled fixed-point arithmetic. Written in the C language, the controller, including the full computation of the robot dynamics, runs at a 133 Hz sampling frequency. Although further improvements in computation time are possible, this speed was adequate in demonstrating the efficacy of dynamic compensation. The details of this implementation are discussed in (Griffiths, 1986).

A similar fifth order polynomial trajectory as in the previous section was used for this experiment. Figure 4.4 shows the trajectory errors for three controllers: the digital PD controller, the feedforward controller using a digital servo, and the computed torque controller. The computed torque and the feedforward controllers both show a significant reduction in tracking errors for Joints 1 and 2 compared with the PD control, with no clear distinction between feedforward and computed torque. With the addition of dynamic component, the peak tracking errors for Joint 1 are reduced from 4.4° to 2.2° and for Joint 2, from 3.5° to 2.0°. As before, the trajectory errors for Joint 3 were not reduced by the computed torque or the feedforward controller. Again, this seems to indicate that the model for the third link may not be very good.

The trajectory tracking performance of the computed torque controller is not as good as that of the analog feedforward controller of the previous section. The main reason for this is the slow sampling frequency (133 Hz) of the digital controller, as compared to the 1 KHz sampling frequency at which the reference inputs were given to the analog servos. Improvements in the computation time should also improve the tracking performance of the computed torque controller.

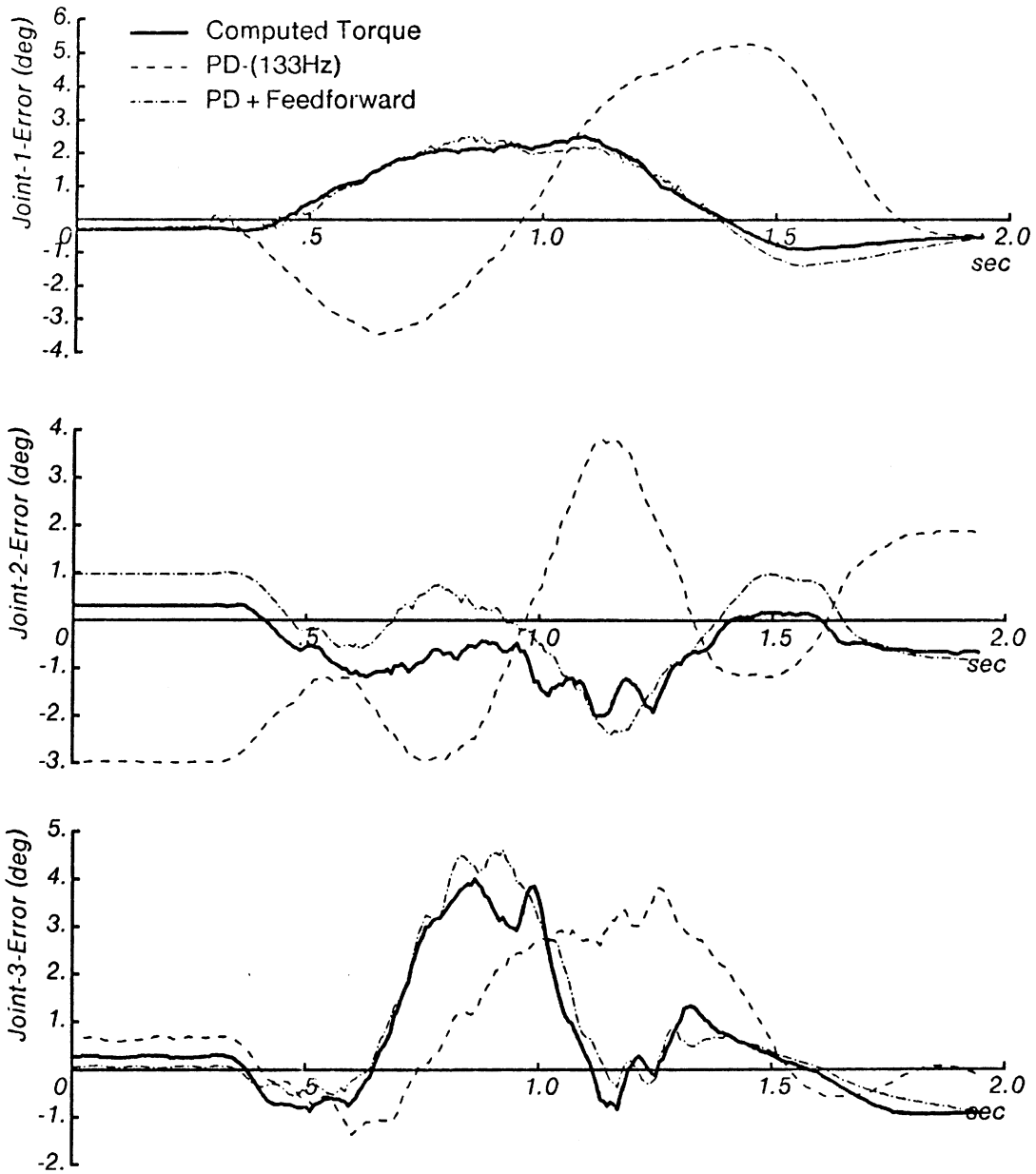


Figure 4.4: Trajectory errors of the three digital controllers for full 1.3s motion.

4.3 Conclusions

In this chapter, I presented some experimental results of using an estimated dynamic model of the manipulator for dynamic compensation via feedforward and computed torque control methods. The results indicate that dynamic compensation can improve trajectory accuracy significantly and that the estimated rigid body model of the manipulator is quite accurate and adequate for control purposes for Joints 1 and 2. The unmodelled dynamics of the light third link, including the motor dynamics and friction, are significant and yield larger trajectory errors than at the other two joints. Therefore, for Joint 3, it may be necessary to use a more complete model to improve trajectory following.

The results of the digital implementation of the feedforward and computed torque controllers were not as good as the hybrid feedforward controller. This indicates that if a robot was being used solely for free space movements without significant variation of its loads, then a hybrid controller using an independent analog servo for each joint may be quite adequate. A hybrid controller, however, is not flexible, and cannot handle varying loads or interactions with the environment.

Chapter 5

Dynamic Stability Issues in Force Control

5.1 Introduction

It is necessary to make robots interact with their environments to make them versatile. This is especially true for assembly applications with tight tolerances since robots may not have the necessary positional accuracy or have some unmodelled misalignment. Vision will be a part of the solution for such problems. But even then, robots need to interact with the environments in a stable way. This is the subject of force or compliance control studies that have been getting much attention recently in the robotics field. However, in all of the active continuous feedback implementations mentioned in Chapter 1, researchers have found that their systems could not deal with stiff environments satisfactorily (Whitney, 1985; Caine, 1985). Previous implementations have been either unstable or very slow. The slow performance is due to high joint damping required to maintain stability during contact between the tip of the robot and its work piece, i.e. during interaction with a stiff environment. Without such high damping, researchers have observed a very unstable behavior in a force-controlled robot.

Stability is a condition that must be satisfied with any control systems. Yet, often in the area of robotics, the stability problems have been neglected and treated only as an afterthought. Only during the last two years have there been papers

discussing the stability problems associated with force control (Whitney, 1985; Roberts, 1984, 1985; Kazerooni, 1985, 1986a, 1986b, 1986c; Eppinger, 1986; Wlasich, 1986). Both Whitney (1985) and Roberts (1984) presented stability analysis with respect to the sensor stiffness and showed that a soft force sensor is necessary in order to achieve a stable behavior with stiff environments. Kazerooni (1985) has presented a stability analysis and design method for impedance controller using eigen structure assignment. But his algorithm is quite complicated, and it is not clear if it can be implemented for a multi degree of freedom manipulator. His implementation results have so far been on a single joint manipulator. Wlassich (1986) implemented an impedance controller on a two-link experimental manipulator. Using the impedance control method, it may be desired to make the manipulator behave as if it were a smaller mass (or inertia) than the actual. However, Wlassich discovered that unless the desired mass was larger than the actual mass, the manipulator became unstable against a stiff environment. Eppinger's analysis (1986) is similar to that contained in Section 5.2, but he does not come to any conclusions for remedying the stability problems.

The goal of this and the next two chapters is to present simple analyses to understand the stability problems associated with force control implementations and then to present some design methods that would remedy those problems. This chapter focuses on the dynamic instabilities that are caused by the interaction of the dynamics of the robot with its environment. Since these problems occur whether we are dealing with a single degree of freedom manipulator or a multi degree of freedom manipulator, a simple one link manipulator is used in the analyses. The analyses are then verified by single link experiments on the third link of the direct drive arm. Multi-link cases are discussed in Chapters 6 and 7.

After presenting the dynamic stability problems in Section 5.2, three different solutions are suggested to improve the dynamic stability of a force-controlled manipulator. The first method, use of compliant covering, is similar to the results and suggestions of Whitney (1985) and Roberts (1984). The second method, adaptation to the environment stiffness, is new and has not been reported before. The third method, use of joint torque control, has been mentioned by Wu and Paul (1980) and by Luh, Fisher, and Paul (1983); but their results were used mainly to reduce joint friction in a position controlled system, and not in active force control.

As mentioned in Chapter 1, the direct drive arm is an ideal device for testing

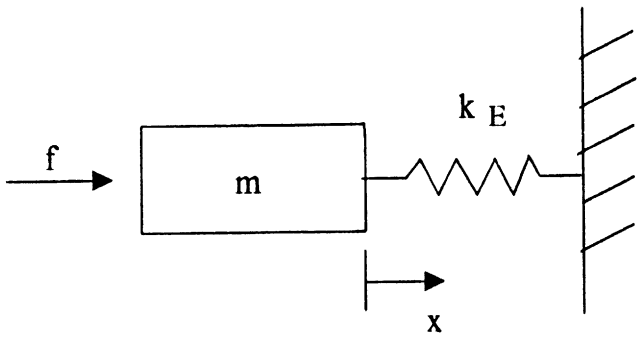


Figure 5.1: Model of the robot and the environment

these ideas for the following reasons:

- the dynamics are ideal since there is little friction and no backlash,
- torques to the motors can be measured and controlled accurately.

5.2 Stability Problems

In this section, I will discuss the dynamic stability problems of a force-controlled robot. Two types of force control methods are used as examples, although the problems are general and not limited to these two. At first, the stability problems will be analyzed using a simple model of the robot and then demonstrated by experiments on the direct drive arm.

5.2.1 General Stability Analysis

To simplify the analysis, I will consider a very simple model of the robot and the environment (Fig. 5.1). The robot, in contact with its environment, is modelled as a mass m, and the environment plus the wrist force sensor is modeled as a simple spring of stiffness k_E. Then the dynamic equation of the above simplified system is,

$$f - k_E x = m\ddot{x} \tag{5.1}$$

Stiffness Control For a stiffness controller (Salisbury, 1980) of the above model, the control input is given by,

$$\begin{aligned} f&=k_p(x_d-x)+k_v(\dot{x}_d-\dot{x})+k_f(k_p(x_d-x)-f_{ext}) \\ &=k_p(x_d-x)+k_v(\dot{x}_d-\dot{x})+k_f(k_p(x_d-x)-k_Ex) \end{aligned} \tag{5.2}$$

where k_p is the desired stiffness, k_v is the velocity gain, and k_f is the force feedback gain. Substituting (5.2) into (5.1), the dynamics of the total controlled system is described by,

$$m\ddot{x}+k_v\dot{x}+(1+k_f)(k_p+k_E)x=input\ terms \tag{5.3}$$

Impedance Control An impedance controller (Hogan, 1985a, 1985b, 1985c) results in a similar form as (5.2), but with k_f depending on the desired apparent mass. The control equation of an impedance controller, discussed in (Hogan, 1985b) for the simple one degree of freedom robot, is as follows:

- Desired impedance:

$$m_d\ddot{x}+k_v(\dot{x}-\dot{x}_d)+k_p(x-x_d)=f_{ext} \tag{5.4}$$

- Control Input:

$$\begin{aligned} f&=\tfrac{m}{m_d}(k_p(x_d-x)+k_v(\dot{x}_d-\dot{x}))+(1-\tfrac{m}{m_d})f_{ext} \\ &=\tfrac{m}{m_d}(k_p(x_d-x)+k_v(\dot{x}_d-\dot{x}))+(1-\tfrac{m}{m_d})k_Ex. \end{aligned} \tag{5.5}$$

m_d is the desired mass, k_v is the desired damping, and k_p is the desired stiffness. (5.5) shows that the force gain k_f is directly related to the desired mass m_d.

$$k_f=(1-\frac{m}{m_d}) \tag{5.6}$$

Therefore, if $m_d < m$, then k_f is negative and large. If $m_d = m$, then the force sensor feedback is disabled. If $m_d > m$, then k_f is small but positive. This positive force feedback, however, does not cause instability because of the $-k_Ex$ term already present in the open loop system (5.1). The total system with the above impedance control is described by

$$m\ddot{x}+\frac{m}{m_d}k_v\dot{x}+\frac{m}{m_d}(k_p+k_E)x=input\ terms \tag{5.7}$$

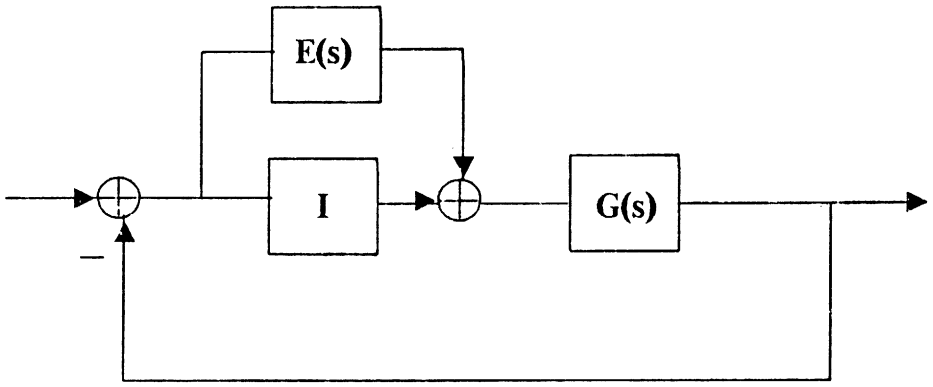


Figure 5.2: Error model for unmodelled dynamics

This equation is almost identical to (5.3) for the stiffness controlled system. In both cases, the closed loop equation includes a position term x which is multiplied by the effective stiffness of the environment k_E and the force feedback gain k_f. In fact, this form of equation will result for any type of force control methods.

The system described by (5.3) or (5.7) is a stable system since all the poles have negative real parts given positive gains, k_v and k_p. In free space or in contact with a soft environment, k_E would be small, and the manipulator will behave satisfactorily. But if k_E is large, i.e. $k_E >> k_p$, the system will be undesirably underdamped if k_v was computed with only k_p in mind, neglecting the large k_E term. The combination of the force sensor and the environment is approximately a very stiff spring, and from the stability point of view, force feedback is a *very high gain* position feedback.

Yet, given the above completely linear and perfectly modelled system, this is still a stable system since the poles are still on the left hand plane. However, real systems are non-linear, and they are never perfectly modelled. In such real situations, the system may actually be unstable in presence of unmodelled high frequency dynamics. Figure 5.2 shows the nominal closed-loop model of a force-controlled system and the use of multiplicative error model to represent the unmodelled dynamics. $E(s)$ is not completely known and is only an approximation to the unmodelled dynamics. In a single input single output system, for stability robustness with the above error model (Lehtomaki, 1982) shown in Figure 5.2, we need:

$$|E(s)| < |1 + G^{-1}(s)| \tag{5.8}$$

This relation is derived in Appendix 2 from the Nyquist criterion for stability. For my simple model of combined manipulator-environment system, the loop transfer

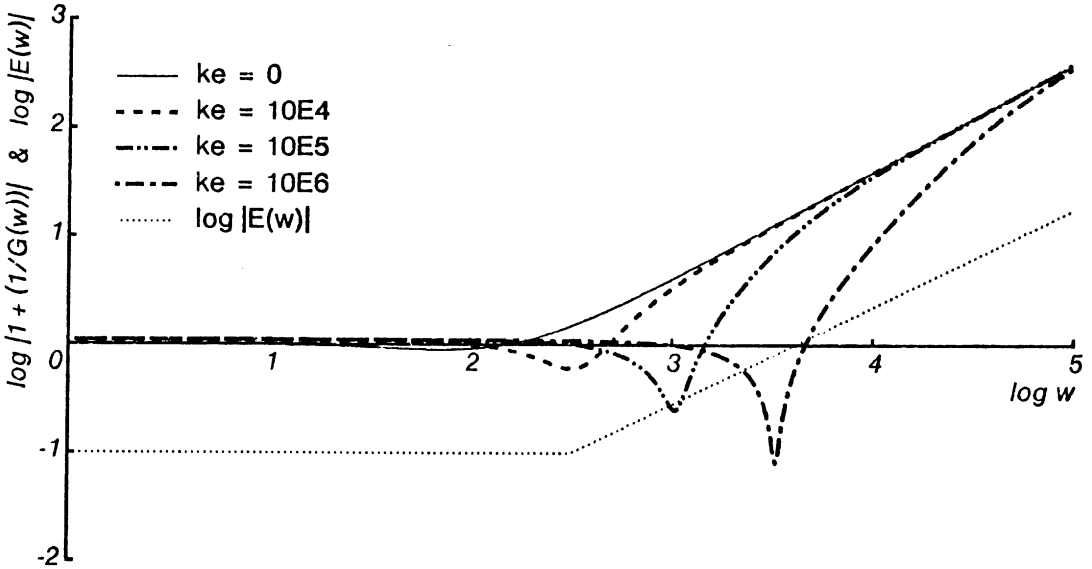


Figure 5.3: $|I + G^{-1}(s)|$ for $k_v = 250$, $m = 1$, $k_f = 10$, and $k_p = 1000$; Typical $|E(s)|$

function is:

$$G(s) = \frac{sk_v + k_p(1 + k_f) + k_E k_f}{s^2 + k_E} \tag{5.9}$$

The plots of $|E(s)|$ and $|1 + G^{-1}(s)|$ for different values of k_E are shown on Figure 5.3. The plot of $|E(s)|$ is a hypothetical one, but its shape is typical of the high frequency unmodelled dynamics (Kazerooni, 1985). The figure shows that for a small k_v, which still gives overdamped response in free space, $|I + G^{-1}(s)|$ has a large dip in the high frequency region for large values of k_E. Since the unmodelled dynamics are greater for higher frequencies, this results in a non-robust system. In fact, with $E(s)$ as shown on the figure, $|E(s)| > |I+G^{-1}(s)|$ for $k_E > 10^5\ Nt/m$, and this system then may be unstable for contact with such stiff environments. But for a soft environment, the diagram shows that the system remains stable in presence of unmodelled dynamics. Such behaviors have been noticed by the researchers who have tried implementing force control for a manipulator.

In summary, there are three problems affecting stability for a robot interacting with its environments. They are:

1. force sensor feedback is essentially high gain position feedback,

2. there are always unmodelled high frequency dynamics in the robot system, and

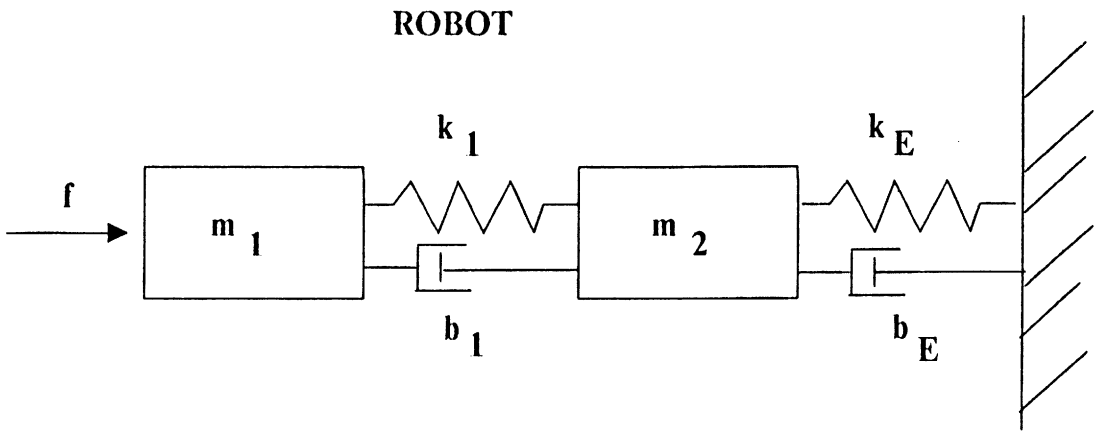


Figure 5.4: 2 mass model (flexibility)

3. the robot must deal with stiff environments.

In the next two sections, I will verify this general simple analysis with an example and also by implementation on the MIT Serial Link Direct Drive Arm.

5.2.2 Example of Unmodelled dynamics

The general analysis of the previous subsection is verified in this section with a slightly more complex model of a robot. Even single degree of freedom robots are not completely rigid and there are always flexible modes in the system. Adding a flexible mode to the simple system (Fig. 5.1), a fourth order model of the robot in contact with the environment is shown in Figure 5.4.

With this system, if a stiffness controller or any other force controller was designed straightforwardly neglecting the flexibility, thus treating it as a simple one mass system, the root locus of the closed-loop system with stiff environment ($k_E = 8 \times 10^5$ N/m) would behave as shown in Figure 5.5. As the force gain k_f is increased, the poles move to the right half plane and the system becomes unstable. When the robot is in contact with a soft environment ($k_E = 10^3 N/m$), the closed-loop poles do not move to the right half plane for the same force gains as shown in Figure 5.6. These two cases agree with the behavior predicted by the general analysis of the previous section.

5.2.3 Experimental Verification of Instability

In this section, experimental results on the third link of the direct drive arm are presented to verify the analyses of previous sections. The third link is being controlled in a pure force control mode as shown on the block diagram of Figure 5.7.

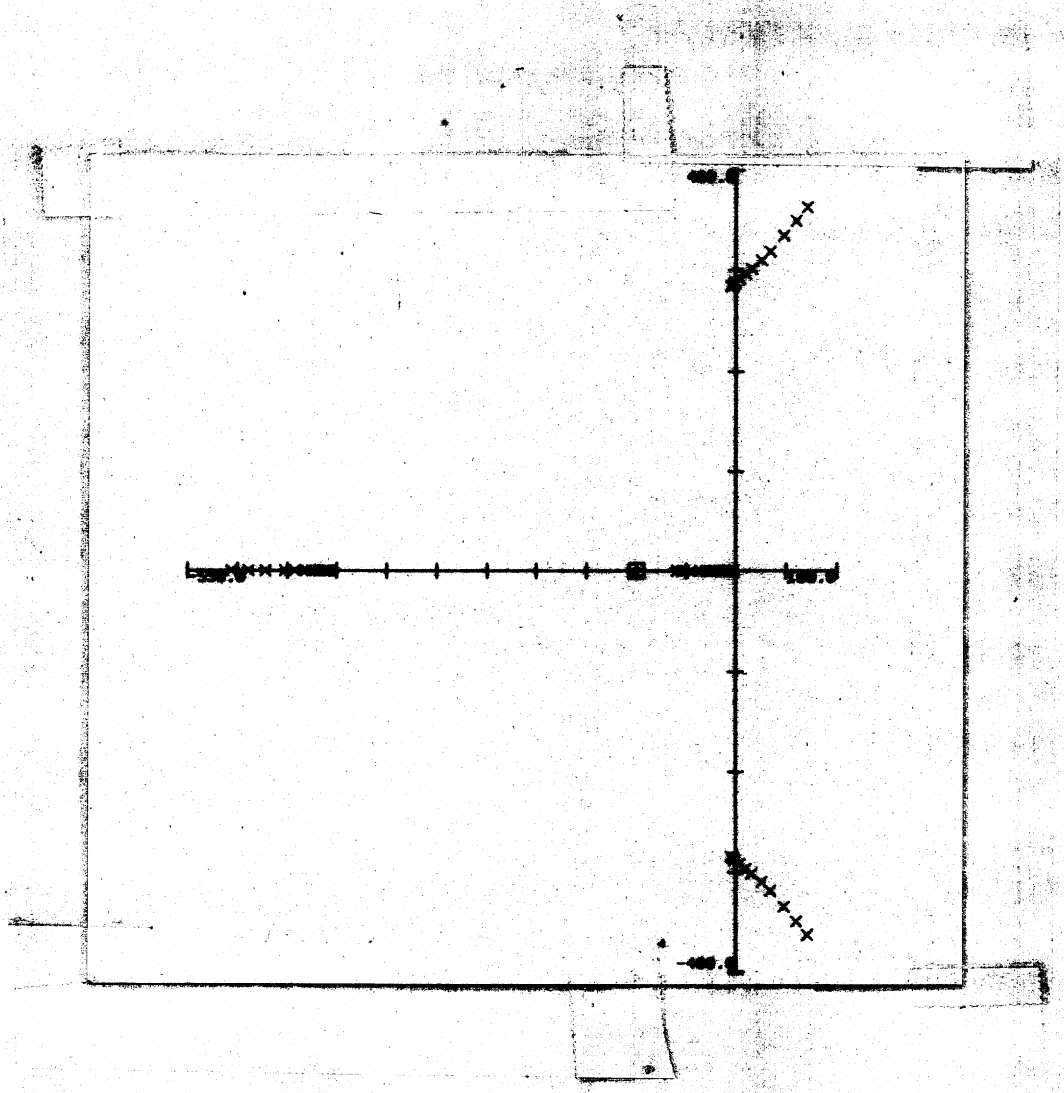


Figure 5.5: Root locus for 2 mass model as k_y varies

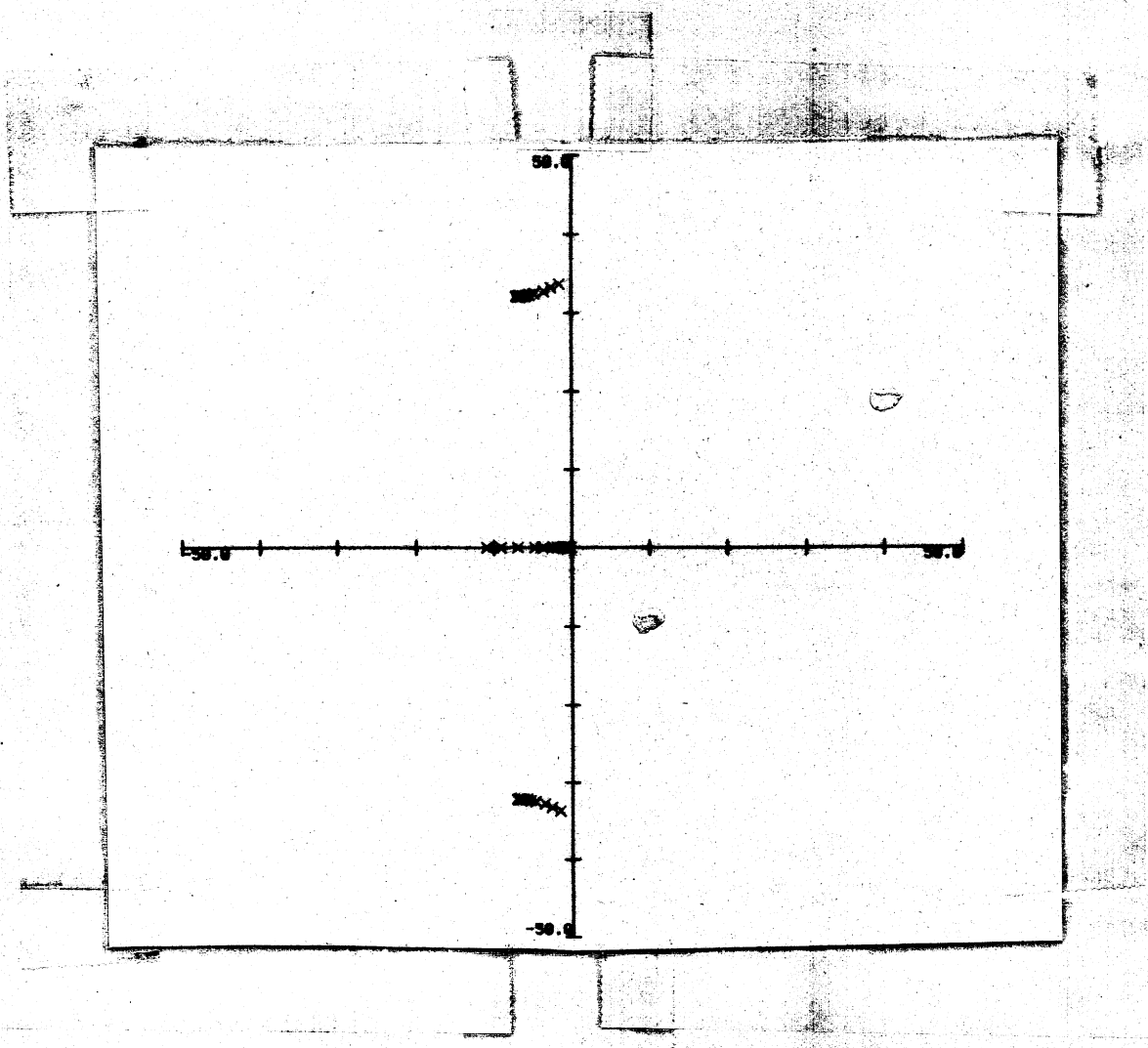


Figure 5.8: Root locus for soft environment as k_y varies

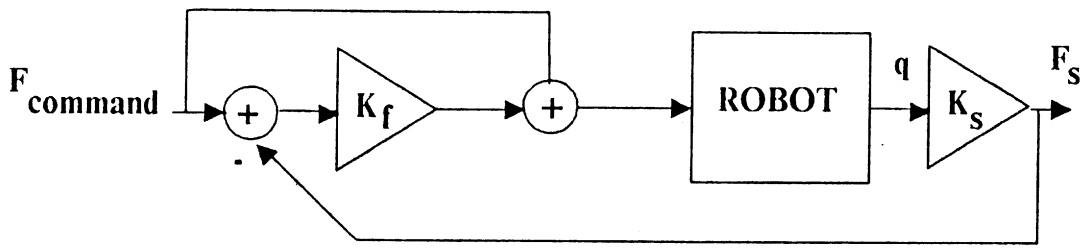


Figure 5.7: Simple Force Controller with $K_f = 0.222$

The Barry Wright FS6-120A 6-axis force/torque sensor was used to measure the contact force. As the block diagram shows, the tip force sensor feedback is a pure gain and has no dynamic compensation. For the results shown in this section I did not make any attempt to reduce the instabilities that were present. Such efforts are discussed in the Sections 5.3, 5.4, and 5.5.

Figure 5.8 shows the step responses of the simple force controller on three different surfaces using same gains and inputs. The negative bias shown on the top plot is from an offset drift in the force sensor and should be ignored. As expected, the robot becomes unstable when it comes in contact with the stiff aluminum surface. The spikes on the force data are produced as the robot bounces on this surface. The stability is improved when it is in contact with a rubber surface. Since the rubber used in this experiment is a hard rubber, it is still quite underdamped. But as the root locus (Fig. 5.6) of the section 5.2.2 indicated, the closed loop poles are still on the left hand plane, and the robot is stable and remains in contact with the surface. The last plot of Figure 5.8 is the force step response of the robot in contact with my fingers. Since my fingers (and the hand) are very compliant, the robot is stable under the simple force control.

This and the last sections have demonstrated some of the dynamic stability problems associated with a force-controlled system present even in a single joint system. In the next three sections, I will present several methods of remedying those stability problems.

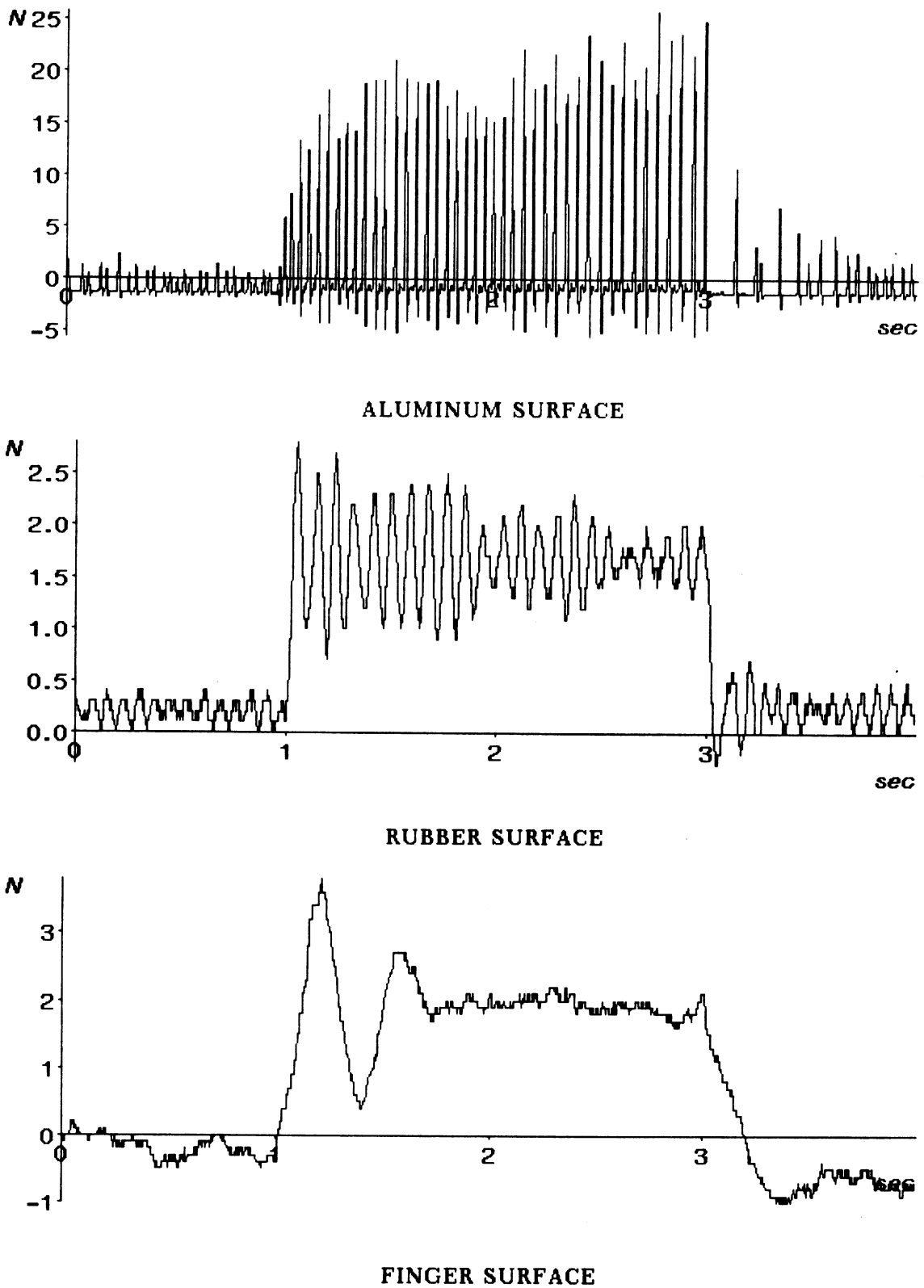


Figure 5.8: Step responses on three surfaces

5.3 Compliant Coverings

As shown in equations (5.3) and (5.7), the force sensor feedback is essentially a position feedback with the gain dominated by the stiffness k_E, representing the effective stiffness of the force sensor and the environment. In general, the stiffness of a wrist force sensor, k_s, is very high so that it has a good force resolution and dynamic range. Since the sensor mass is typically small, the effective stiffness k_E is a result of serial combination of the sensor stiffness k_s and the environment stiffness k_{env}. Then, to reduce k_E, one can either make the sensor soft or make the environment appear soft by attaching a compliant covering to the contact surface. Either approach gives the same result and the root locus of such a system will have the pattern shown in Figure 5.6 of Section 5.2.2. The experimental result for such a system can be seen in Figure 5.8 in the force step response on rubber or finger surface.

Whitney (1985) and Roberts (1985) have both suggested using soft sensors. But that may not be very practical since the softer the sensor is, the more the sensor will bend. Then the dynamic range of the sensor will be limited and it would also be difficult to control the tip position accurately due to large sensor deflections (Roberts, 1985). On the other hand, if a thin compliant covering is used on top of the force sensor, the environment will always seem soft to the force controller. This method preserves the large dynamic range of a stiff force sensor and also improves stability.

Using compliant coverings is not unlike how human arms are made. In fact, the skin on our arm helps us greatly in force control. It absorbs the energy during the collision phase of a contact with a stiff environment and it also acts as a contact sensor to give us some relative position information. In that respect, it would be even better to make the compliant coverings to be the tactile sensors of the type discussed in (Siegel, Garabieta, and Hollerbach, 1986).

5.4 Adaptation to the Environment Stiffness

Another method in dealing with a stiff environment is to somehow estimate the environment stiffness and then to incorporate the knowledge of the environment in the controller to achieve a stable and robust behavior. One simple method of

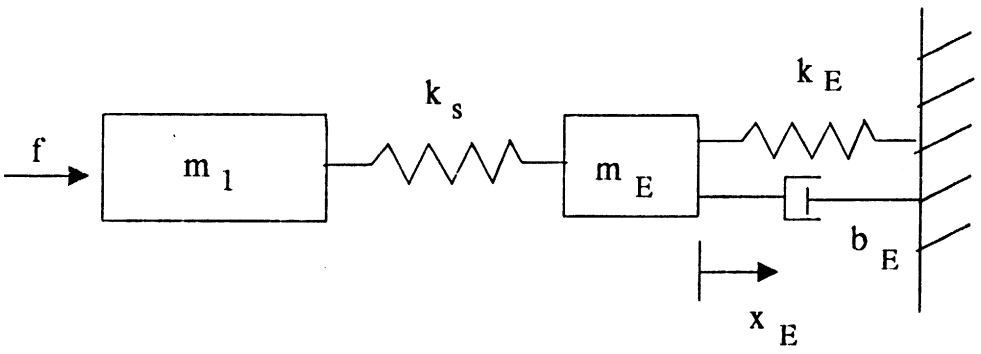


Figure 5.9: Fourth order model of the robot and the environment

incorporating such knowledge is to vary the velocity gains according to the environment stiffness and damping, so that the system remains stable without being too sluggish. The problem, then, is first to identify the dynamic characteristics of the environment. Here I will present two methods, one simpler than the other.

Let's consider a fourth order model for a situation where a manipulator is in contact with its environment (Figure 5.9). The force sensor is modelled as a spring of stiffness k_s. The environment is modelled as a spring, a damper, and a mass. The mass m_E is the manipulator load, a tool, for example. In practical situations, if the system becomes unstable, the manipulator will lose contact with the environment. In this section, however, I am considering a simplified situation where the manipulator remains in contact. The former situation is more complex and will not be addressed in this thesis.

The behavior of the system in Figure 5.9 is described by,

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_E \\ \dot{x}_s \\ \dot{x}_E \end{bmatrix} = \begin{bmatrix} 0 & 0 & -k_s & 0 \\ 0 & -b_E/m_E & k_s & -k_E \\ 1/m_1 & -1/m_E & 0 & 0 \\ 0 & 1/m_E & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_E \\ x_s \\ x_E \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} f \tag{5.10}$$

where

$$p_1 = m_1 v_1 = m_1(\dot{x}_s + \dot{x}_E), \quad p_E = m_E v_E = m_E \dot{x}_E$$

and x_s is the displacement of the sensor spring. Ignoring the viscous damping in the sensor, the equation for the dynamics of the environment only is decoupled as,

$$f_s = k_s x_s = m_E \ddot{x}_E + b_E \dot{x}_E + k_E x_E \tag{5.11}$$

The measurement of f_s is available using a wrist force/torque sensor. Although x_E cannot be measured directly, it is related by,

$$x_{robot} - x_s = x_E \tag{5.12}$$

where x_{robot} is the movement of the manipulator from the point of contact with the environment and $x_s = f_s/k_s$. However, typically we will not be able to measure the other state, $\dot{x}_E$.

5.4.1 Least Squares

One method of identifying the dynamics of the environment is to use a least squares algorithm. If the measurements of $\ddot{x}_E$ and $\dot{x}_E$ as well as f_s and x_E were available, then we can use straightforward least squares algorithm on the continuous time equation (5.11). However, as discussed above, at best we only have measurements of f_s and x_E. In this situation, the continuous time equation for the dynamics of the environment can be transformed to a discrete version by a number of different methods (Franklin and Powell, 1980). Then, using least squares method, the coefficients for the discrete equation can be estimated. The actual continuous time parameters can then be obtained from these discrete coefficients by simple algebraic transformations. In the rest of this section, these steps are derived, and both simulation and experimental results are presented.

Derivations

The Laplace transform of the continuous time equation (5.11) for the dynamics of the environment is

$$f_s(s) = k_s x_s(s) = (m_E s^2 + b_E s + k_E) x_E(s). \tag{5.13}$$

For the bilinear transformation method of converting a continuous time system to the discrete time domain, the relation between s and z is given by

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right). \tag{5.14}$$

Substituting (5.14) into (5.13),

$$\frac{x_E(s)}{f_s(s)} = \frac{1 + 2z^{-1} + z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \tag{5.15}$$

The difference equation corresponding to (5.15) is

$$f_s[n] + 2f_s[n-1] + f_s[n-2] = a_0 x_s[n] + a_1 x_s[n-1] + a_2 x_s[n-2]. \qquad (5.16)$$

Then, using a well documented recursive least squares method (Ljung, 1983), the coefficients a_0, a_1, and a_2 can be estimated. The continuous time parameters, then, are computed from the estimated coefficients by the following algebraic relations:

$$b_E = \frac{T}{4}(a_0 - a_2)$$

$$m_E = \frac{T^2}{16}(a_0 + a_2 - a_1)$$

$$k_E = \frac{1}{2}(a_1 + \frac{8m_E}{T^2}) \qquad (5.17)$$

Since the bilinear transformation is not an exact transformation, the above derivation is also approximate. But for fast sampling time, the transformation is quite accurate.

Simulation

The simulation results of the recursive least squares estimation is shown in Figure 5.10. The signal is a step response with some random noise and the sampling frequency is $200\,Hz$. As shown in the figure, the stiffness estimate reaches the actual value quickly, within 0.05 second or 10 samples. The damping estimate reaches the final value in 0.1 second. The estimate for the mass settles to the final value after 0.2 second, and there is a significant bias error in this estimate. These behaviors are expected. The b_E and m_E parameters are mainly affected by the $\dot{x}_E$ and $\ddot{x}_E$ signals respectively, whose measurements are not available directly. Therefore, the discrete algorithm is essentially numerically differentiating and double differentiating the x_E measurements in order to arrive at estimates of b_E and m_E. The estimates for these parameters then should not be as good as the estimate of k_E. For stability, the stiffness parameter is the most important and as shown in the simulation results, it may be possible to estimate it quickly enough for the controller to adapt to the changing environment conditions.

Experiments

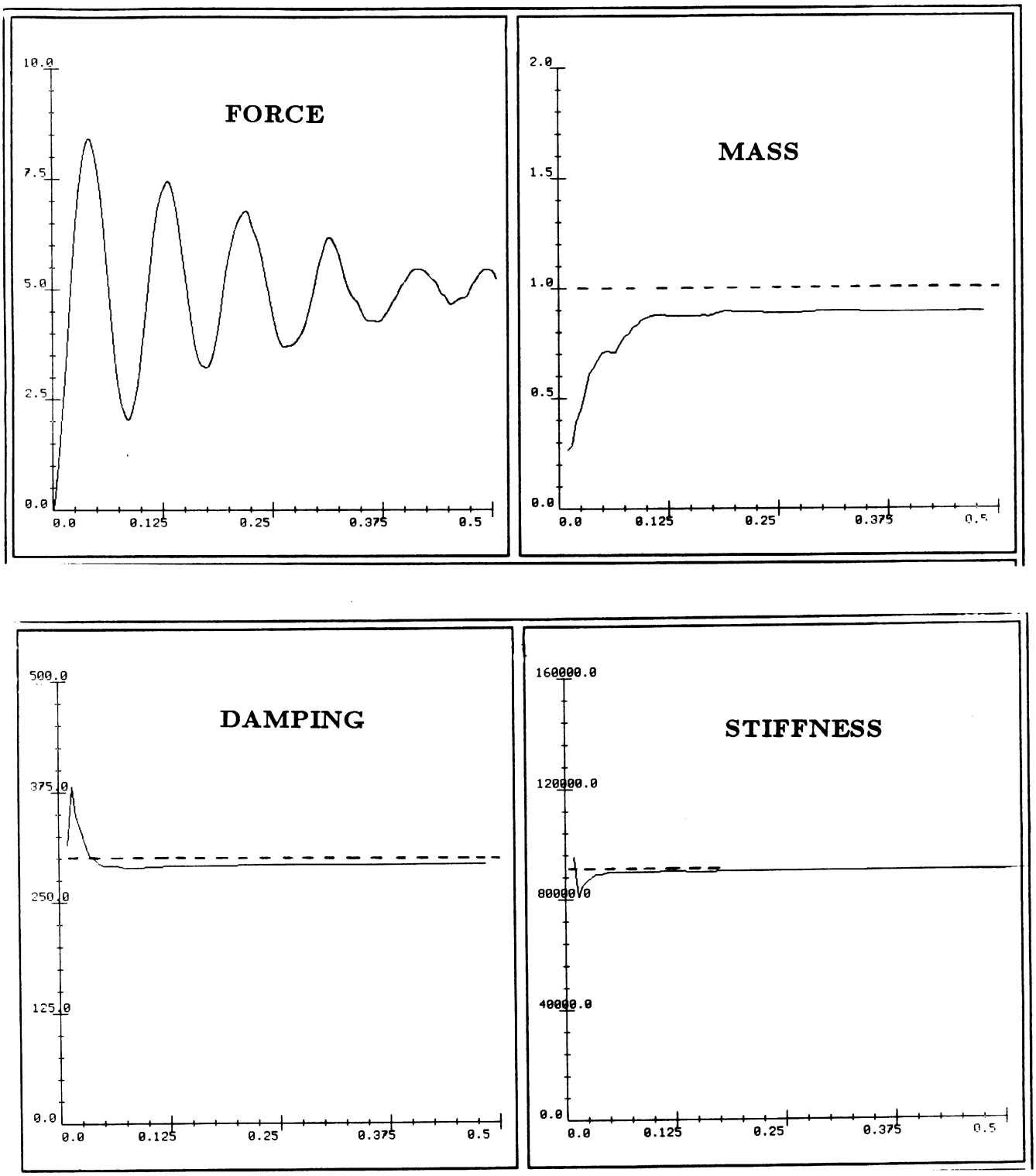


Figure 5.10: Simulation of Estimation of Environment Dynamics

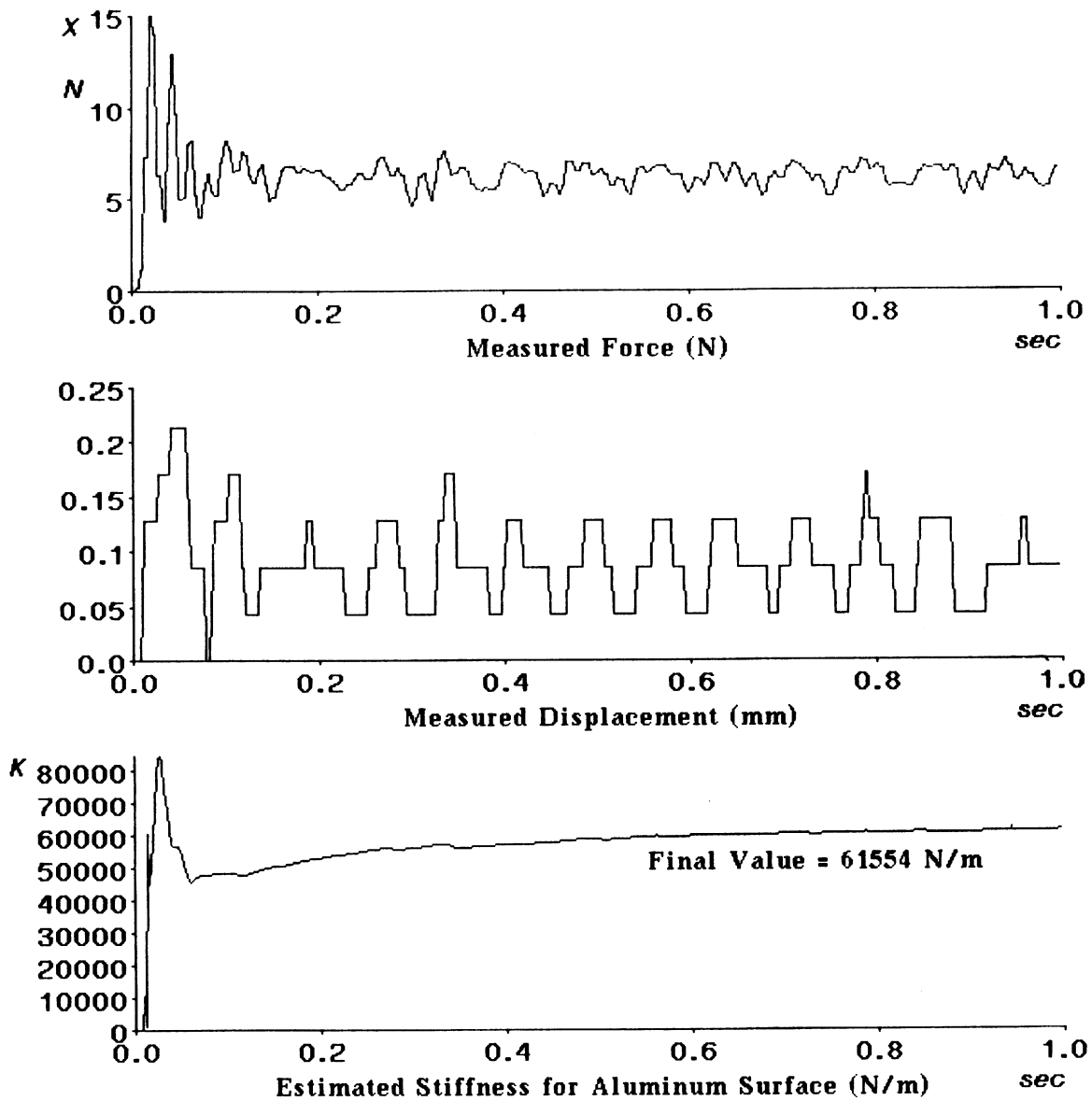


Figure 5.11: Experimental Stiffness Estimate on Aluminum Surface

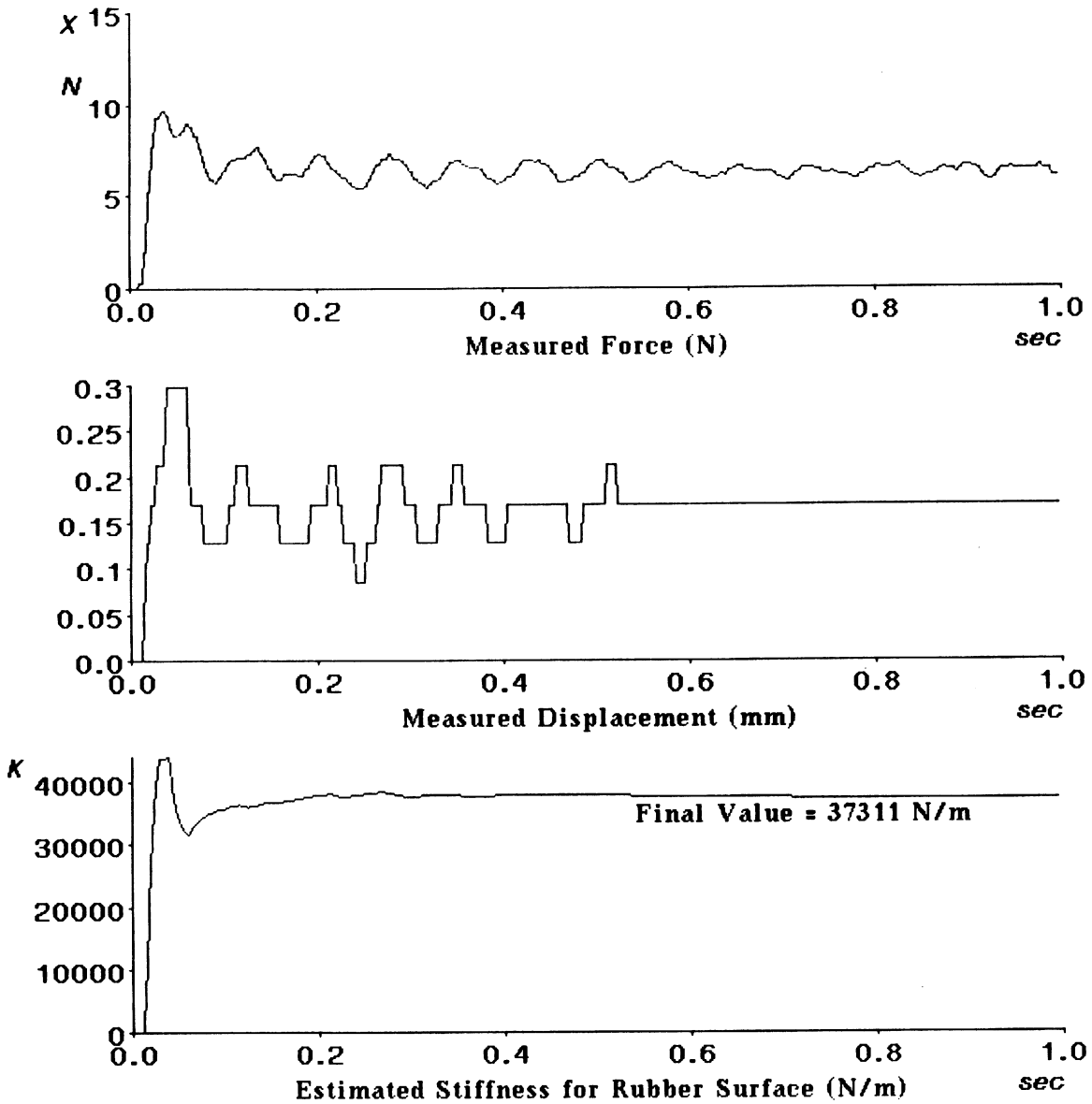


Figure 5.12: Experimental Stiffness Estimate on Rubber Surface

Estimation using the least squares method was tested on two different surfaces, aluminum and rubber. Figures 5.11 and 5.12 show the step force responses for one link robot, the position displacements, and the estimation results of the effective environment stiffness of an aluminum and a hard rubber surface, respectively. The measurements of the position displacements were from the resolver at the joint. Since the stiffness of the sensor is very high ($>> 10^6 N/m$), I neglected the deflections of the force sensor in the position measurements.

The results show that the estimated value of stiffness for the hard rubber surface is 60% of that for the aluminum surface. Although the actual stiffness of aluminum may be much higher than the estimate, the effective stiffness of the aluminum surface and the end effector (a bearing attached to sensor) together is lower than the actual stiffness. In control, it is this effective stiffness that is important for stability considerations. The estimates for the mass and the damping were not consistent and are not shown. As discussed in the simulation section, this was expected since those estimates depend on the derivatives of the position data. For the examples presented, the positional displacements are too small for the derivatives to be accurate. With the 16 bit resolution resolver, the maximum displacements are only 2.5 bits for aluminum and 3.5 bits for rubber.

5.4.2 Adaptive Observer

For the environment equation (5.11), an adaptive observer technique (Narendra and Kudva, 1974; Narendra and Valavani, 1976; Shih, 1985a) may also be used to estimate the mass m_E, the stiffness k_E, and the damping constant b_E of the environment. Actually m_E may already be known through load identification, but I will continue to treat it as an unknown in the following analysis. Following the method presented by Narendra and Kudva (1974), a stable adaptive observer can be formulated for the equation (5.11). Dividing by m_E, (5.11) can be rewritten as,

$$Df_s = \ddot{x}_E + \beta \dot{x}_E + \Omega x_E \tag{5.18}$$

where

$$D = \frac{1}{m_E}, \quad \beta = \frac{b_E}{m_E}, \quad \Omega = \frac{k_E}{m_E}.$$

Define additional state variables y and w as,

$$\dot{y} = -\lambda y + x_E$$

$$\dot{w} = -\lambda w + f_s \tag{5.19}$$

In terms of the new state variables, (5.18) can be written as,

$$\dot{x}_E = -\beta x_E - \Omega y + Dw + \lambda x_E + \lambda(\beta - \lambda)y \tag{5.20}$$

The derivation of (5.20) is in Appendix 3. Equations (5.19) and (5.20) together represent a non-minimal realization of the original equation (5.18). In a matrix form, this equivalent non-minimal realization can be written as,

$$\begin{bmatrix} \dot{x}_E \\ \dot{y} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \lambda - \beta & \lambda(\beta - \lambda) - \Omega & D \\ 1 & -\lambda & 0 \\ 0 & 0 & -\lambda \end{bmatrix} \begin{bmatrix} x_E \\ y \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} f_s \tag{5.21}$$

All of the states in (5.21) can be either measured or computed by (5.19). Then, following the structure of (5.20), the adaptive observer equations are formulated.

$$\dot{\hat{x}}_E = -\hat{\beta} x_E - \hat{\Omega} y + \hat{D} w + \lambda x_E + \lambda(\hat{\beta} - \lambda)y - \alpha(\hat{x}_E - x_E)$$

$$\left.\begin{aligned} e &= \hat{x}_E - x_E \\ \tfrac{d}{dt}\hat{\beta} &= \gamma_1 e x_E \\ \tfrac{d}{dt}\hat{\Omega} &= \gamma_2 e y \\ \tfrac{d}{dt}\hat{D} &= -\gamma_3 e w \end{aligned}\right\} \quad \gamma_i > 0 \tag{5.22}$$

This adaptive observer can be proved to be stable using Lyapunov analysis, given sufficiently rich inputs. There exists a discrete version of the above formulation (Kudva and Narendra, 1974) and also other forms of adaptive observer that may be useful for this application (Shih and Lang, 1985a, 1985b).

There may be other methods for identifying the environment dynamics. One possible method is to estimate the environment stiffness k_E by estimating the frequency of oscillation caused by interaction with the environment. The frequency information may be obtained by taking the FFT of the force measurement f_s.

5.4.3 Feasibility

The above methods show that it is theoretically possible to identify the dynamics of the environment in order to use them in the control law. However, as shown in the experimental estimation results, using practical sensors, we may not have

enough joint position sensor resolution to measure x_E accurately. Assuming that the force sensor is much stiffer than the environment, for a 16 bit resolution joint angle measurements and $0.5\,m$ lever arm, a static analysis ($f = k_E x_E$) shows the following upper limits in the stiffness of an environment that can be estimated:

force	k_E
$1N$	$2 \times 10^4\,N/m$
$10N$	$2 \times 10^5\,N/m$

This shows that even for rather optimistic 16 bit resolution measurements, unless the contact force is undesirably large, stiff environments cannot be identified accurately. This characteristic may limit the usefulness of this approach in dealing with stiff environments.

5.5 Joint Torque Control

I have so far assumed that a force-controlled system uses a high gain feedback of the signal from a tip force sensor, which is typically characterized as a very stiff spring. From the stability point of view, feedback control using the output of such a force sensor is equivalent to a very large gain position feedback which can lead to instability. Stability can be improved if the loop gain is reduced. Adding a compliant covering, as discussed in Section 5.3, was one such approach. Lowering the force feedback gain also reduces the loop gain but also has the undesirable effect of deteriorating the force resolution. Another useful approach is to achieve force control without using the tip force sensor in the feedback loop, hence removing the high gain component from the closed loop system. Instead, the end effector force can be controlled by relying on the measurements of and the ability to command joint torques accurately. The loop gain is 0 since such a control method is an open-loop control from the point of view of interaction forces at the end effector. However, in practical systems, a closed-loop torque servo is implemented at each joint so that the joint torque can be specified accurately.

This approach has been investigated previously by Wu and Paul (1980) and also by Luh, Fisher, and Paul (1983) using strain gauges on the motor shaft of a geared system, but they did not use their experimental joint torque control method to perform any active force control. Instead, the joint torque feedback was used

mainly to reduce frictional effects at the joints for the Stanford Arm (Luh, Fisher, and Paul, 1983). For the direct drive arm, since there is very little friction at the joints, the measurements of joint torques can be obtained by measuring the motor currents. In Chapter 3, I used this capability to compute joint torques in order to estimate the link inertial parameters.

Wu and Paul (1980) presented a good comparison between using wrist force sensing and joint torque sensing in implementing force control. Wrist sensing provides accurate force/torque measurements at the hand; but because the robot structure is inherently a low bandwidth flexible system and the sensor is situated at the end of this structure, a high gain feedback will produce instability as shown in Section 5.2. Therefore, only a slow closed loop system can be implemented stably using a wrist sensor. On the other hand, since joint torque sensors are situated before the low bandwidth robot structure, a high bandwidth torque inner loop can be implemented around each joint. But since the sensors are not at the hand, the hand forces and torques cannot be inferred as accurately as the wrist sensing.

Since both the wrist sensing and the joint sensing have good and bad features, one reasonable method is to combine the two methods to provide a stable high bandwidth force-controlled system. The high bandwidth open-loop joint torque control with inner torque servo loop will provide stability and fast response, and the lower bandwidth outer loop with wrist force sensing will provide extra accuracy. To the best of my knowledge, this method has not been implemented before, probably due to the lack of a suitable manipulator. In the direct drive arm, one has this capability since torques can be commanded quite accurately at each joint.

In the rest of this section, the stable property of an open-loop force controller using joint torque sensing, is discussed in more detail, and the results of one-link force control experiments are presented demonstrating the performances of using joint torque sensing and also the combination of joint torque and wrist force sensing.

5.5.1 Dominant Pole

Without the wrist sensor in the force feedback loop, the dynamics of a simple manipulator in contact with its environment is given by

$$f - k_E x = m\ddot{x} + b\dot{x} \tag{5.23}$$

If this system is controlled purely by commanding the force or the torque at the joint, i.e. in open loop mode, the response should be very underdamped since k_E may be high for a stiff environment. One simple classical compensation method of improving stability is to create a dominant pole in the loop transfer function (Roberge, 1975). This can be done simply by putting a low-pass filter in the forward path so that

$$\tau = f\frac{a}{s+a}, \tag{5.24}$$

where τ is the actual input torque (or currents) to the actuator and a is much less than the resonant frequency of the original system. Then, the total system dynamics is described by

$$x(s) = f\left(\frac{a}{s+a}\right)\left(\frac{1}{ms^2+bs+k_E}\right). \tag{5.25}$$

Because of the dominant pole at $s = a$, this compensated system behaves in a much more stable manner despite the two high frequency underdamped poles. Figure 5.13 compares the step responses of the above system (5.25) to the original system without the dominant pole (5.23).

For real actuators, the explicit low-pass filtering of the input signal may not even be necessary, since the combined system of the amplifiers and the manipulator structure behaves essentially as a low-pass filter. For the MIT Serial Link Direct Drive Arm, there is a an analog current loop at each amplifier to the motor which, together with the rotor inertia, has the cutoff frequency at approximately 30 Hz. Because of this low-pass characteristic, explicit low-pass filtering was not necessary to achieve stability. However, even for the direct drive motors, there are some nonlinear characteristics as deadzone for small torques (Asada, Youcef-Toumi, and Lim, 1984), cogging, and other effects due to the imperfect commutation circuitry. Some of these nonlinear characteristics can be reduced by implementing another torque feedback loop at each joint. As discussed in Chapter 3, for the DDARM the torque at each joint is computed from the three phase currents by the following relation,

$$\tau = K_T(I_a \sin(n\theta + offset) + I_b \sin(n\theta + 120 + offset) + I_c \sin(n\theta + 240 + offset)) \tag{5.26}$$

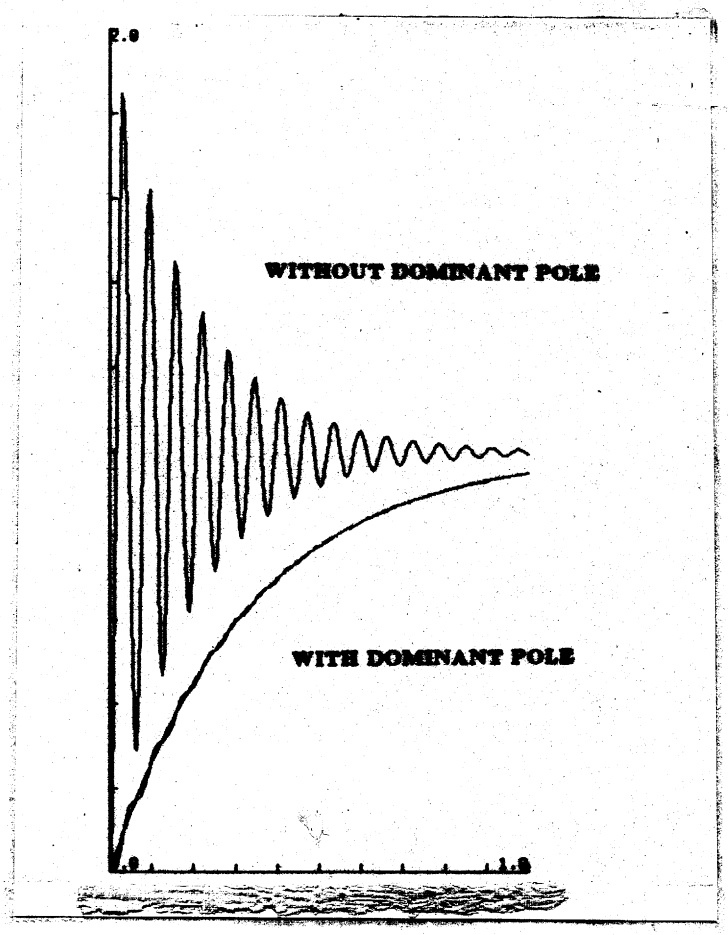


Figure 5.13: Step responses of (5.23) with and without the dominant pole

where

$$n = \begin{cases} 15 & \text{for Joint 1} \\ 9 & \text{for Joints 2 and 3} \end{cases} \qquad offset^{1} = \begin{cases} 60° & \text{for Joint 1} \\ -33.3 & \text{for Joint 2} \\ -108 & \text{for Joint 3} \end{cases}$$

Figure 5.14 shows the block diagram of the joint torque servo and also the comparison between the commanded and the measured torques for the third joint of the DDARM with and without the added joint torque feedback. The pole of the lag filter shown in the figure is separate from the dominant pole discussed above. The dominant pole refers to the low pass structure of the whole actuator system, including the torque servo, amplifier, and the motor. As the plots in Figure 5.14 show, torques can be commanded accurately at each joint.

The analysis of this section shows that the force control via joint torque sensing is stable and well behaved, due to the open-loop structure and the dominant pole created either explicitly or inherently by the amplifier and the robot structure. But as discussed earlier, it is also desirable to include the wrist force sensing to improve the accuracy of the force control system. In such a case, using the same analysis as above, one should also use a low pass filter in the control loop involving the wrist force sensor feedback, thus creating a dominant pole. The pole of this wrist sensing loop should be at a much lower frequency than the pole of the joint torque path, so that the stability is not affected. Then, the combined multi-feedback loop system should have the stability and the high bandwidth from the joint torque control mechanism, and the steady state accuracy from the wrist force control mechanism. The block diagram of a such a system is shown in Figure 5.15.

5.5.2 One Link Force Control Experiments

The two suggested stable force control methods were implemented on the third link of the DDARM. The first method, using joint torque sensing only, is an open loop method of commanding the appropriate torque at the joint and relying on the dominant pole provided by the joint torque servo and the amplifier to keep the system stable in contact with stiff environments. In this case the wrist force sensor is used only to record the force data at the tip of the robot and is *not* used in the

[1] The offsets are due to abnormalities in the commutation circuitry as well as offsets in resolvers.

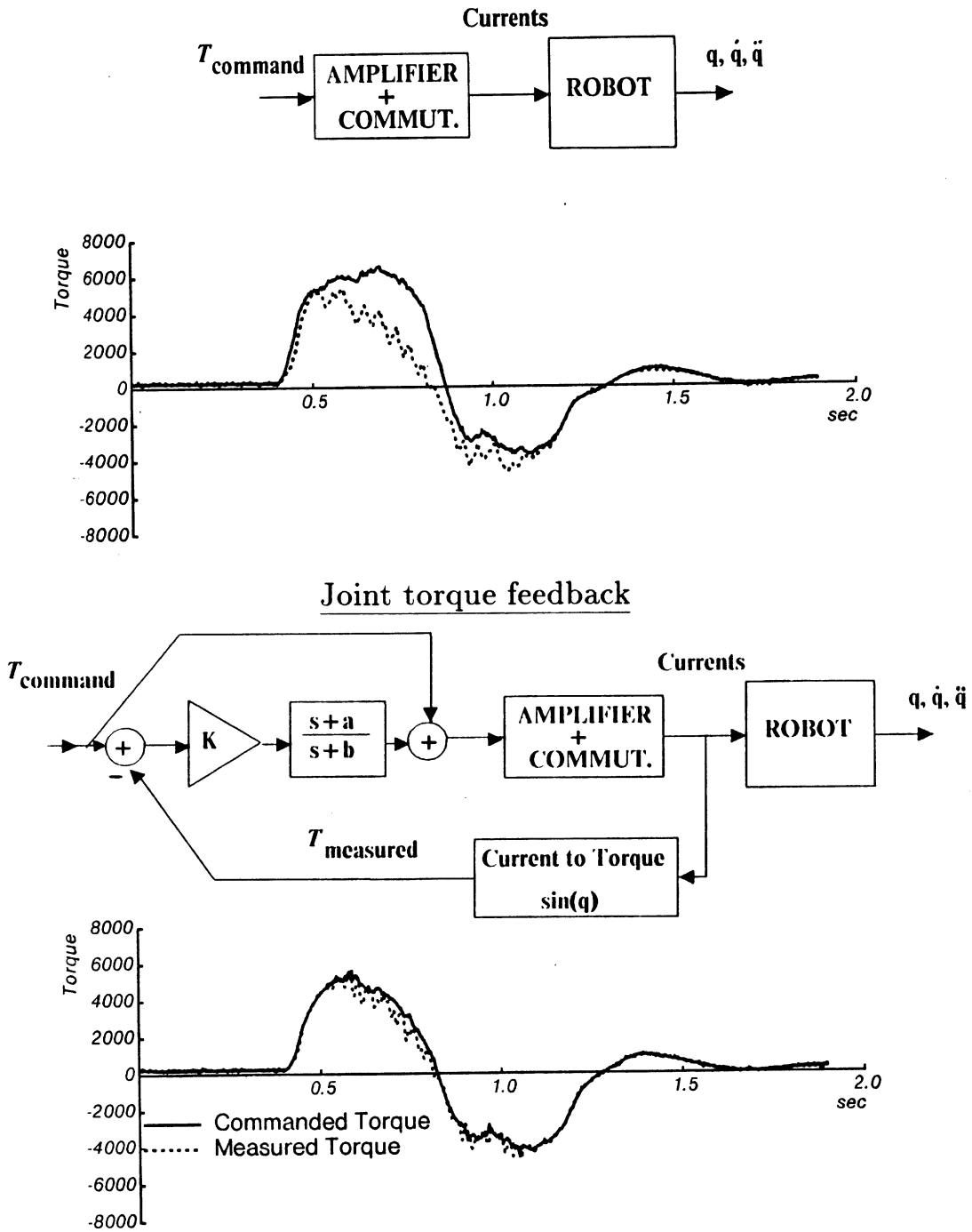


Figure 5.14: With and without explicit joint torque servo

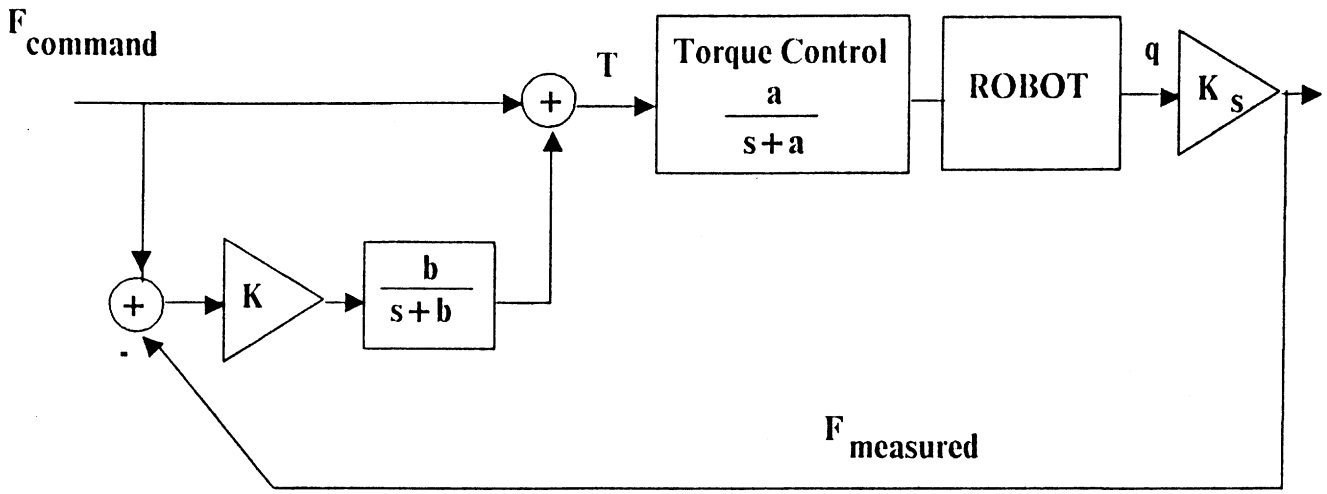


Figure 5.15: Structure of a general force controller

feedback. The second method uses both the joint torque sensing and the wrist force sensing as shown in Figure 5.15. The two methods are summarized below.

1. Using joint torque sensing only:

$$\tau = f_{desired} \cdot l_3, \quad l_3 = 0.4445\, m$$

2. Combination of joint torque sensing and wrist force sensing:

$$\tau = f_{desired} \cdot l_3 + k_f \left(\frac{b}{s+b} \right) (f_{desired} - f_{measured})$$

The third link has an explicit joint torque feedback loop shown in Figure 5.14 implemented digitally at 500 Hz. The sampling frequency of 500 Hz was used throughout the one link experiments.

Force Step Response

In Figure 5.16, a stable force step response of the second method is compared to the unstable response of a simple pure gain force feedback without the dominant pole for the robot in contact with a stiff aluminum surface. The negative bias shown on the top plot is from an offset drift in the force sensor and should be ignored. The plots are untouched without any low-pass filtering of the data to reduce the noise

for plotting purposes. As the figure shows, although there is a significant amount of noise in the force data, the step response of the compensated system is definitely stable, whereas the response of the system without the proper compensation is unstable and the manipulator bounces on the environment surface.

Figure 5.17 shows the force step responses of the two stable methods discussed above. As expected from the analysis, there is no noticeable difference in dynamic behavior between the two controllers; but the second method, combining the joint torque sensing and the wrist force sensing, has much better steady state accuracy. For either method, the accuracy is worse for the lower force inputs since the nonlinear characteristics (such as dead zone and cogging) of the motor and the amplifier for small torque levels are more severe (Asada, Youcef-Toumi, and Lim, 1985). The commanded force levels ($10\,N$ and $15\,N$) used in the experiments are less than 2% of the capacity of the DDARM, which can exert greater than $500\,N$ of force at its tip.

Performance Tests

Step inputs are not really the best inputs in testing the performance of force control methods. Their high frequency contents and the large initial torques that are required at the discontinuities can excite undesirable structural modes. But it does qualitatively test stability. For evaluating performance of force controllers, better methods are:

1. Sine wave force command with the manipulator in contact with the stationary environment,

2. Constant force command with the manipulator in contact with the environment moving in a sinusoid trajectory.

Both of these test inputs may give us information about bandwidth.

Sine Wave Force Command The responses to a $1Hz$ sine wave force command are given in Figure 5.18 for the two stable methods. Similar to the step response results, the accuracy for the second method is much greater than the first method. But again, for either method, neglecting the bias errors, the manipulator follows the sine wave command faithfully. Figure 5.19 shows the responses of the

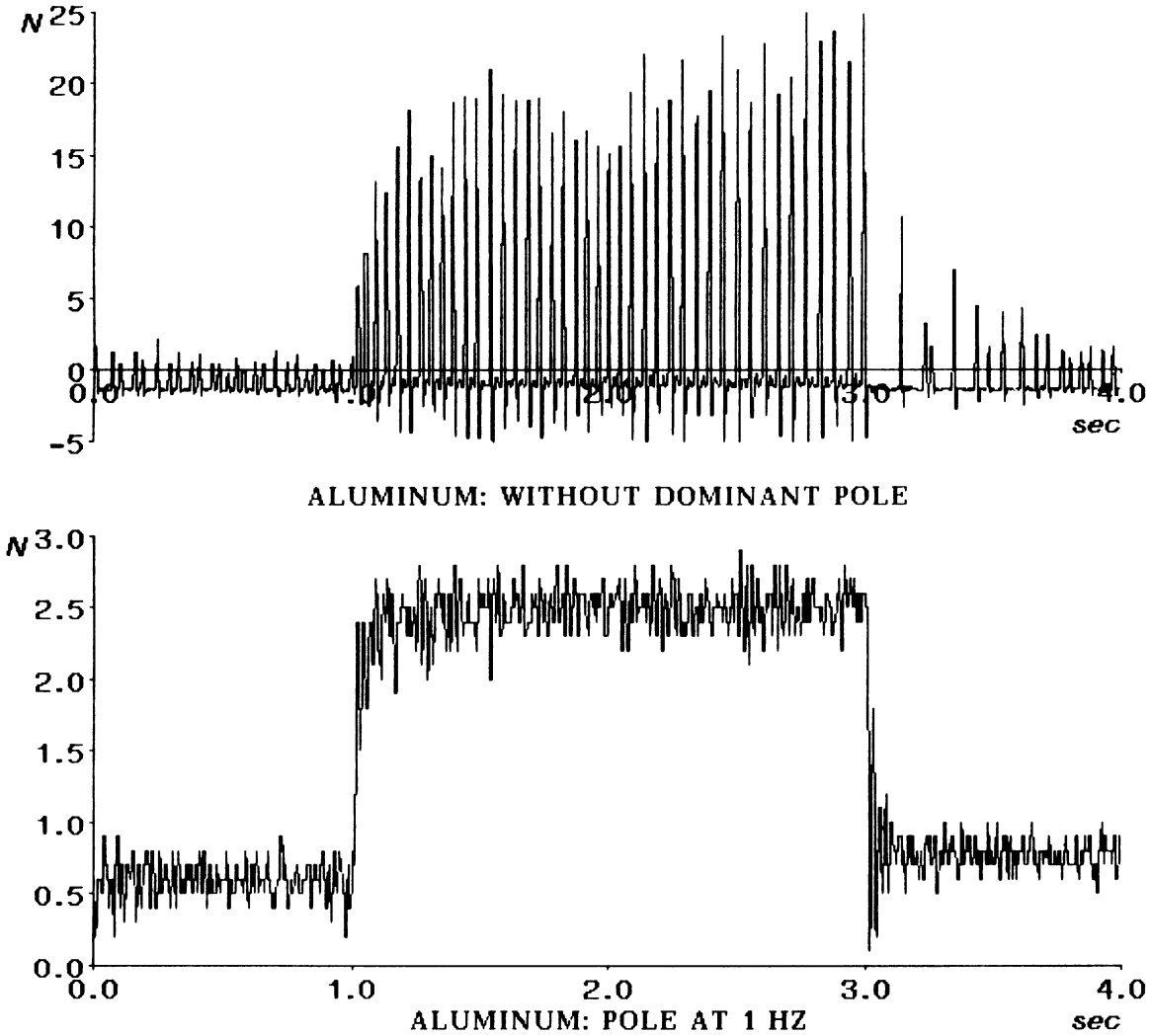


Figure 5.16: Force step responses with and without the dominant pole

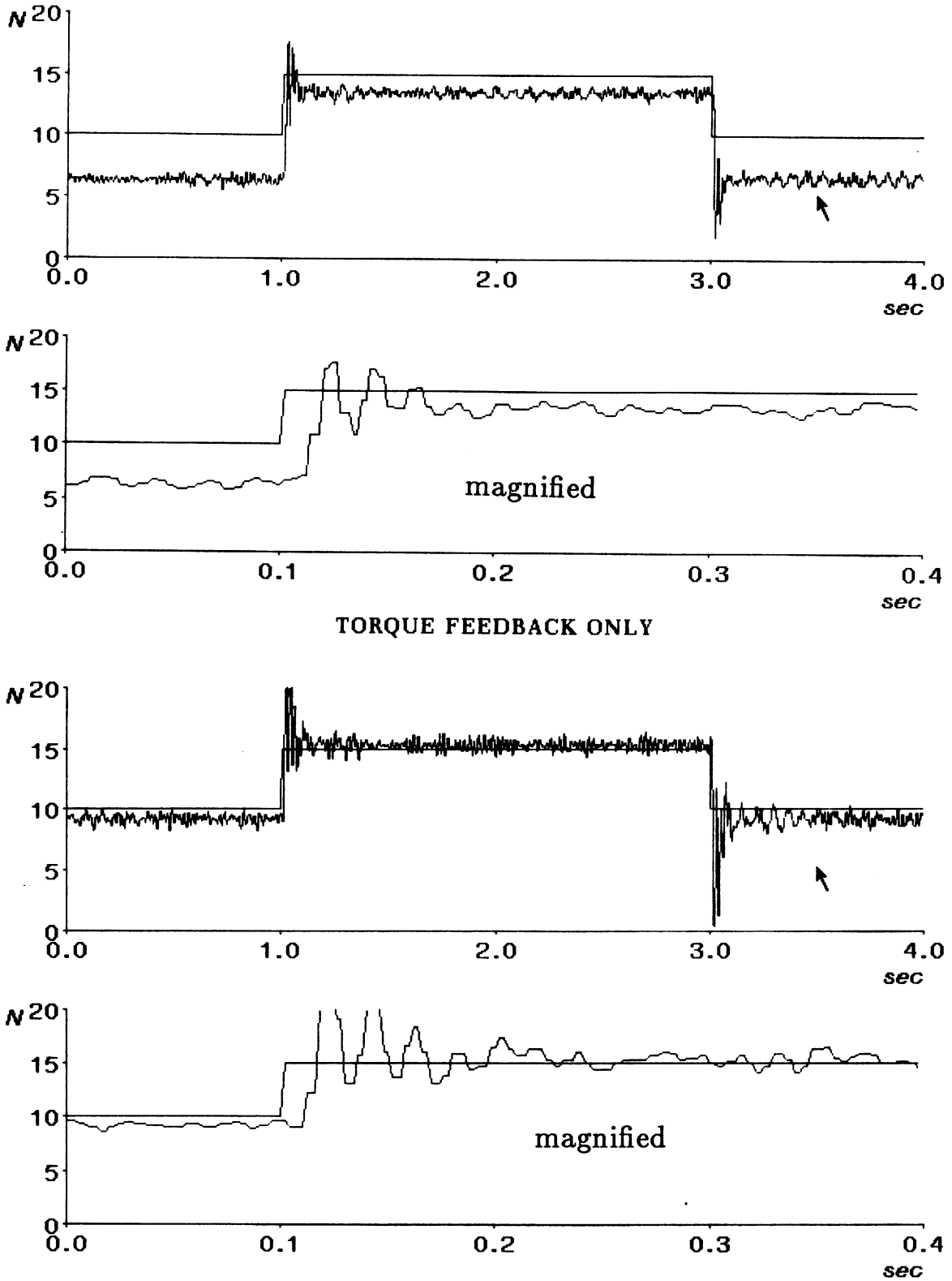


Figure 5.17: Force step responses for the two stable methods overlayed on the step inputs

combined method to sine wave force commands of frequencies up to $20Hz$. Although there is a noticeable lag in the measured data for higher frequencies, the manipulator still has no trouble following the sine wave command. Therefore, using the usual definition of bandwidth (frequency at which the output magnitude is reduced by 3dB from the DC value), the bandwidth tested in this way is greater than $20Hz$.

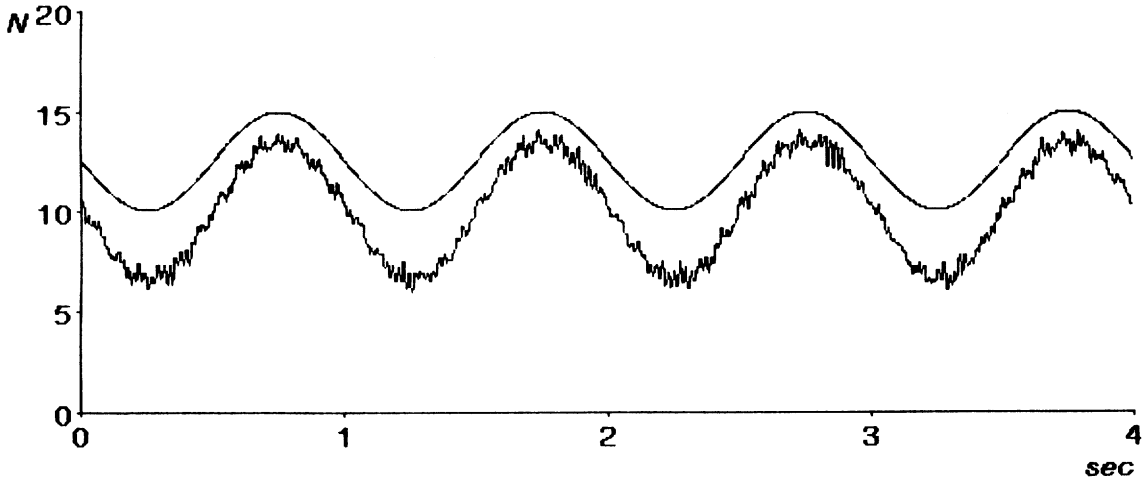
Constant force command with positional disturbances The second performance test, a constant force command while the contacting environment is moving, is made with an eccentric cam on a gear motor as shown in Figure 5.20. The circular cam is attached to the gear motor at a point $0.5\,inch$ off from the center providing an approximate sine wave positional disturbance to the force controller. The responses of the two methods are plotted on Figure 5.21. The second method is able to follow the cam accurately with the desired force. Neglecting the high frequency noise, the low frequency variation in the measured force is within 10% of the desired $12.5N$. Higher cam speed resulted in larger variations.

It is difficult to define a measure of bandwidth for this test because this test is really a disturbance rejection test. The movements of the cam provided a positional disturbance to the force-controlled system as shown in the simplified block diagram of Figure 5.22. Therefore, a reasonable measure of this disturbance rejection is the frequency w_{dr} at which the variation is 10% (5.27).

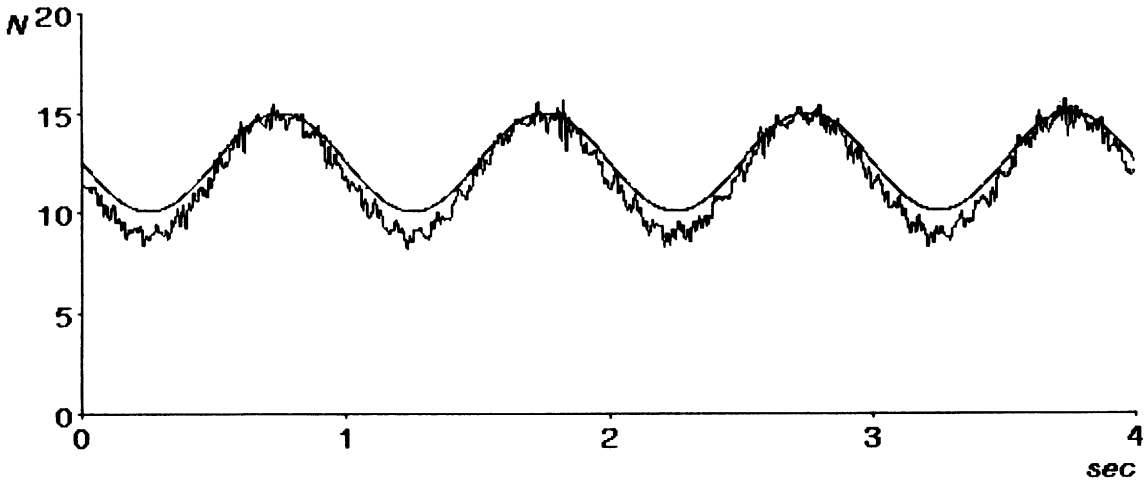
$$\left.\frac{f_{command} - f_s}{f_{command}}\right|_{w_{dr}} = 10\% \tag{5.27}$$

Under this measure, this system is well behaved up to approximately $0.8Hz$. The response of the first method is not as accurate at this cam speed, suggesting a slightly lower frequency limit.

The frequency limit determined by the positional disturbance is much lower than the bandwidth determined by the sinusoid force trajectory testing. This is not surprising since for the sinusoid trajectory tests, the movement of the link is infinitesimal; whereas for the cam tests, the controller has to move the large link inertia over a significant distance. It is not, however, clear which measurement of performance is more useful. For applications that involve following undulated surfaces, the positional disturbance test is more relevant. But for peg-in-hole types



TORQUE FEEDBACK ONLY: 1 Hz SINE RESPONSE



FORCE TORQUE FEEDBACK: 1 Hz SINE RESPONSE

Figure 5.18: $1Hz$ sine wave responses for the two methods with the commanded inputs overlayed

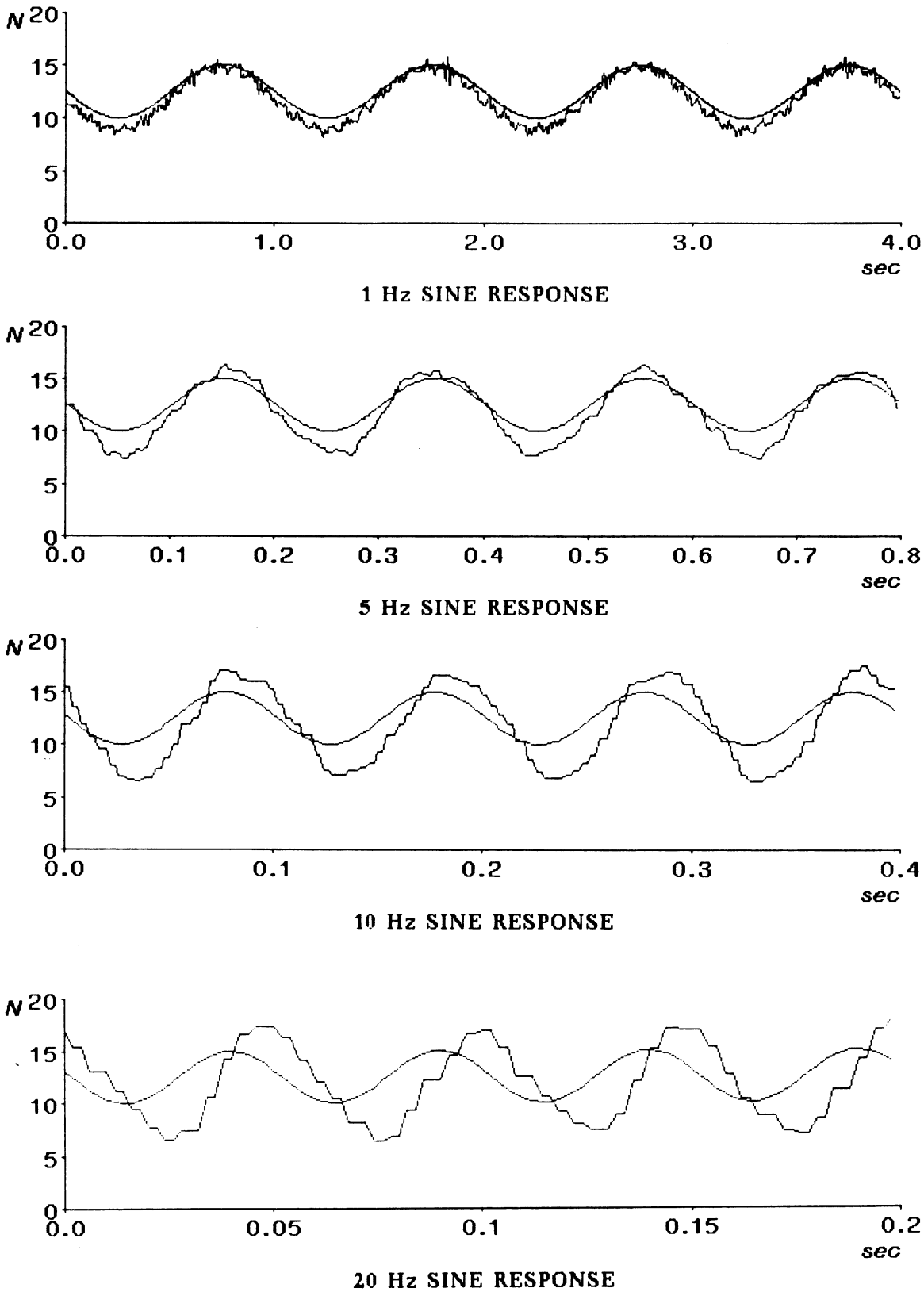
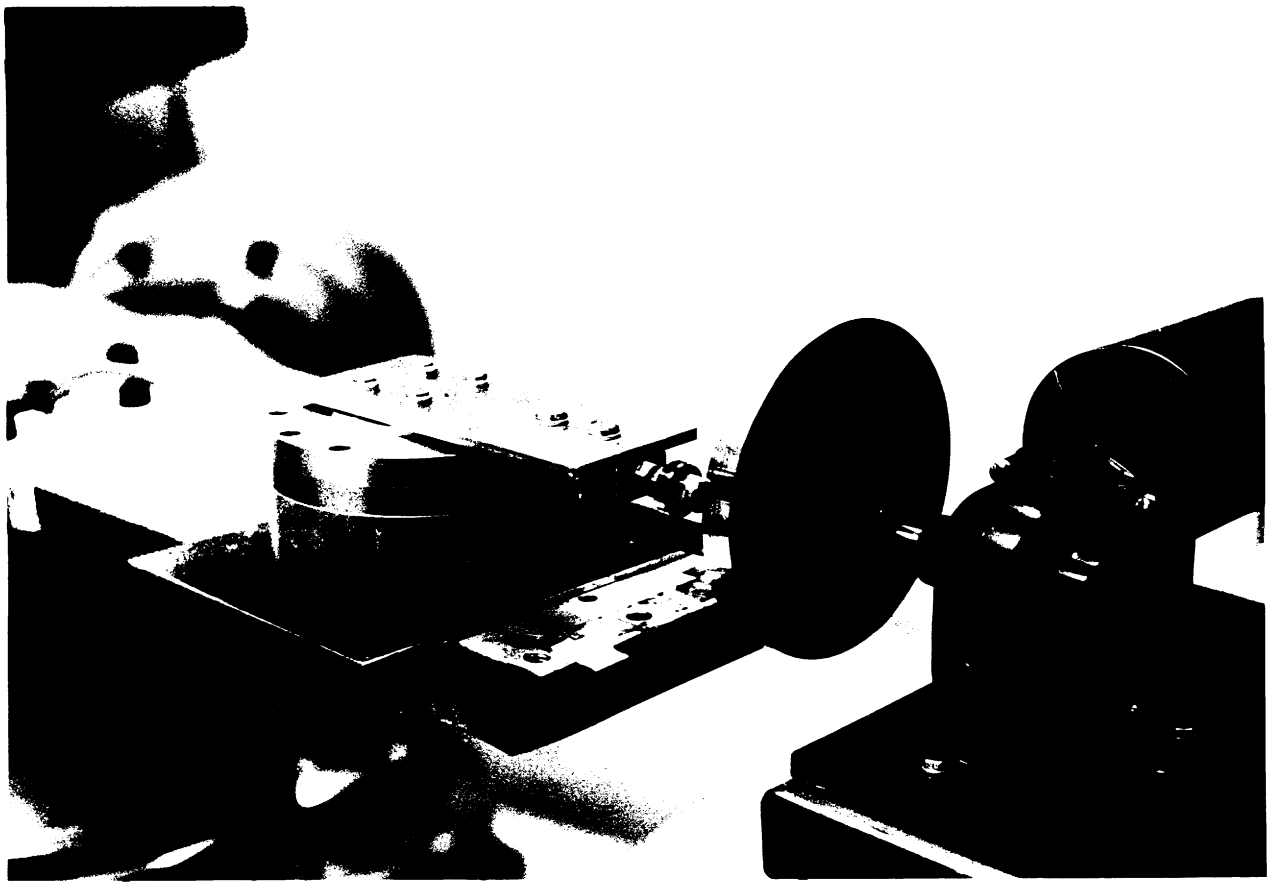


Figure 5.19: Sine wave responses for the second method using both joint torque and wrist force sensing



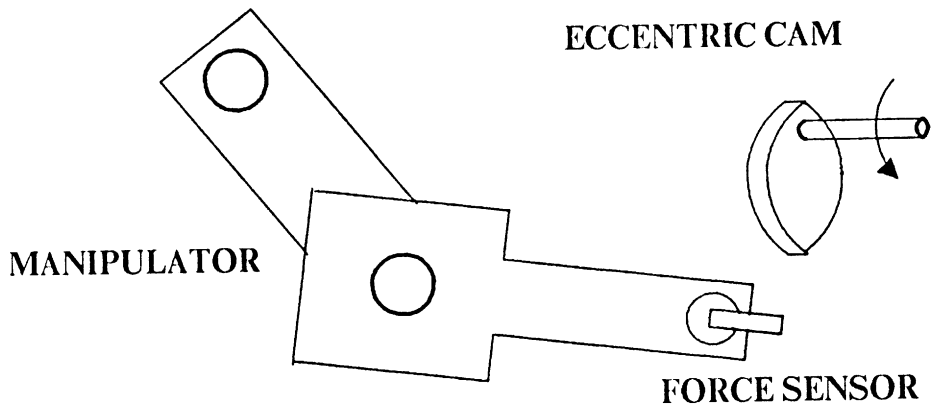


Figure 5.20: Eccentric cam with gear motor for testing force response to positional disturbance

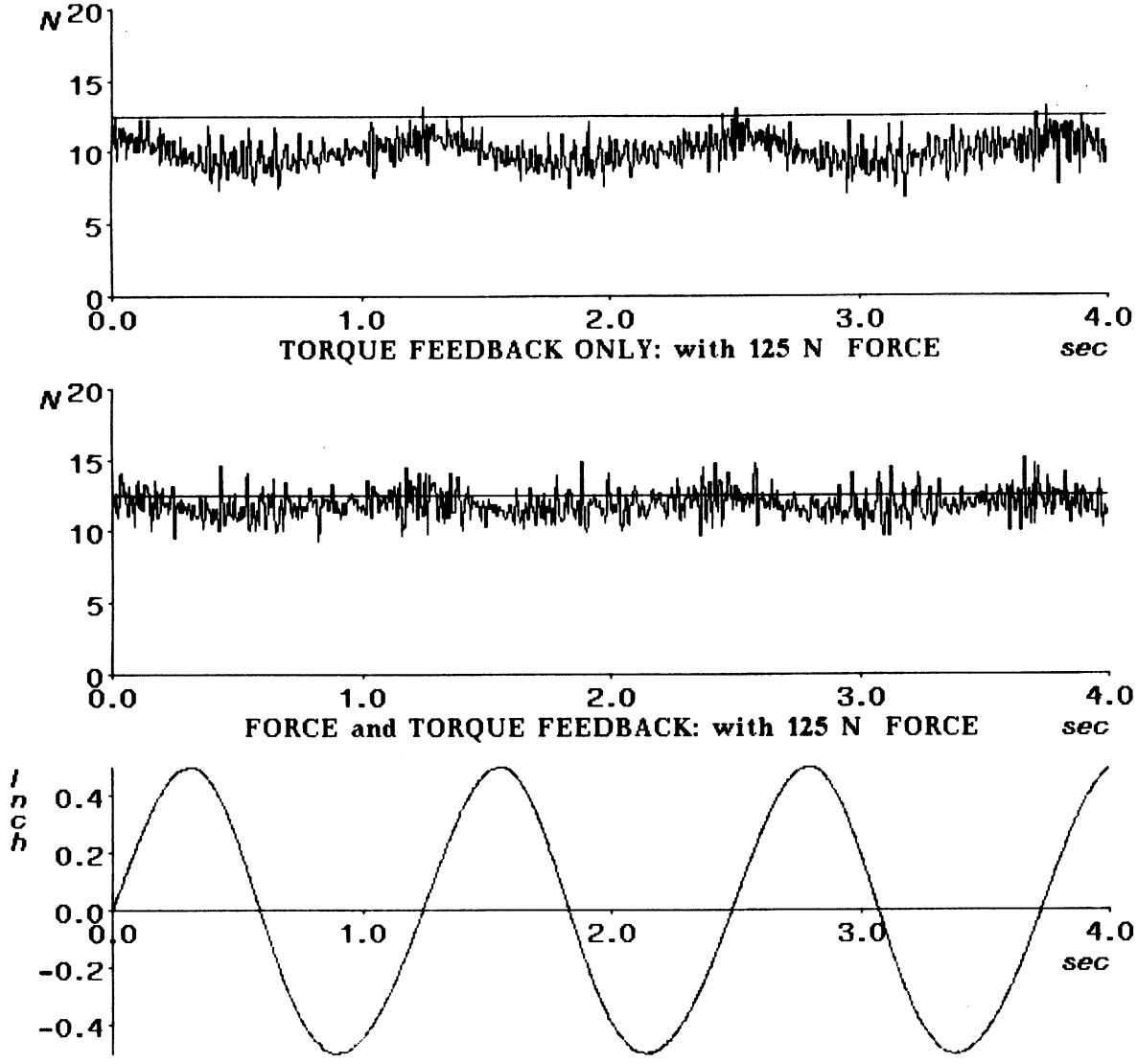


Figure 5.21: Force responses to a sinusoidal positional disturbance

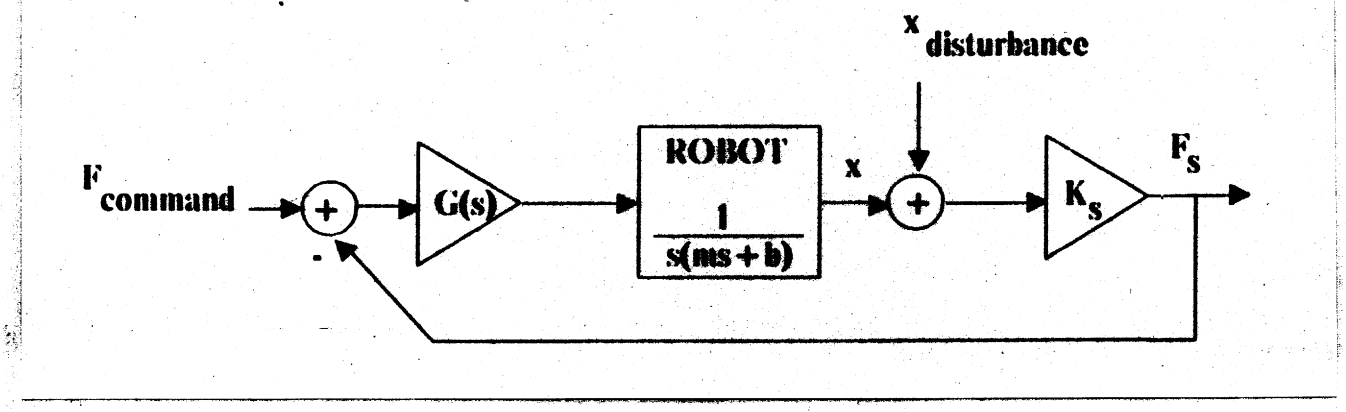


Figure 5.22: Positional Disturbance Rejection

of applications in which the movements are very small, the sinusoid trajectory test may be more relevant.

Chapter 6

Kinematic Stability Issues in Force Control

6.1 Introduction

The dynamic stability problems discussed in Chapter 5 can occur whether the manipulator consists of simply one link or multi-links. For a multi-link manipulator, there is typically some type of coordinate transformation either at the planning stage or in the active feedback computation path. If the transformation is in the feedback path, then it will have some effect on the dynamics of the closed-loop system. In the worst case, the kinematic transformation in the feedback path may even make the system unstable by making the closed-loop poles move into the right half s-plane. This chapter focuses on these issues of kinematically induced instability. The purpose is not to prove any type of stability but instead to prove instabilities by counter examples against stability.

The simplest coordinate system to use in controlling a manipulator is the joint coordinate system. A stable and well behaved response can be obtained using even very simple control algorithms in this way. But the simplest coordinate system for humans to visualize and plan tasks for a manipulator is the cartesian coordinate system. There are several ways to control a non-cartesian robot as a polar manipulator or a revolute manipulator in cartesian coordinates. In pure position control mode, the most widely used method is to compute the inverse kinematics at the tra-

jectory planning stage and then use joint coordinates for the real time control. For compliance or force control, the algorithms must deal with the interaction forces as well as the desired positions and cannot simply execute the preplanned joint trajectory. As stated in the earlier chapters, there are several well known cartesian compliance control algorithms. They are summarized below, and their main structures are shown on Figure 6.1.

Hybrid Control (Raibert and Craig, 1981):[1] The cartesian positions and the velocities are computed from the joint positions and velocities, respectively, by direct or forward kinematics. Neglecting the integral terms,

$$\tau = \mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}(\mathbf{x}_d - \mathbf{x}) + \mathbf{K}_v\mathbf{J}^{-1}\mathbf{S}(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_f\mathbf{J}^T(\mathbf{I} - \mathbf{S})(\mathbf{f}_d - \mathbf{f}) \qquad (6.1)$$

where $\mathbf{S}$ is the diagonal selection matrix whose (i,i) entry is 1 if i^{th}axis is to be position controlled, and 0 if it is to be force controlled.

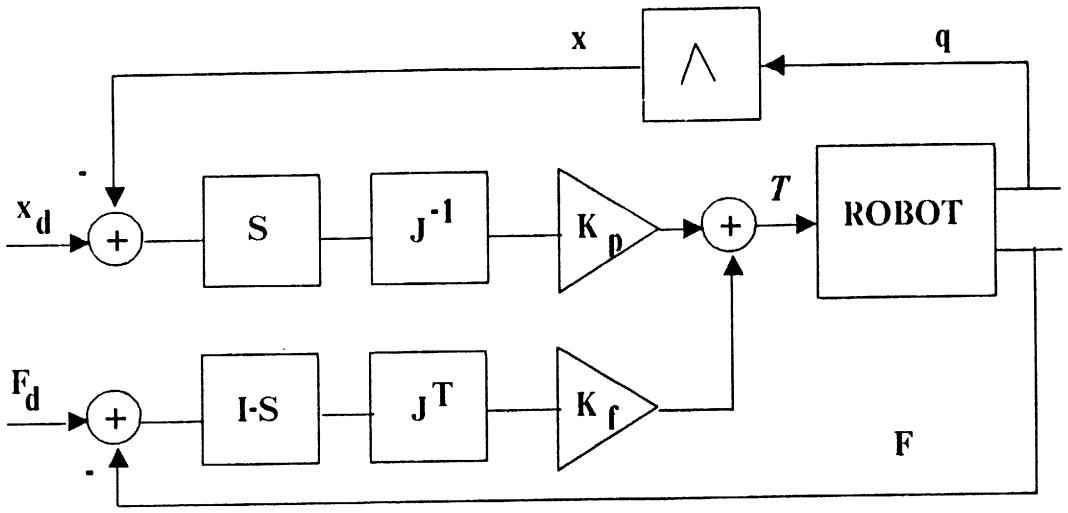
Resolved Acceleration (Luh, Walker, and Paul, 1980b; Khatib, 1983): In the original paper by Luh, Walker, and Paul, the resolved acceleration method was formulated only as a dynamic cartesian trajectory controller. But with a simple modification, this method can be used to control force and position for different cartesian degrees of freedom as the hybrid controller. The modified resolved acceleration controller is given by (6.2).

$$\tau = \mathbf{M}(\mathbf{q})\mathbf{J}^{-1}(\mathbf{q})[\mathbf{S}\mathbf{x}_m^* - \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})] + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{I} - \mathbf{S})\mathbf{f}^* \qquad (6.2)$$

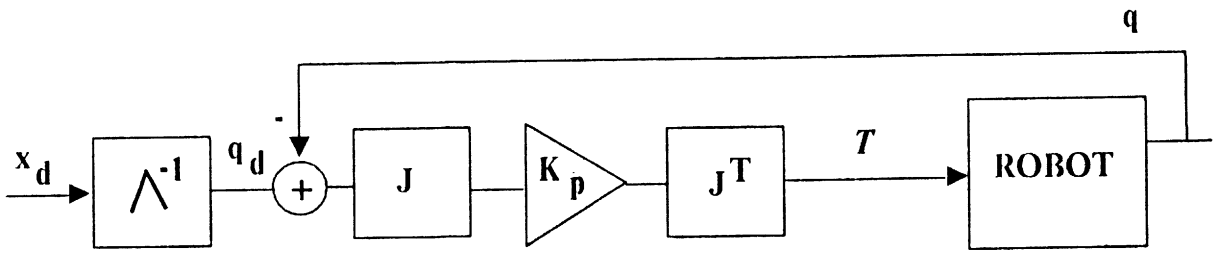
$$\mathbf{x}_m^* = \ddot{\mathbf{x}}_d + \mathbf{K}_v(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}), \quad \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) = \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}$$

$\mathbf{f}^*$ is the command vector for active force control, which is the only modification from the original formulation by Luh, Walker, and Paul. As shown in Appendix 4, Khatib's operational space method is essentially identical to this modified resolved acceleration controller. The impedance control algorithm in (Hogan, 1985b) also fits under the category of resolved acceleration method. The only difference is that the force gain is a function of the desired mass for the impedance controller.

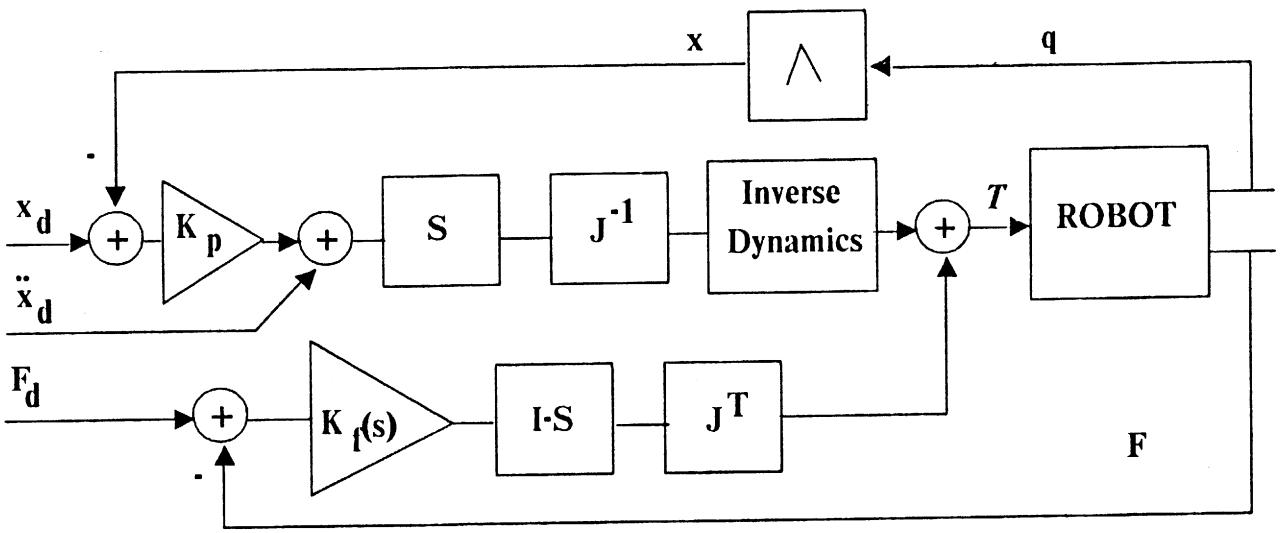
[1] Actually all three of the methods discussed are hybrid controllers in that all of them can be formulated to control position and force for different axes. However, for convenience, when the term the hybrid control is mentioned without any other clarifying description, I am always referring to the method of Raibert and Craig.



(a) Hybrid Controller



(b) Stiffness Controller



(c) Resolved Acceleration Controller

Figure 6.1: Block diagrams for the hybrid, the stiffness, and the resolved acceleration controllers shown without the velocity terms

As in the Raibert and Craig's method, the cartesian positions and the velocities are computed from the joint positions and velocities, respectively, by direct or forward kinematics. However, this controller includes the complete rigid body dynamics of the manipulator, so that if the dynamic modelling were to be exact, the motion of the manipulator would be completely decoupled at the end effector in cartesian coordinates. Then, the response under the pure position control mode would be that of a unit mass along each cartesian degree of freedom. For example, the resulting behavior in the x-axis is

$$\ddot{x}_e + k_{vx}\dot{x}_e + k_{px}x_e = 0, \quad \text{where } x_e = x_d - x. \tag{6.3}$$

Stiffness Control (Salisbury, 1980): The desired cartesian trajectory is transformed to joint coordinates at the trajectory planning stage.

$$\tau = \mathbf{J}^T\mathbf{K}\mathbf{J}(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \tag{6.4}$$

Although the kinematic errors are computed in joint coordinates in Salisbury's original paper (1980), it is also possible to implement stiffness control by computing the errors in cartesian coordinates as in the hybrid control. Then the slightly modified stiffness control algorithm is:

$$\tau = \mathbf{J}^T(\mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) + \mathbf{K}_v(\dot{\mathbf{x}}_d - \dot{\mathbf{x}})) \tag{6.5}$$

Since the stiffness, not the pure force, is to be controlled, the above controller equations are shown without any force feedback term. A force term, however, can be added if the stiffness matrix alone does not provide enough force resolution or if pure force control is desired in some direction.

Potential stability problem arises since the coordinate transformation, either by the Jacobian inverse or the Jacobian transpose, is directly in the feedback loop. Hence, when considering the closed-loop stability of a cartesian force controlled system, the effects of these matrices must be understood. The goal of this chapter is to study stability problems that can be caused by kinematic transformations in the feedback loop. In particular, it will be shown that the stiffness and the resolved acceleration methods do not become unstable for either revolute and polar manipulators. However, due to its inherent control structure, the hybrid control

method proposed by Raibert and Craig can cause instability on a revolute manipulator regardless of the choice of gains. These instabilities occur not only at the points of kinematic singularities, where the Jacobian inverses are not defined, but at a wide range of the manipulator work space, where the Jacobian inverses are well defined. This study includes both intuitive and rigorous analytical results as well as experimental results with the direct drive arm to support the above claims.[2]

A planar two-link revolute manipulator (Figure 6.2) will be mainly used in the analyses.

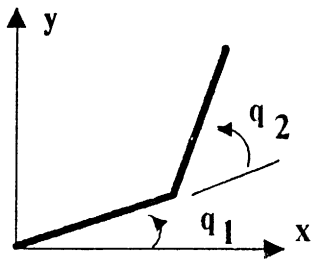


Figure 6.2: Simple two-link planar manipulator

For this planar manipulator, the forward kinematics are

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix} \quad \text{where } s_1 = \sin(\theta_1) \text{ and } s_{12} = \sin(\theta_1 + \theta_2). \tag{6.6}$$

The Jacobian for the fixed cartesian coordinates is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}. \tag{6.7}$$

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\theta}}$$

The inverse Jacobian is

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{6.8}$$

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1}\dot{\mathbf{x}}$$

[2]The motivation for this chapter came from the author's observation of instability when the hybrid controller was tried on the DDARM.

For the lengths l_1 and l_2, I am using the lengths of the DDARM configured as a two-link planar manipulator by locking the second joint and moving only the first and the third joints.

$$l_1 = 0.462m, \quad l_2 = 0.445m$$

6.2 Intuitive Stability Analysis for Revolute Manipulators

Before a more rigorous treatment, it may be helpful to get an intuitive understanding of why and how the instability occurs. The analysis presented in this section is approximate and *not* mathematically rigorous. This analysis will reveal that there indeed is a potential problem with the Raibert and Craig's hybrid approach which uses the Jacobian inverse for coordinate transformations, but not with the stiffness approach which uses the Jacobian transpose. Although the resolved acceleration method also uses the Jacobian inverse, its behavior is different because of its dynamic compensation. Further discussion of the resolved acceleration method is postponed until the next section.

6.2.1 Hybrid Control of Raibert and Craig

The equation for the hybrid controller (6.1) is repeated below.

$$\boldsymbol{\tau} = \mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}\mathbf{x}_e + \mathbf{K}_v\mathbf{J}^{-1}\mathbf{S}\dot{\mathbf{x}}_e + \mathbf{K}_f\mathbf{J}^T(\mathbf{I}-\mathbf{S})(\mathbf{f}_d - \mathbf{f}) \tag{6.9}$$

where

$$\begin{aligned}
\mathbf{x}_e &= \mathbf{x}_d - \mathbf{x} \\
\dot{\mathbf{x}}_e &= \dot{\mathbf{x}}_d - \dot{\mathbf{x}} \\
\mathbf{S} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

The first $\mathbf{S}$ specifies position control in both the x and y directions. The second $\mathbf{S}$ specifies position control only in the y direction and force control in the x direction, and the reverse for the third $\mathbf{S}$.

Let's consider the cases when the manipulator is in free space and has no mass at the end of the force sensor so that for any axis selected to be force controlled,

the desired and the measured forces are 0's. From the results of Chapter 5, this situation is the most stable situation from the dynamic stability point of view. Also, let's look only at the position component, since the velocity component is analogous. Then,

$$\boldsymbol{\tau} = \mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}\mathbf{x}_e \approx \mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}\mathbf{J}\boldsymbol{\theta}_e \tag{6.10}$$

$$\text{where } \boldsymbol{\theta}_e = \boldsymbol{\theta}_d - \boldsymbol{\theta}$$

Also, let $\mathbf{K}_p = \mathbf{I} > 0$ for simplicity.

Case 1 Let $\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, i.e. position control in both x and y directions. Then,

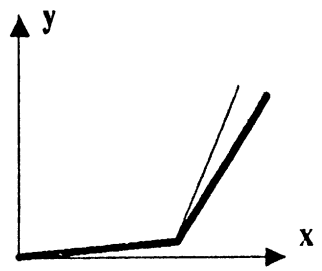
$$\boldsymbol{\tau} = \mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}\mathbf{J}\boldsymbol{\theta}_e = \boldsymbol{\theta}_e = \boldsymbol{\theta}_d - \boldsymbol{\theta}$$

In this case, the hybrid controller becomes a simple independent joint controller and the system remains stable assuming that the control designer has designed a stable joint coordinate control system.

Case 2 Let $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, i.e. position control in y and force control in x. Since the manipulator is assumed to be in free space, the x axis force is being controlled to 0 force. With this selection matrix, position errors in the x direction should not cause any restoring torques. Expanding the simplified hybrid controller equation,

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{J}^{-1}\mathbf{S}\mathbf{J}\boldsymbol{\theta}_e \qquad (6.11) \\ &= \mathbf{J}^{-1}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{J}\boldsymbol{\theta}_e \\ &= \frac{1}{l_1 l_2 s_2}\begin{bmatrix} l_2 s_{12}(l_1 c_1 + l_2 c_{12}) & l_2 s_{12}(l_2 c_{12}) \\ (-l_1 s_1 - l_2 s_{12})(l_1 c_1 + l_2 c_{12}) & (-l_1 s_1 - l_2 s_{12})(l_2 c_{12}) \end{bmatrix}\begin{bmatrix} \theta_{1e} \\ \theta_{2e} \end{bmatrix}. \end{aligned}$$

Let's consider the configuration (Fig.6.3) with $\theta_{1e} = 0$ and $\theta_{2e} > 0$.



$$\theta_1 \approx 0$$
$$0 < \theta_2 < 90$$
$$\theta_{1e} = 0$$
$$\theta_{2e} > 0$$

Figure 6.3: Unstable configuration for hybrid control

Although θ_{2e} has both x and y components, the hybrid controller is being commanded to correct for the y-axis error only. Then, from (6.12), the restoring torques, τ_1 and τ_2, are

$$\begin{aligned}
\tau_1 &= [\frac{1}{l_1 l_2 s_2}(l_2 s_{12})(l_2 c_{12})]\theta_{2e} \\
&> 0 \\
\tau_2 &= -[\frac{1}{l_1 l_2 s_2}(l_1 s_1 + l_2 s_{12})(l_2 c_{12})]\theta_{2e} \\
&= -\alpha(\theta_1, \theta_2)\theta_{2e} \\
&= \alpha(\theta_1, \theta_2)(\theta_2 - \theta_{2d}) \\
&< 0
\end{aligned}$$

For τ_2, the quantity inside the bracket, $\alpha(\theta_1, \theta_2)$, is positive for the configuration considered above. This is a positive feedback on Joint 2 and $\tau_2 < 0$. Then, the response for this hybrid control would be as shown in Figure 6.4. Since τ_1 is positive and τ_2 is negative, the manipulator opens up and moves toward a singularity. This movement will continue and eventually the manipulator will pass through a singularity and become unstable.

Since the x-axis is being force controlled to $f_x = 0$, and there is no restoring force to the position error in the x-axis, this response seems to be an admissible response if only looked from the point of view of cartesian coordinates. The restoring torques are moving the manipulator to reduce the position error in the y direction without any concern for position error in the x direction. However, in reality, this is an unstable behavior since the manipulator is being pushed toward a singularity, where the $\mathbf{J}^{-1}$ matrix is undefined.

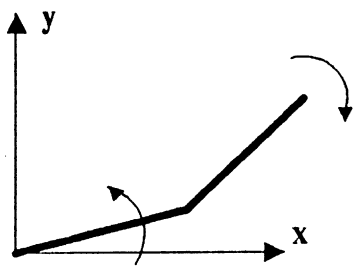


Figure 6.4: Response of the Hybrid Controller with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

6.2.2 Stiffness Control of Salisbury

The potential instability problem of the hybrid controller does not occur with the stiffness control method of Salisbury. As before, let's look only at the position or the stiffness component of the controller and assume that the manipulator is in free space. Then the equation for the stiffness controller is,

$$\tau = \mathbf{J}^T\mathbf{K}\mathbf{J}(\boldsymbol{\theta}_d - \boldsymbol{\theta}) = \mathbf{J}^T\mathbf{K}\mathbf{J}\boldsymbol{\theta}_e. \tag{6.12}$$

Assuming that the stiffness matrix $\mathbf{K}$ is diagonal, the above equation is expanded as

$$\begin{aligned} \mathbf{J}^T\mathbf{K}\mathbf{J}\boldsymbol{\theta}_e &= \begin{bmatrix} J_{11} & J_{21} \\ J_{12} & J_{22} \end{bmatrix} \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \theta_{1e} \\ \theta_{2e} \end{bmatrix} \\ &= \begin{bmatrix} k_x J_{11}^2 + k_y J_{21}^2 & k_x J_{11}J_{12} + k_y J_{21}J_{22} \\ k_x J_{11}J_{12} + k_y J_{21}J_{22} & k_x J_{12}^2 + k_y J_{22}^2 \end{bmatrix} \begin{bmatrix} \theta_{1e} \\ \theta_{2e} \end{bmatrix}. \end{aligned} \tag{6.13}$$

The diagonal entries of the $\mathbf{J}^T\mathbf{K}\mathbf{J}$ matrix is always positive for positive stiffness values. Let's consider the same configuration as before, where only the y-axis is being position controlled,

$$\mathbf{K} = \begin{bmatrix} 0 & 0 \\ 0 & k_y \end{bmatrix}, \quad \theta_{1e} = 0, \quad \theta_{2e} > 0. \tag{6.14}$$

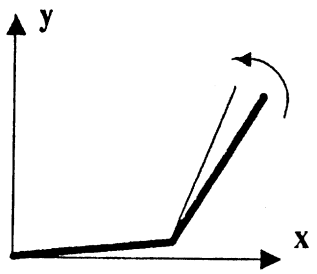


Figure 6.5: Response of the Stiffness Controller with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

From (6.14),

$$\begin{aligned} \tau_1 &= k_y J_{21} J_{22} \\ &= k_y(l_1 c_1 + l_2 c_{12}) + l_2 c_{12} \\ &> 0 \\ \tau_2 &= k_y J_{22}^2 \\ &> 0 \end{aligned}$$

Unlike the case of hybrid control, the τ_2 component of the restoring torques is positive, which is indicative of a negative feedback on Joint 2. Then, as shown in Figure 6.5, since both τ_1 and τ_2 are positive, the manipulator corrects for the error in y position correctly and stably without moving toward a singularity. This stable behavior can be generalized by simple matrix operations for manipulators with more than two joints and for other manipulator configurations and parts of the workspace.

6.2.3 Summary of the intuitive analytical results

The analyses presented in this section suggest that the hybrid control method has a potential instability problem related to the use of the Jacobian inverse in the coordinate transformation. The Jacobian transpose in stiffness control does not seem to create a similar problem. However, as stated at the beginning of this section, the above simple analyses are only approximate. I treated the multivariable manipulator system as essentially SISO systems and looked at the responses due to an error in only one component of the state vector. But a typical manipulator is a full

MIMO system with the dynamics of each state coupled to one another. Therefore, one needs to evaluate the closed-loop eigenvalues of the whole manipulator system under different controllers for a more concrete stability analysis.

6.3 Root Loci, Simulations, and Experiments for Revolute Manipulators

In this section, the intuition gained from the previous approximate analyses are verified by eigenvalue computations, simulations, and experiments of a revolute manipulator under different controllers. For computing the eigenvalues, since a revolute manipulator is a nonlinear system, I can only study local stability by linearizing the dynamics about some equilibrium points. The simulations, however, are done using the full nonlinear dynamics. The experimental results are from the implementations on the DDARM configured as a 2-link planar manipulator with the second joint locked at $\theta_2 = 180°$.

For a planar manipulator, the dynamics are given by,

$$\tau = \mathrm{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathrm{b}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{6.15}$$

Assuming small velocities and linearizing the above equation about some $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}} = 0$, the resulting dynamic equation is

$$\delta\tau = \mathrm{M}(\boldsymbol{\theta})\delta\ddot{\boldsymbol{\theta}}. \tag{6.16}$$

In state space form,

$$\delta\dot{\mathrm{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \delta\mathrm{x} + \begin{bmatrix} \mathbf{0} \\ \mathrm{M}(\boldsymbol{\theta})^{-1} \end{bmatrix} \delta\tau, \quad \text{where } \delta\mathrm{x} = \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\dot{\boldsymbol{\theta}} \end{bmatrix}. \tag{6.17}$$

The inertia matrix $\mathrm{M}(\boldsymbol{\theta})$ of a simple 2-link revolute manipulator (Fig. 6.2) is given in (Brady, et al.,1982).

$$\mathrm{M}(\boldsymbol{\theta}) = \begin{bmatrix} I_1 + I_2 + m_2 l_1 l_2 \cos\theta_2 + \frac{1}{4}(m_1 l_1^2 + m_2 l_2^2) + m_2 l_1^2, & I_2 + \frac{1}{4} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos\theta_2 \\ I_2 + \frac{1}{4} m_2 l_2^2 + \frac{1}{2} m_2 l_1 l_2 \cos\theta_2, & I_2 + \frac{1}{4} m_2 l_2^2 \end{bmatrix}$$

The inertial parameters used in the analyses are the estimated inertial parameters of the DDARM as presented in Chapter 3 after converting the 3-link parameters to

those of the simple 2-link configuration.

$$
\begin{aligned}
I_1 &= 8.095\ kg \cdot m^2 \\
I_2 &= 0.253\ kg \cdot m^2 \\
m_1 &= 120.1\ kg \\
m_2 &= 2.104\ kg \\
l_1 &= 0.462\ m \\
l_2 &= 0.4445\ m
\end{aligned}
$$

6.3.1 Hybrid Control

Let's consider the stability of the simple manipulator under the hybrid control for the same configurations as in Section 6.2.1. At first, let's consider the cases with the robot in free space as before, which is the most dynamically stable configuration. It will be shown later that contacts do not improve the kinematic instabilities.

In state space form, the hybrid controller can be written as

$$
\delta\boldsymbol{\tau} = \left[-\mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}\mathbf{J},\ -\mathbf{K}_v\mathbf{J}^{-1}\mathbf{S}\mathbf{J} \right] \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\dot{\boldsymbol{\theta}} \end{bmatrix}. \tag{6.18}
$$

Then, the closed-loop system is described as

$$
\begin{aligned}
\delta\dot{\mathbf{x}} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}(\boldsymbol{\theta})^{-1}\mathbf{K}_p\mathbf{J}^{-1}\mathbf{S}\mathbf{J} & -\mathbf{M}(\boldsymbol{\theta})^{-1}\mathbf{K}_v\mathbf{J}^{-1}\mathbf{S}\mathbf{J} \end{bmatrix} \delta\mathbf{x} \\
&= \boldsymbol{A}\delta\mathbf{x}
\end{aligned} \tag{6.19}
$$

To guarantee local stability at the equilibrium points, the eigenvalues of the $\boldsymbol{A}$ matrix must have negative real parts.

Case 1 $\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$: Both the x and y axes are position controlled. The intuitive analysis showed that the hybrid control is stable in this case. The root locus diagram (Fig. 6.6) shows the closed-loop eigenvalues of the hybrid controlled manipulator as θ_2 varies from 10° to 90° and $\theta_1 = 0°$. As expected, the eigenvalues are on the left hand side of the s-plane (LHP) and the manipulator *is* stable.

The simulation (Fig. 6.7) of this case also shows a stable behavior. The manipulator is initially at $(\theta_1, \theta_2) = (0°, 70°)$ and the desired point is given in cartesian coordinates as $(x, y) = (0.462\,m, 0.4445\,m)$. The gains are:

$$
k_{p1} = 2500,\ \ k_{v1} = 300,\ \ k_{p2} = 400,\ \ k_{v2} = 30.
$$

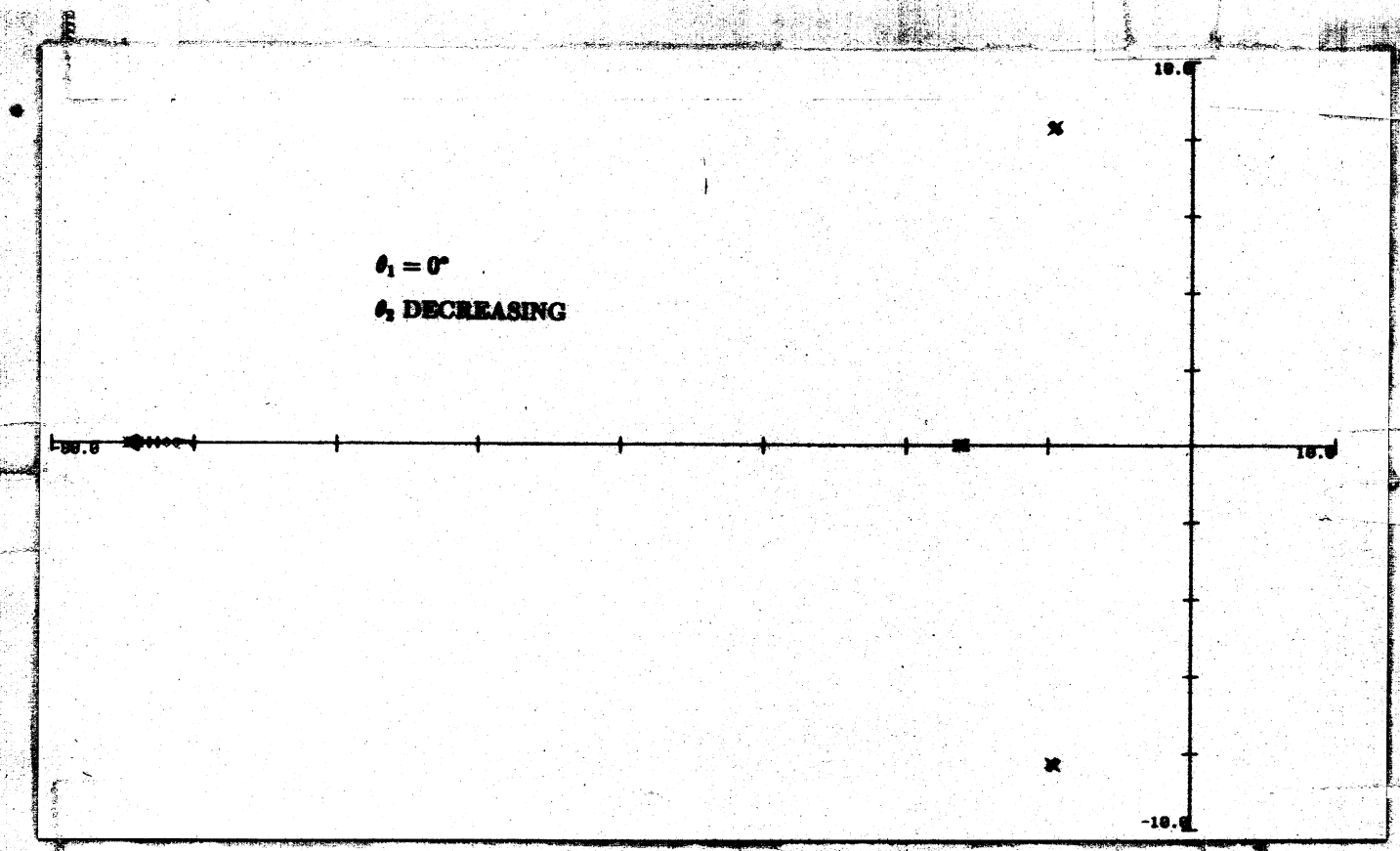


Figure 6.6: Root locus for the hybrid control with $\Phi = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

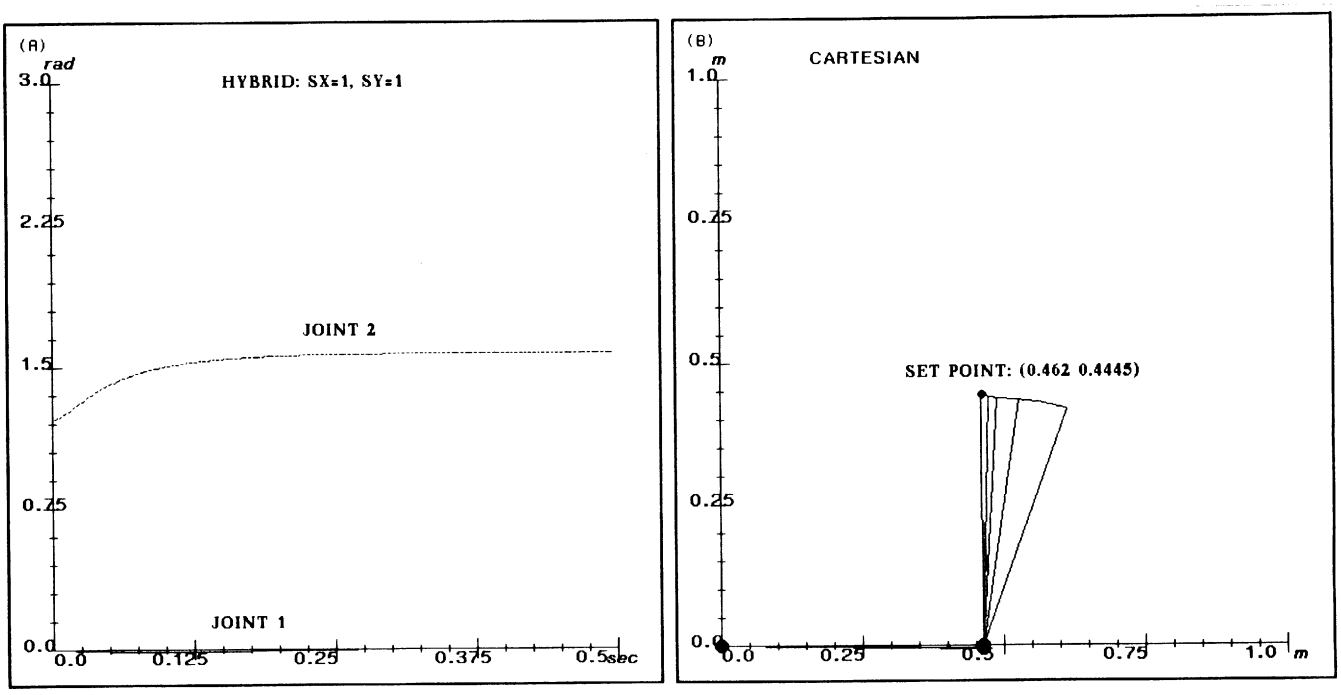


Figure 6.7: Simulation for the hybrid control with $\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

These gains are chosen reasonably and they approximately match the actual gains used in the experiments.

In the experimental results shown in Figure 6.8, the geometric configuration of the arm is 180° rotated from the configuration shown on the simulation results because it was more convenient for me to operate the robot in this configuration. However, the dynamic behavior of the manipulator in either configuration is identical. The results on the direct drive arm also verify that this case is indeed stable for the hybrid control. Figure 6.8 also shows the manipulator under the hybrid controller executing a triangular trajectory, defined by straight lines between the three corner points.

Case 2 $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$: The x-axis is force controlled to 0 force and the y- axis is position controlled. The root locus of this case for $\theta_1 = 0$ and θ_2 varying from 90° to 70° is shown in Figure 6.9. Because one axis is being controlled to 0 force, the behavior of the manipulator along that axis is that of a pure mass or a double integrator. Therefore, two of the poles are at the origin, and only the remaining two poles are shown to be varying. For θ_2 near 90°, the system is stable and the

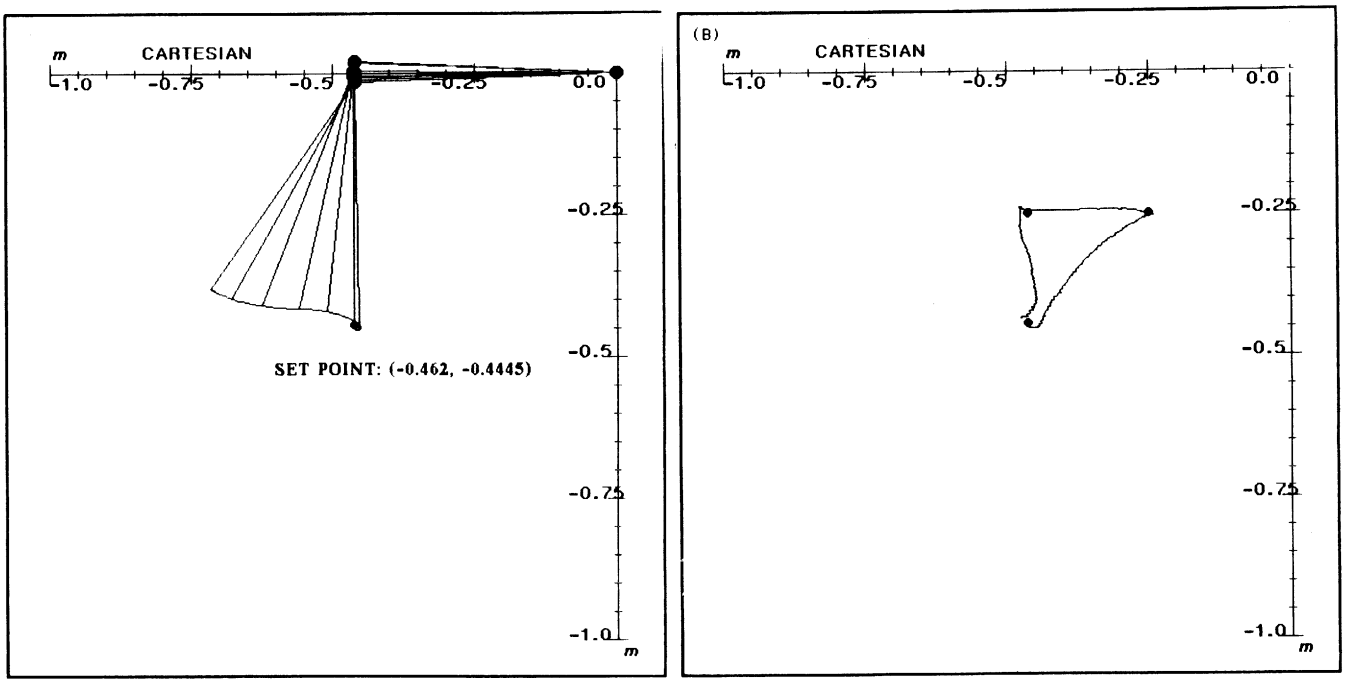


Figure 6.8: Experimental results for the hybrid control with $\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

eigenvalues are negative. However, as θ_2 becomes smaller, the poles move into the right half of the s-plane. The interaction of the inertia matrix $\mathbf{M}(\boldsymbol{\theta})$ with the $\mathbf{J}^{-1}$ matrix is such that the eigenvalues of the $\boldsymbol{A}$ matrix of the equation (6.19) have become positive.

In the above root locus, the manipulator was assumed to be in free space. Figure 6.10 shows the root locus for the manipulator in contact with a stiff wall ($k_s = 10^5\, N/m$). Since the instability is generated by the interaction of the inertia matrix and the Jacobian inverse, a manipulator in contact with its environment does not solve this instability problem.

The simulations of this case with and without contact (Figs. 6.11 and 6.12) show the expected unstable behaviors. The experimental result is shown in Figure 6.13 for the manipulator in free space. The manipulator is initially at $(x, y) = (-0.462\, m, -0.4445\, m)$, which is a stable configuration according to the root locus diagram. Then, with a very light force, I pulled the tip of the manipulator along the $-x$ direction, i.e. toward the more unstable configuration. Then, the manipulator becomes unstable approximately at the point A of the figure, which corresponds to $\theta_2 = 75°$ in the coordinate system used in the root locus analyses and the simulations. This point agrees well with the root locus diagram of the ideal manipulator,

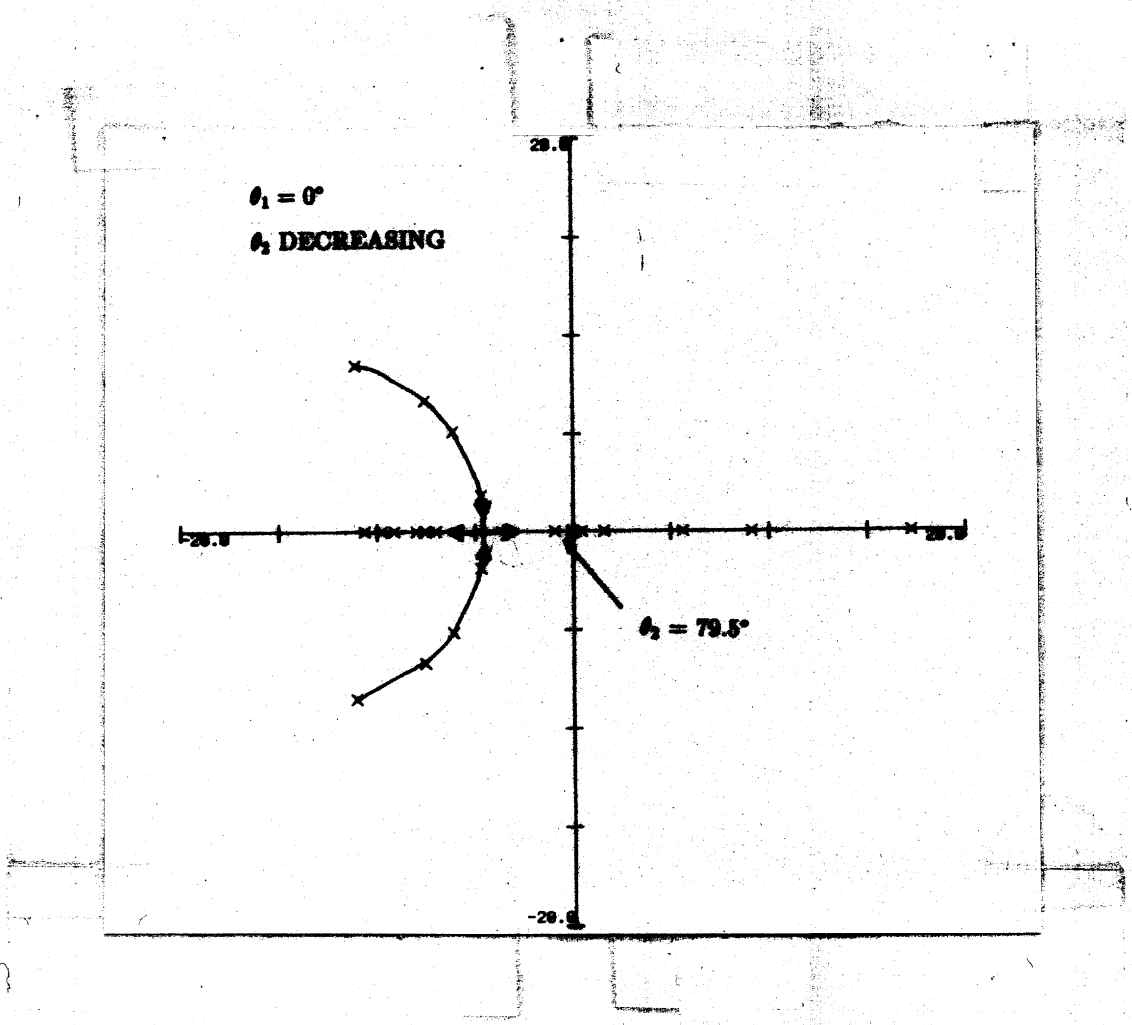


Figure 6.9: Root locus for the hybrid control with $S = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

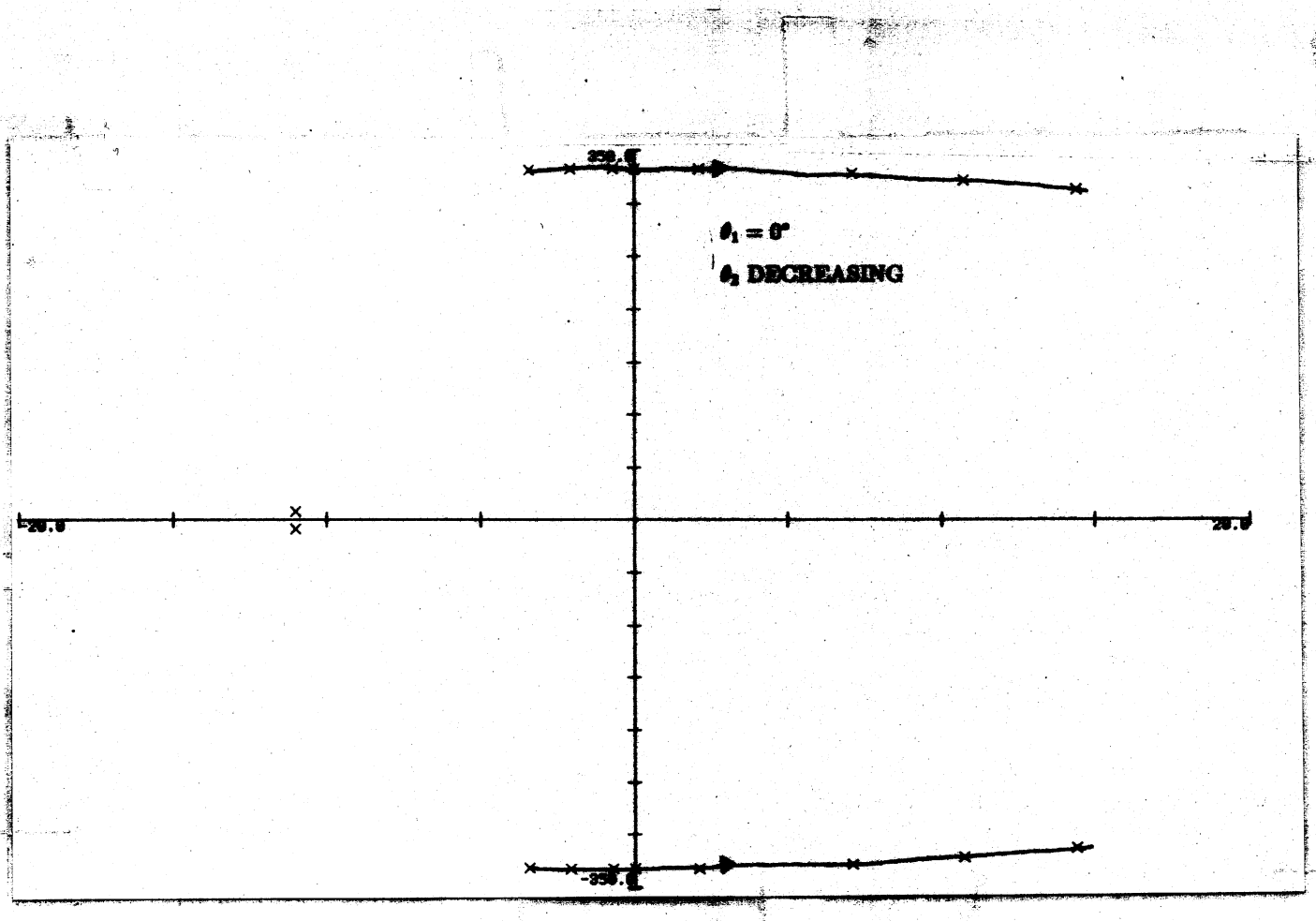


Figure 6.10: Root locus for the hybrid control with $S = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ and in contact with a stiff wall ($k_e = 10^6\ N/m$)

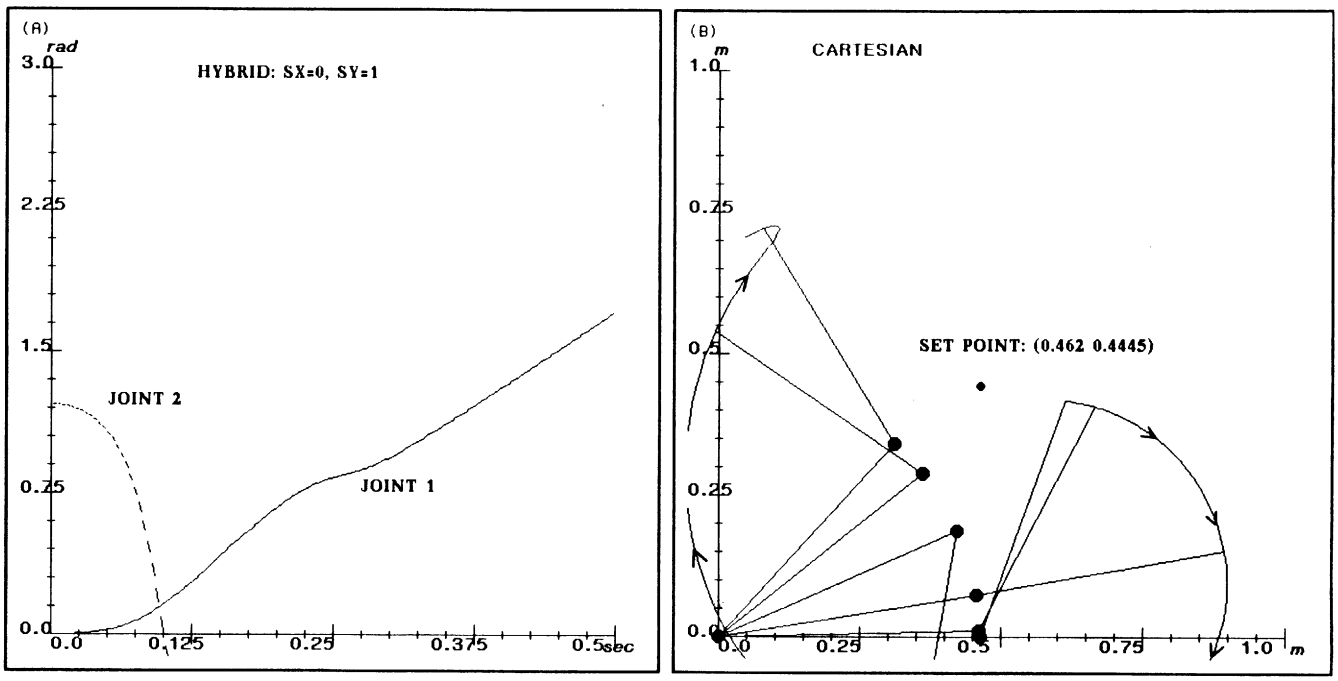


Figure 6.11: Simulation for the hybrid control with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

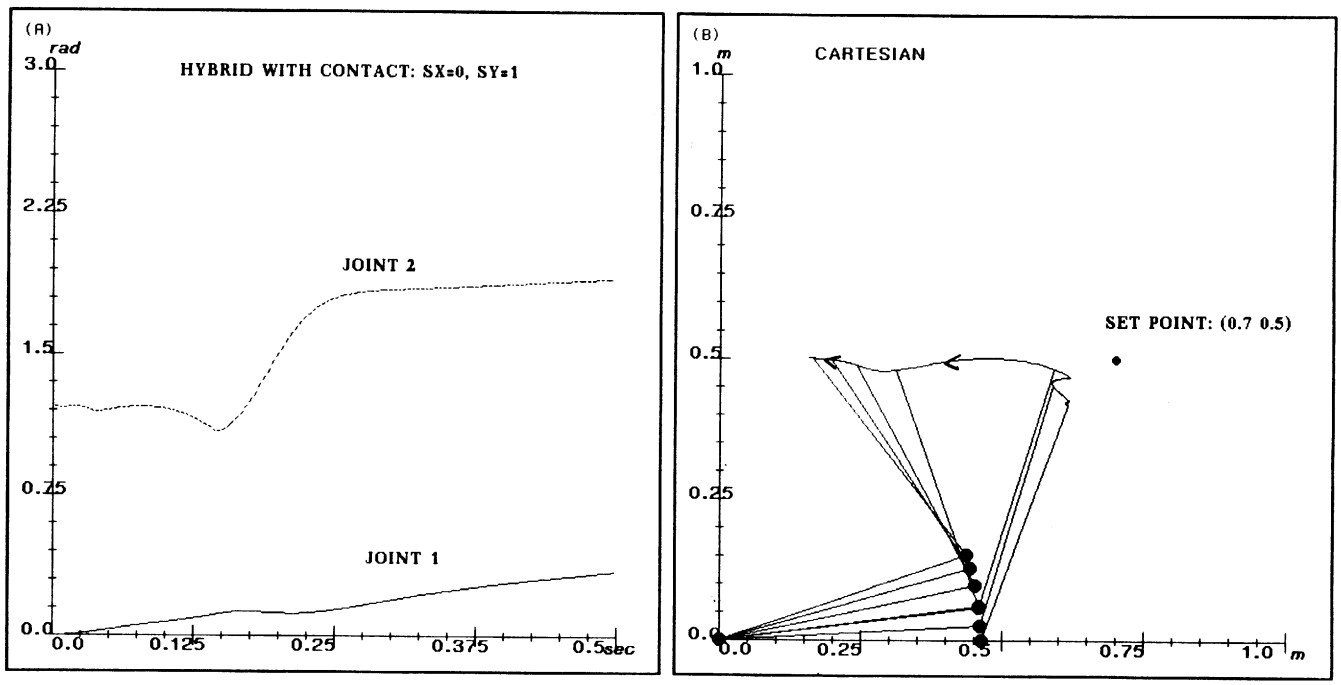


Figure 6.12: Simulation for the hybrid control with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ and in contact with a stiff wall ($k_s = 10^5\ N/m$)

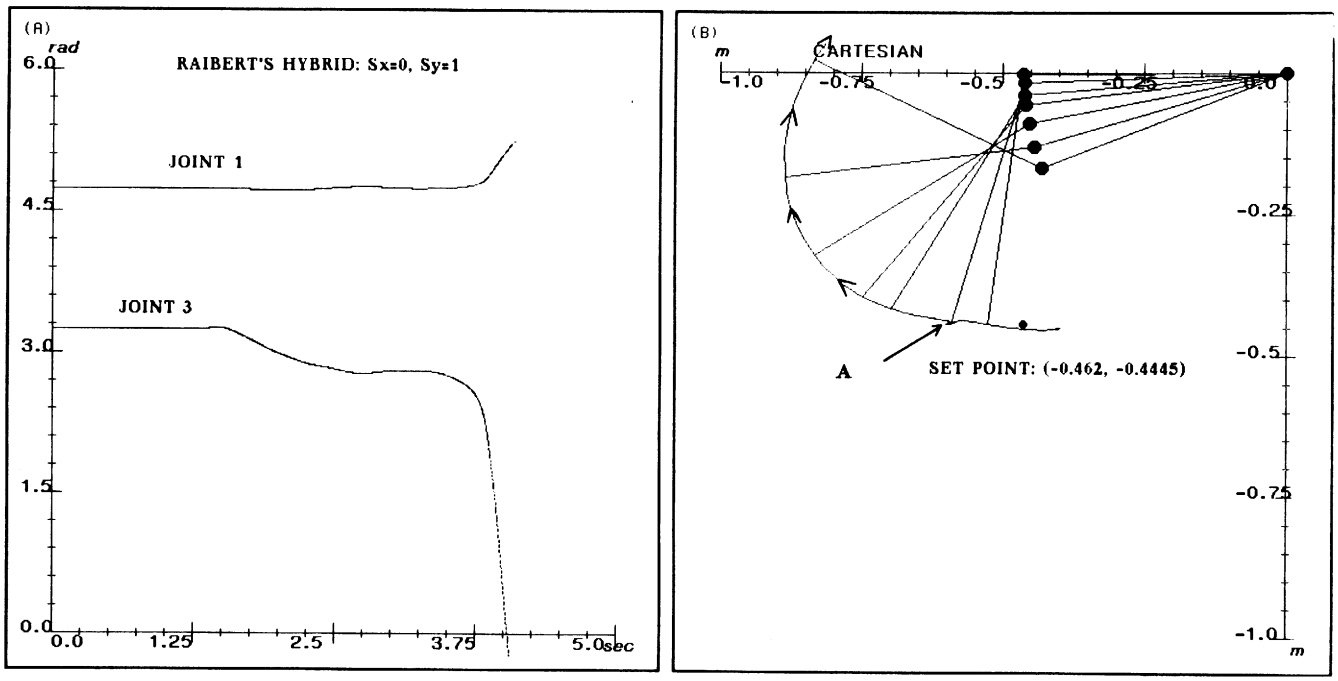


Figure 6.13: Experimental results for the hybrid control with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

which shows that one of the poles becomes positive at $\theta_2 < 79.5°$. Except for the 180° rotation in the manipulator configuration from that used in the simulation, the unstable behavior in the experiment corresponds very closely with the behavior in the simulation. For the case of contact with environment, since instability is predicted, I did not try the same experiment with the DDARM for fear of injuring the robot and myself.

6.3.2 Resolved Acceleration Force Control

In state space form, the linearized resolved acceleration controller is written as

$$\delta\boldsymbol{\tau} = \begin{bmatrix} -\hat{\mathbf{M}}(\boldsymbol{\theta})\mathbf{J}^{-1}\mathbf{S}\mathbf{K}_p\mathbf{J}, & -\hat{\mathbf{M}}(\boldsymbol{\theta})\mathbf{J}^{-1}\mathbf{S}\mathbf{K}_v\mathbf{J} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\dot{\boldsymbol{\theta}} \end{bmatrix}. \tag{6.20}$$

$\hat{\mathbf{M}}(\boldsymbol{\theta})$ is the model of the actual inertia matrix $\mathbf{M}(\boldsymbol{\theta})$. Then, the closed-loop system is described as

$$\delta\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}(\boldsymbol{\theta})^{-1}\hat{\mathbf{M}}(\boldsymbol{\theta})\mathbf{J}^{-1}\mathbf{S}\mathbf{K}_p\mathbf{J} & -\mathbf{M}(\boldsymbol{\theta})^{-1}\hat{\mathbf{M}}(\boldsymbol{\theta})\mathbf{J}^{-1}\mathbf{S}\mathbf{K}_v\mathbf{J} \end{bmatrix} \delta\mathbf{x}. \tag{6.21}$$

If we assume perfect modelling so that $\hat{\mathbf{M}}(\boldsymbol{\theta}) = \mathbf{M}(\boldsymbol{\theta})$, the closed-loop system becomes

$$\begin{aligned} \delta\dot{\mathbf{x}} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{J}^{-1}\mathbf{S}\mathbf{K}_p\mathbf{J} & -\mathbf{J}^{-1}\mathbf{S}\mathbf{K}_v\mathbf{J} \end{bmatrix} \delta\mathbf{x} \\ &= \boldsymbol{A}\delta\mathbf{x}. \end{aligned} \tag{6.22}$$

This $\boldsymbol{A}$ matrix can be decoupled as

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} \mathbf{J}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{S}\mathbf{K}_p & -\mathbf{S}\mathbf{K}_v \end{bmatrix} \begin{bmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} \\ &= \mathbf{Q}^{-1}\mathbf{B}\mathbf{Q}. \end{aligned} \tag{6.23}$$

The above equation shows that the $\boldsymbol{A}$ matrix for the resolved acceleration is a similarity transform of the stable matrix $\mathbf{B}$ consisting only of $\mathbf{S}$, $\mathbf{K}_p$, and $\mathbf{K}_v$. Since the eigenvalues are preserved under any similarity transform, the eigenvalues of the $\boldsymbol{A}$ matrix is simply given by the eigenvalues of $\mathbf{B}$, which are designed to be stable by the choices of $\mathbf{K}_p$ and $\mathbf{K}_v$ matrices. Therefore, under perfect modelling, the resolved acceleration control results in a stable system.

This is in contrast to the earlier intuitive analysis which seemed to suggest that any method using the Jacobian inverses can lead to instability. However, as I mentioned earlier, that analysis was only approximate, and it did not predict a correct behavior for this case. Although the Jacobian inverse is still a problem at kinematic singularities, at any other part of manipulator workspace, it does not interact in any harmful way with the inertia matrix since the inertia matrix is cancelled by the resolved acceleration controller.

The eigenvalues are plotted in Figure 6.14 for the resolved acceleration controller with the following gains:

$$\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{K}_p = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix} \quad \mathbf{K}_v = \begin{bmatrix} 40 & 0 \\ 0 & 40 \end{bmatrix}$$

For this controller, the eigenvalues are negative and do not vary with the manipulator configuration since the matrix $\mathbf{B}$ in (6.23) is independent of the manipulator dynamics. This stability property is quite robust to the modelling errors. Even when I added 50% errors in the inertial parameters, the eigenvalues remained in the left half plane. However, if the modelling errors are very large, then the eigenvalues will eventually become positive. For example, if we model the inertia matrix

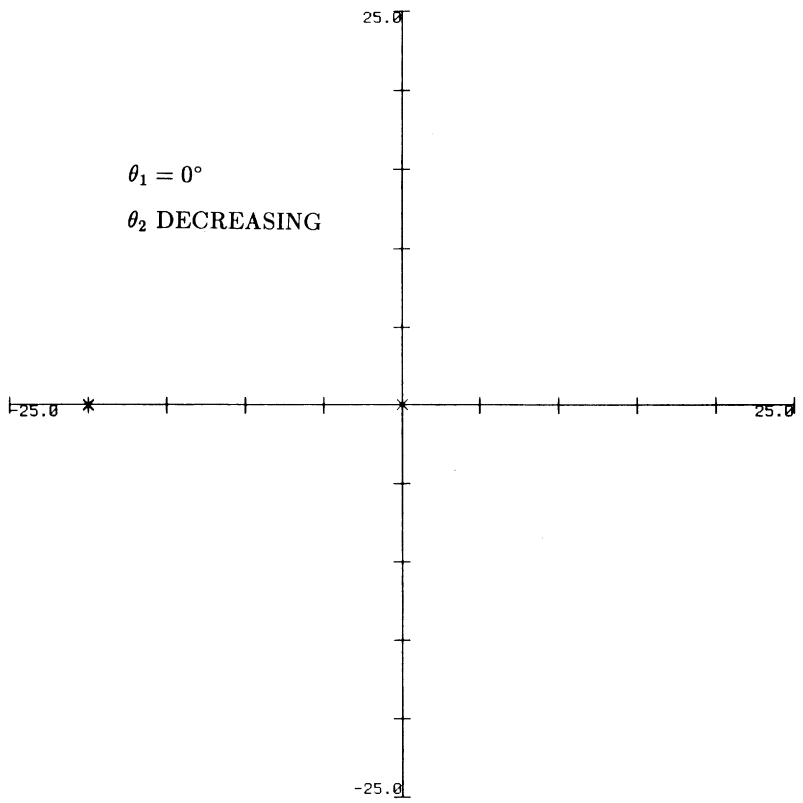


Figure 6.14: Root locus for the resolved acceleration control with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

as the identity matrix, this reduces to the structure of the hybrid controller of Raibert and Craig, and the system will become unstable. Therefore, accurate modelling of the inertial terms of the manipulator is important in force control also as well as in trajectory control. Since I only evaluated the stability for the cases when the velocity and the gravity terms are 0's, I cannot conclusively state how these terms will affect kinematic stability.

Since the system is proven above to be stable, simulation results are not included and only the experimental results are presented. Figure 6.15 shows the responses when both axes are position controlled. The response is stable and for the same triangular trajectory as before, this controller follows the desired path much more accurately than the hybrid controller. This is expected from the trajectory control results of Chapter 4 since the controller structure includes the dynamics of the manipulator. The response when the y-axis is position controlled and the x-axis is

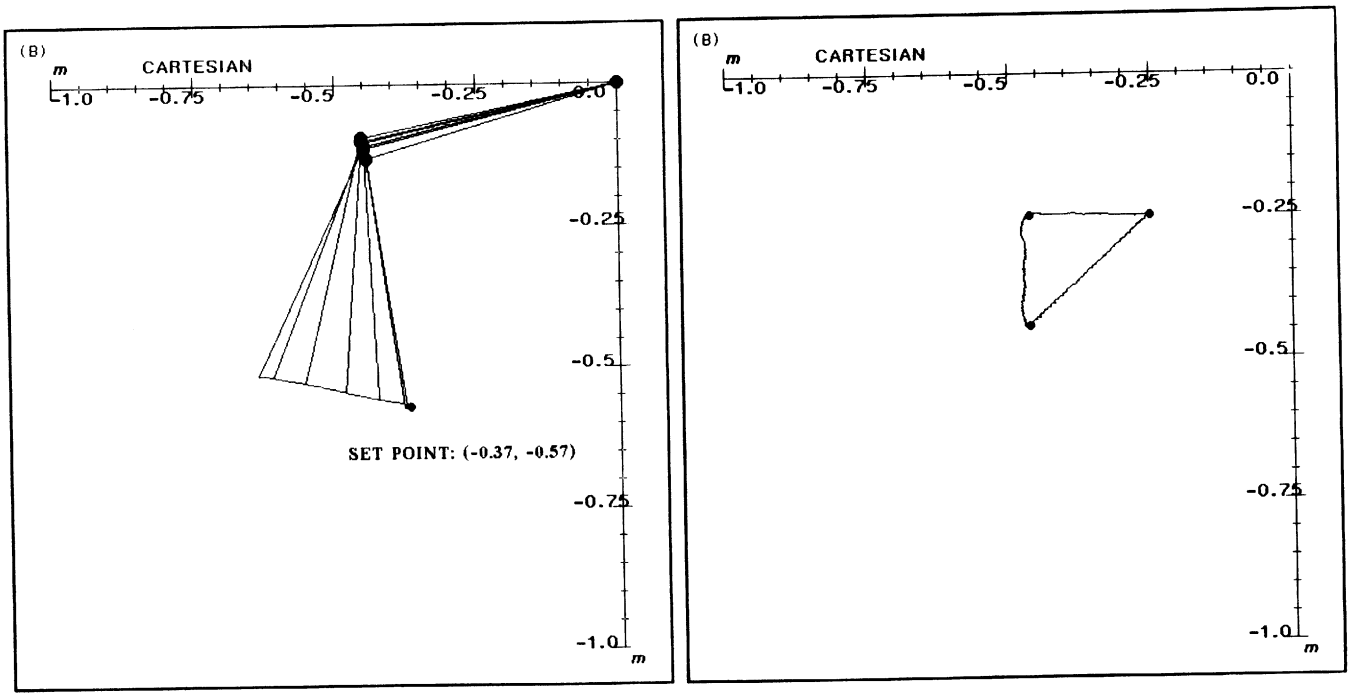


Figure 6.15: Experimental results for the resolved acceleration control with $\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

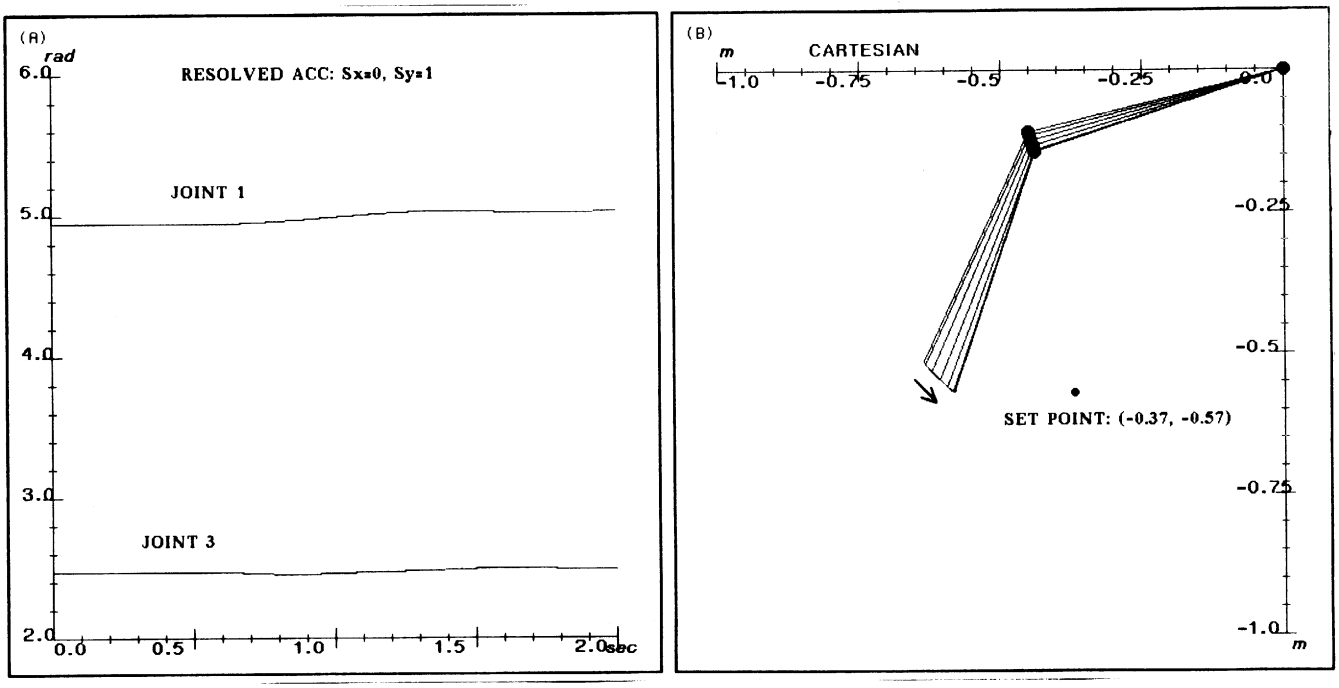


Figure 6.16: Experimental results for the resolved acceleration control with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

force controlled to 0 force is shown in Figure 6.16. The manipulator is stable and as desired only corrects for the error in the y direction, ignoring the position error in the x direction.

6.3.3 Stiffness Control

The stiffness controller is described in state space form as,

$$\delta\tau = \begin{bmatrix} -\mathbf{J}^T\mathbf{K}_p\mathbf{J}, & -\mathbf{J}^T\mathbf{K}_v\mathbf{J} \end{bmatrix} \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \end{bmatrix}. \tag{6.24}$$

The closed-loop system is

$$\begin{aligned} \delta\dot{\mathbf{x}} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}(\theta)^{-1}\mathbf{J}^T\mathbf{K}_p\mathbf{J} & -\mathbf{M}(\theta)^{-1}\mathbf{J}^T\mathbf{K}_v\mathbf{J} \end{bmatrix} \delta\mathbf{x} \\ &= \boldsymbol{A}\delta\mathbf{x} \end{aligned} \tag{6.25}$$

Since dynamics are not included in the controller structure, the inertia matrix is not cancelled, and it is not obvious from the form of the $\boldsymbol{A}$ matrix what the stability characteristics should be. Therefore, I will follow the similar steps of analysis as in the study of the hybrid controller.

Case 1 $\mathbf{K}_p = \begin{bmatrix} 2000 & 0 \\ 0 & 2000 \end{bmatrix}$: The root locus (Fig. 6.17) for this case shows that the eigenvalues are always negative and the system is stable. Simulation results are not interesting since the system is stable and are not included. The experimental results for this case are shown in Figure 6.18. Since dynamics are not included, the stiffness controller performs poorly in following the triangular path. Yet, the response is always stable.

Case 2 $\mathbf{K}_p = \begin{bmatrix} 0 & 0 \\ 0 & 2000 \end{bmatrix}$: The root locus (Fig. 6.19) shows that this case is also stable as predicted by the intuitive analysis. The experimental results shown in Figure 6.20 also verifies the stable property of this method.

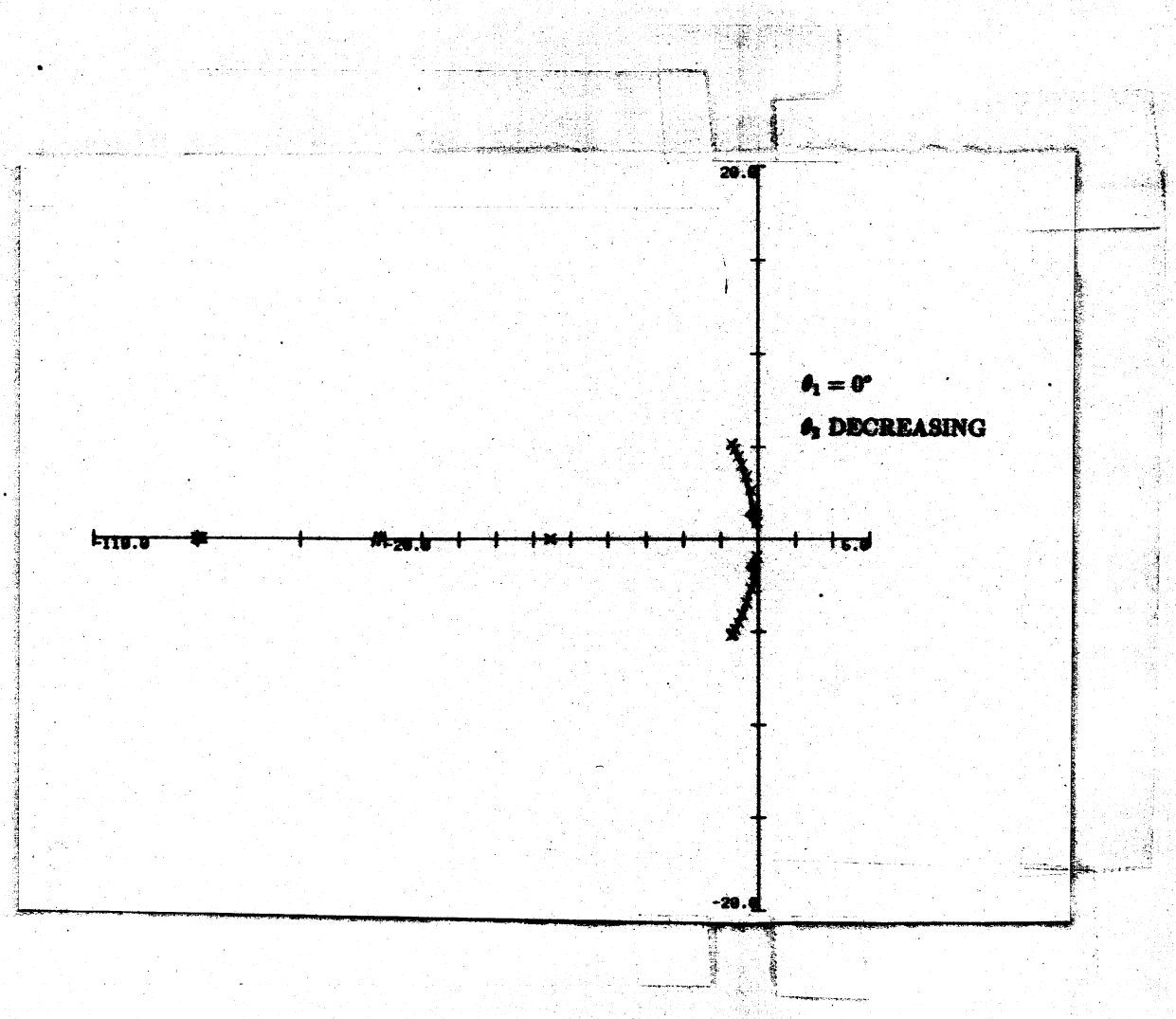


Figure 6.17: Root locus for the stiffness control with $K_p = \begin{bmatrix} 2000 & 0 \\ 0 & 2000 \end{bmatrix}$

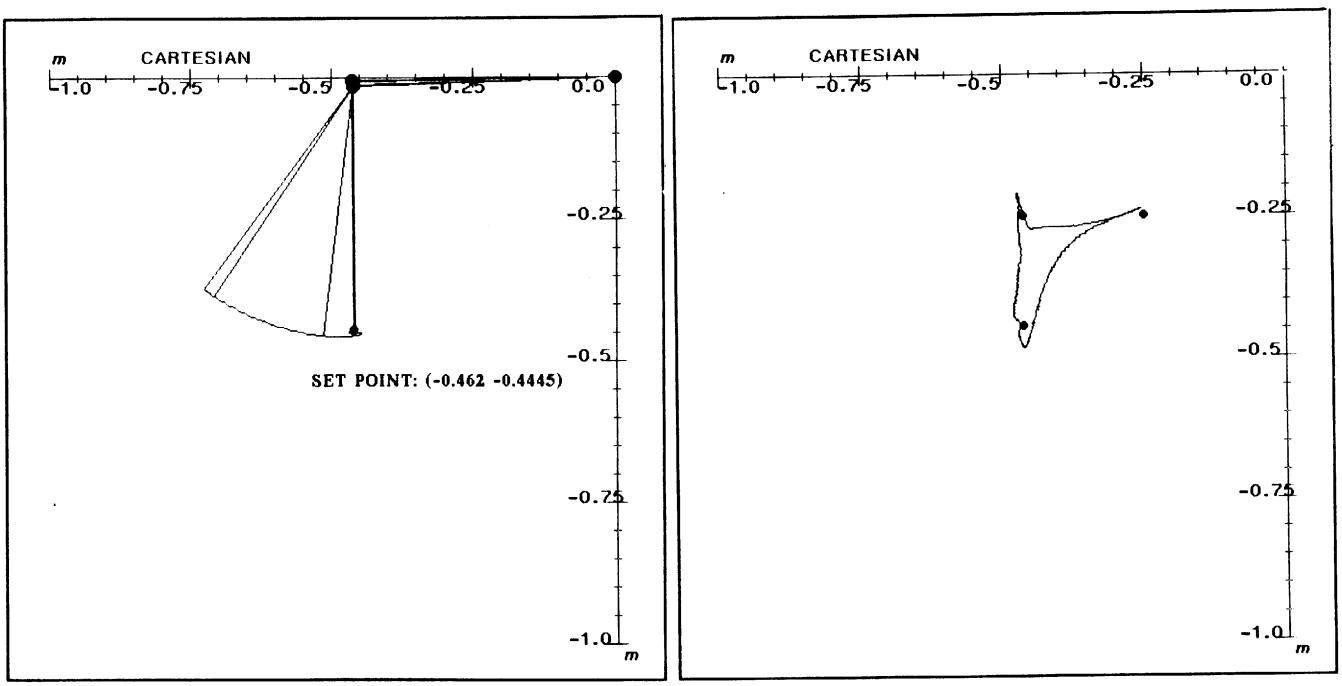


Figure 6.18: Experimental results for the stiffness control with $\mathbf{K}_p = \begin{bmatrix} 2000 & 0 \\ 0 & 2000 \end{bmatrix}$

6.4 Stability for Polar Manipulators

All of the analyses so far have used revolute manipulators, and it was shown that the hybrid control method produces an unstable system for these types of manipulators. This result, however, is not true for polar manipulators. Let's consider a simple planar polar manipulator (Fig. 6.21) as described in (Lozano-Perez, 1983).

The direct kinematic relationships of such a manipulator are:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d_2 \cos\theta_1 \\ d_2 \sin\theta_1 \end{bmatrix}. \tag{6.26}$$

The Jacobian relationship is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -d_2 \sin\theta_1 & \cos\theta_1 \\ d_2 \cos\theta_1 & \sin\theta_1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \end{bmatrix}. \tag{6.27}$$

The inverse Jacobian relationship is:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \end{bmatrix} = -\frac{1}{d_2} \begin{bmatrix} \sin\theta_1 & -\cos\theta_1 \\ -d_2 \cos\theta_1 & -d_2 \sin\theta_1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{6.28}$$

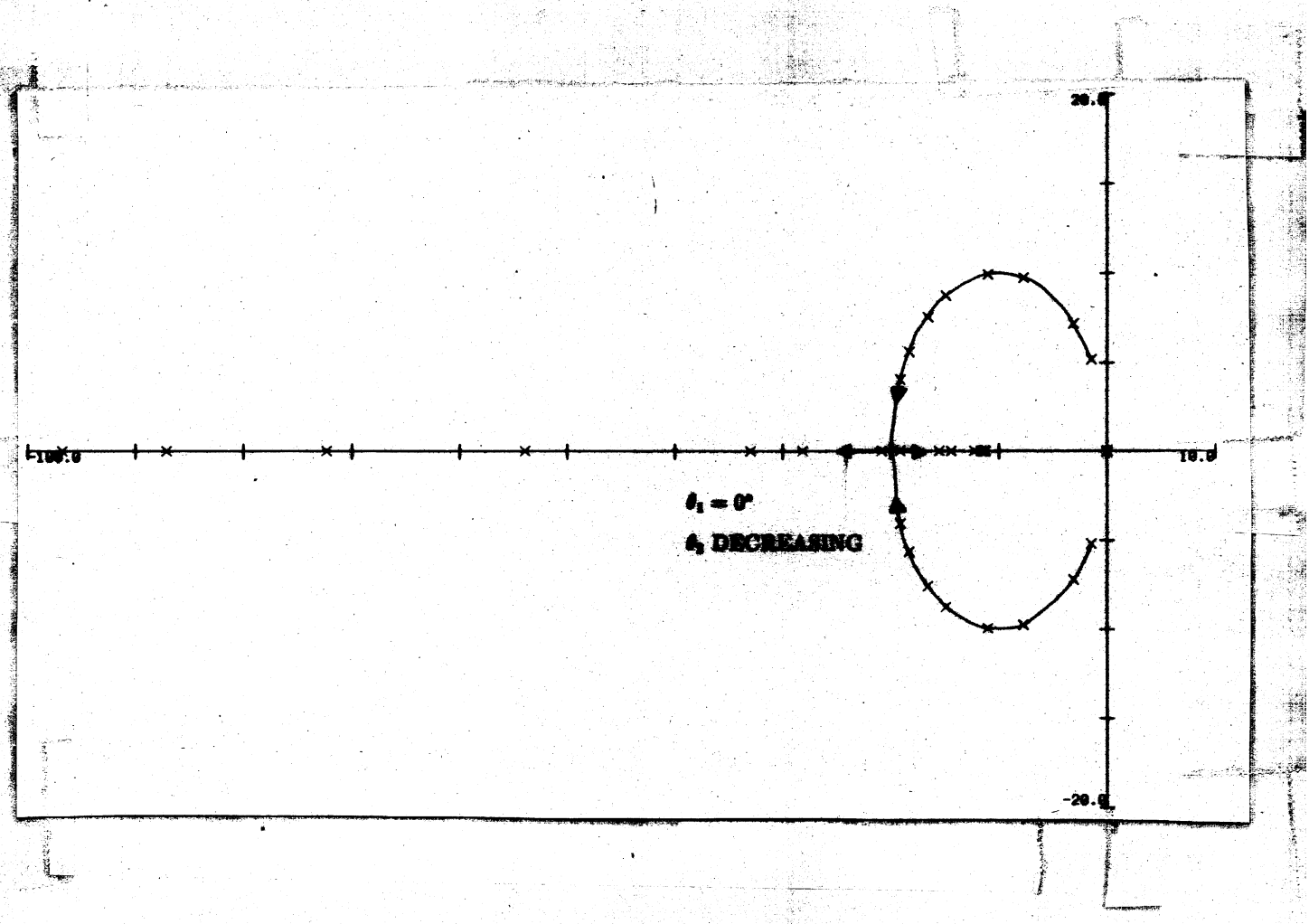


Figure 6.19: Root locus for the stiffness control with $K_p = \begin{bmatrix} 0 & 0 \\ 0 & 2000 \end{bmatrix}$

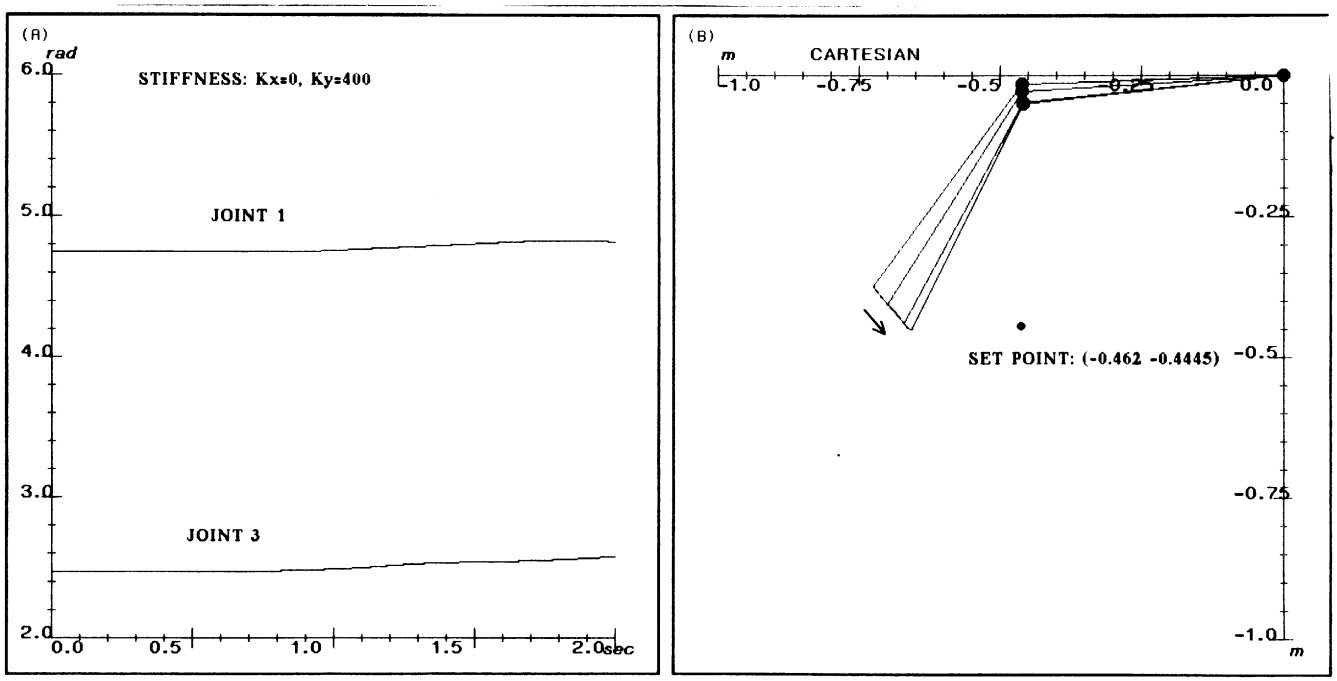


Figure 6.20: Experimental results for the stiffness control with $\mathbf{K}_p = \begin{bmatrix} 0 & 0 \\ 0 & 2000 \end{bmatrix}$

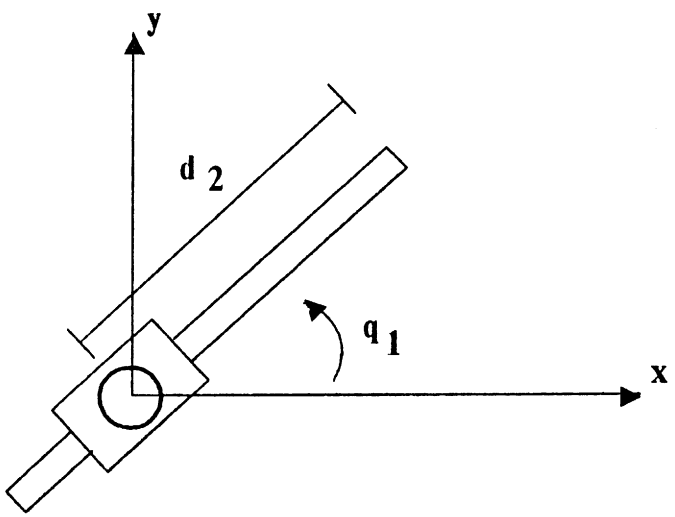


Figure 6.21: Simple planar polar manipulator

The dynamics are:

$$\begin{bmatrix} \tau_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} (d_2 - \frac{1}{2}l_2)^2 m_2 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_2 \\ \ddot{d}_2 \end{bmatrix} + \begin{bmatrix} 2(d_2 - \frac{1}{2}l_2)m_2 \dot{d}_2 \dot{\theta}_1 \\ -(d_2 - \frac{1}{2}l_2)m_2 \dot{\theta}_1^2 \end{bmatrix} \tag{6.29}$$

Intuitive Analysis Following the intuitive analysis for the revolute manipulators, when $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$,

$$\begin{aligned} \tau &= \mathbf{K}_p \mathbf{J}^{-1} \mathbf{S} \mathbf{J} \mathbf{q}_e \\ &= \frac{1}{2} \begin{bmatrix} d_2 \cos^2\theta_1 & \cos\theta_1 \sin\theta_1 \\ d_2^2 \sin\theta_1 \cos\theta_1 & d_2 \sin^2\theta_1 \end{bmatrix} \begin{bmatrix} \theta_{1e} \\ d_{2e} \end{bmatrix}. \end{aligned} \tag{6.30}$$

The diagonal elements of the above gain matrix are always positive, resulting in stable negative feedbacks if, as before, the nonlinear coupled system were to be approximated as two independent SISO systems. Therefore, we would expect the polar manipulators not to exhibit kinematic instabilities under the hybrid control.

Root locus and Simulations The stability analysis of a polar manipulator is studied further by linearizing the dynamics about some equilibrium points as before. In order to be consistent with the results of Raibert and Craig (1981), the kinematic, the dynamic, and the control parameters reported in their study are used for evaluating the hybrid controller by root locus and simulations The root locus with the above S matrix is shown in Figure 6.22 as d_2 varies from $0.05\,m$ to $0.6\,m$. The two non-zero eigenvalues are always negative in this case. The simulation shown in Figure 6.23 also demonstrates a stable behavior.

These results on a polar manipulator verify the stable results reported by Raibert and Craig in their pioneering implementation of hybrid control on the Stanford Arm, which is a polar manipulator. Large friction in their manipulator may have also added stability to their implementations.

6.5 Conclusion on Kinematic Stability

The kinematic instabilities are very dependent on the geometric configuration of and on the types of a manipulator. The hybrid controller of Raibert and Craig produces an unstable system when implemented on a revolute manipulator. The

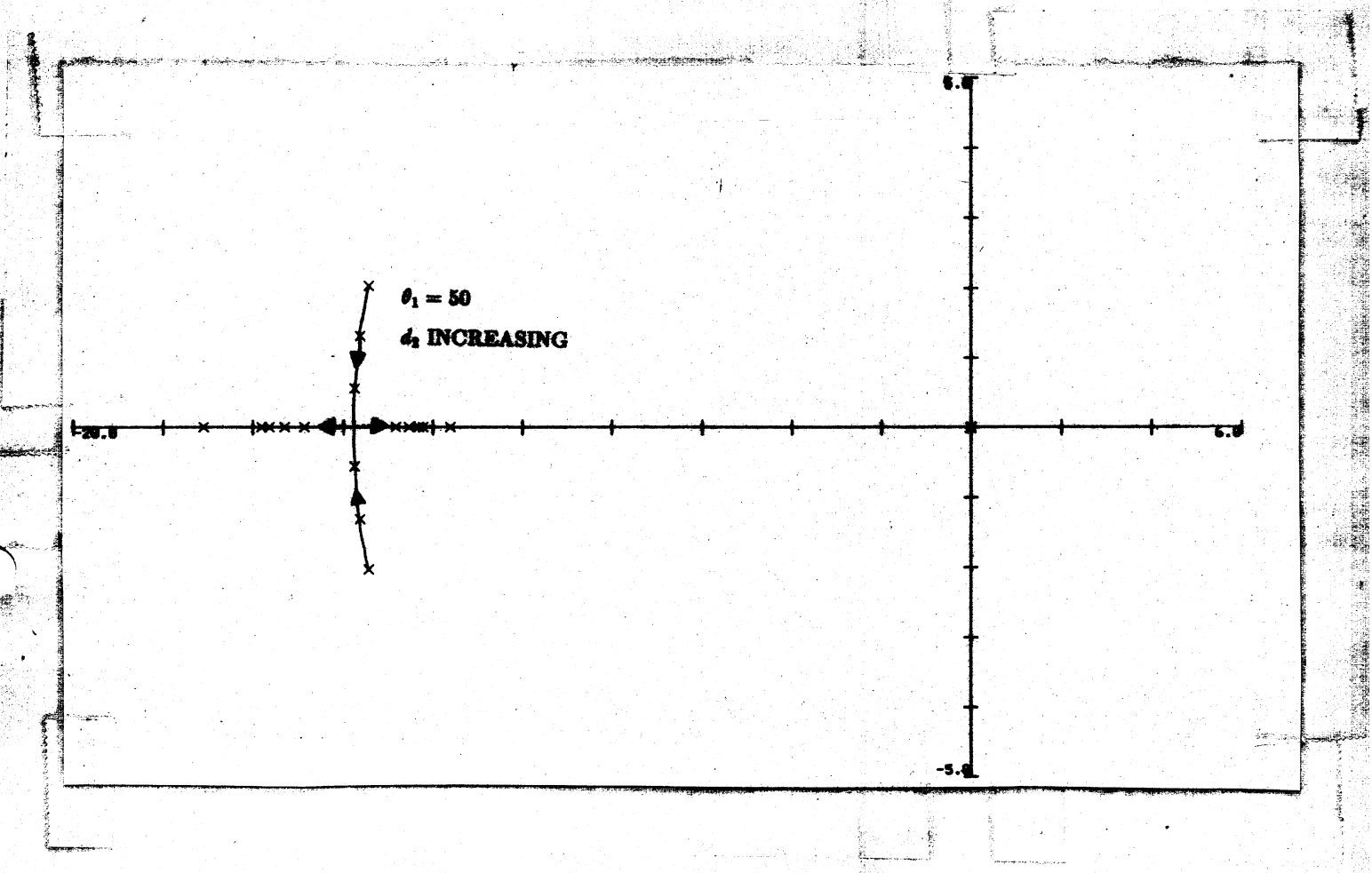


Figure 6.22: Root locus for the hybrid control on a polar manipulator with $S = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

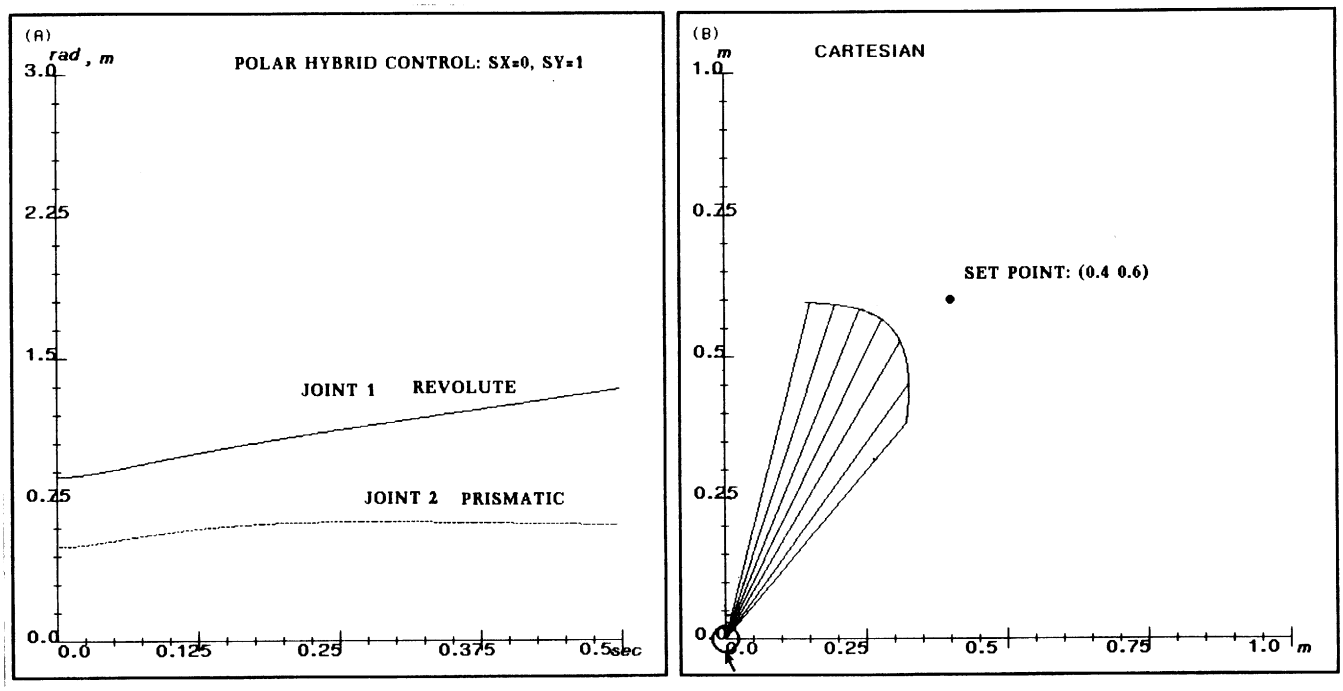


Figure 6.23: Simulation results for the hybrid control on a polar manipulator with $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

same controller results in a stable system on a polar manipulator. On the other hand, the stiffness and the resolved acceleration methods always produce stable results. The stiffness method is stable because it uses the Jacobian transpose for coordinate transformations, and the resolved acceleration method is stable because the inertia matrix is included in the controller. For cartesian trajectory following, the best performance was obtained with the resolved acceleration controller since its inverse dynamics compensation decouples the motion of the tip of the manipulator in cartesian directions.

Chapter 7

Force Control Experiments on the Direct Drive Arm

In Chapter 5, one-link force control experiments were included in order to demonstrate the issues of dynamic stability. In Chapter 6, since I wanted to isolate the kinematic instabilities from the dynamic instabilities, the experiments involved force controllers for a multi-link manipulator operating in free space without any interaction with the environment. This chapter verifies the stability analysis results of the two previous chapters by presenting results of stable force controller implementations on a multi-link manipulator against a stiff environment. As in Chapter 6, the second joint was locked at $\theta_2 = 180°$ and the first and the third joints were controlled to operate the arm in a planar configuration. For all of the experimental results in this chapter, the environment was always a stiff aluminum surface.

7.1 Experimental Setup

The resolved acceleration method, which was shown to be kinematically stable for any manipulator in any configuration, was implemented on the two links of the direct drive arm to control both the position and the interaction force with its environment. The implementation also utilized the dynamic stability results of Chapter 5 by using both joint torque control and dominant pole compensated wrist force feedback. The block diagram shown on Figure 7.1 and the equation below

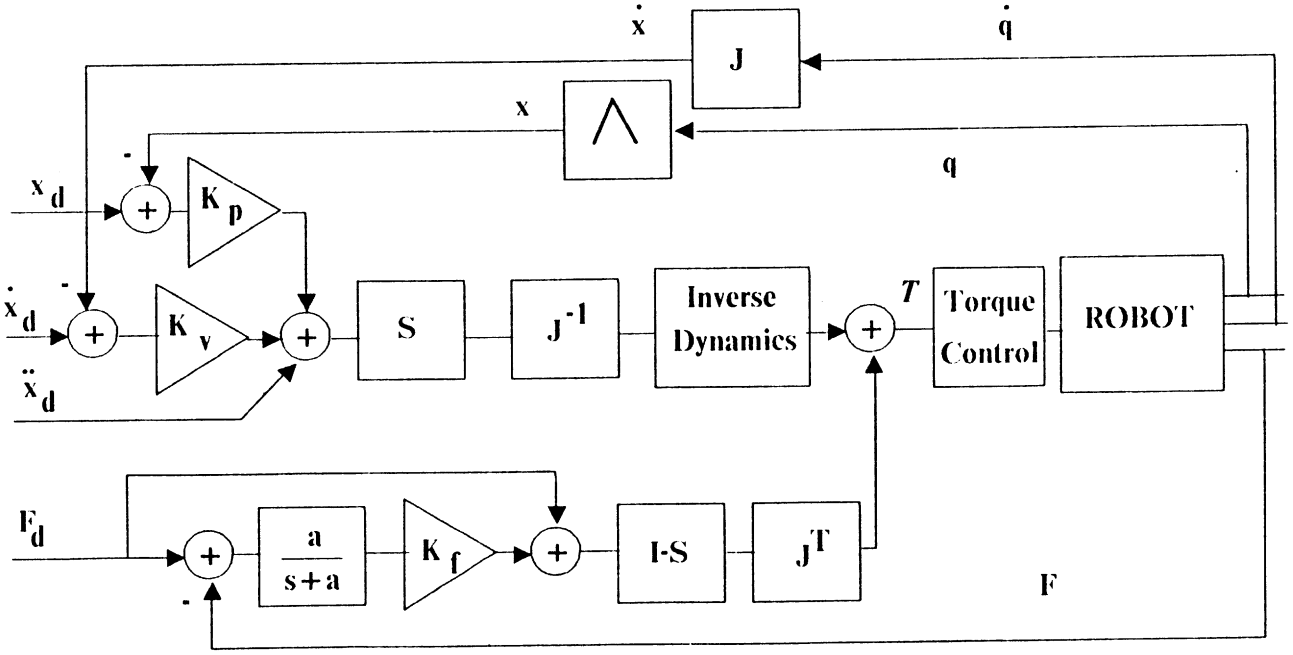


Figure 7.1: The structure of the resolved acceleration force/position controller for the DDARM

describe the general structure of the controller.

$$\tau = A(s)\{\mathbf{M(q)J^{-1}(q)[Sx^*_m - h(q,\dot{q})]} + \mathbf{b(q,\dot{q})} + \mathbf{g(q)} + \mathbf{J^T(I-S)K_f}\left(\frac{a}{s+a}\right)(\mathbf{f_d - f})\} \tag{7.1}$$

$$\mathbf{x^*_m} = \ddot{\mathbf{x}}_d + \mathbf{K}_v(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x})$$

$A(s)$ is the filter representing the torque servo loop at each joint and $\mathbf{K}_f$ is the wrist force sensor feedback gain matrix.

Joints 1 and 3 have torque inner loops, operating at $500Hz$, and the outer loops of both force and position operating at $100Hz$. Actually for Joint 1, the inner loop is really the analog current loop, to which the torque set points are commanded at $500Hz$ sampling intervals. An additional explicit torque servo mentioned in Chapter 5 was not necessary for this joint since the commanded and the measured torques matched quite well. However, since the match was not as good for the third joint, it has an explicit torque feedback in addition to the current feedback in order to reduce some of the undesirable nonlinear effects at the amplifier/motor circuitry. The wrist force sensor feedback was also included in order to increase the steady state force accuracy at the tip of the manipulator. But as discussed in Chapter 5, to insure stability by creating a dominant pole, the force sensor signal is processed through a low pass filter, whose pole is at $1Hz$. As before, the Barry

Wright FS6-120A 6-axis force/torque sensor is used as the wrist force sensor.

The same three types of inputs used in the single joint experiments were also used in the multi-link experiments. They are:

1. square wave force in the x-axis and constant position in the y-axis,
2. sine wave force in the x-axis and constant position in the y-axis, and
3. constant position in the y-axis and constant force in the x-axis in presence of the lateral movement of an eccentric cam (Fig. 5.20).

As before, the measured force data from the wrist force sensor are plotted without any low-pass filtering of the signal.

The control parameters used in the experiments are summarized below:

$$\mathbf{K}_f = \begin{bmatrix} 0.444 & 0 \\ 0 & 0.444 \end{bmatrix}$$

$$\mathbf{K}_p = \begin{bmatrix} 160 & 0 \\ 0 & 160 \end{bmatrix}$$

$$\mathbf{K}_v = \begin{bmatrix} 21 & 0 \\ 0 & 21 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{Force Filter Pole} = 1\,Hz$$

The force gain is set relatively low since the main dynamic performance should come from the joint torque control part.

7.2 Experimental Results

7.2.1 Square Wave Force Input

Using the resolved acceleration method, the x-axis is force controlled with the square wave inputs of $10\,N$ and $15\,N$, and the y-axis is position controlled to be stationary at $y = -0.669\,m$. Figure 7.2 shows a stable and fast step response. Response time, i.e., the delay plus the rise time is approximately $20\,ms$. Also, there is very

little coupling of the two axes, as verified by the trace of the error in y position. Except for some bias position error, due to the deadzones in the motors and also probably due to the roundoff errors in the computations, there is very little change in y position as force steps are commanded along the x-axis.

For the purpose of comparison, the performance of the resolved acceleration force controller without the wrist force sensor feedback is shown in Figure 7.3. Although the dynamic charateristics of the step response is similar to the one with the wrist force sensor, the steady state accuracy has deteriorated. This result agrees with the result of the one joint experiments (Fig. 5.17) of Chapter 5.

7.2.2 Sine Wave Force Input

Sine wave force response is used as a measure of bandwidth for the force controlled system. As in the step response experiments, the y-axis is position controlled to be stationary at $y = -0.669\,m$, and the x-axis is force controlled with the sine wave inputs of

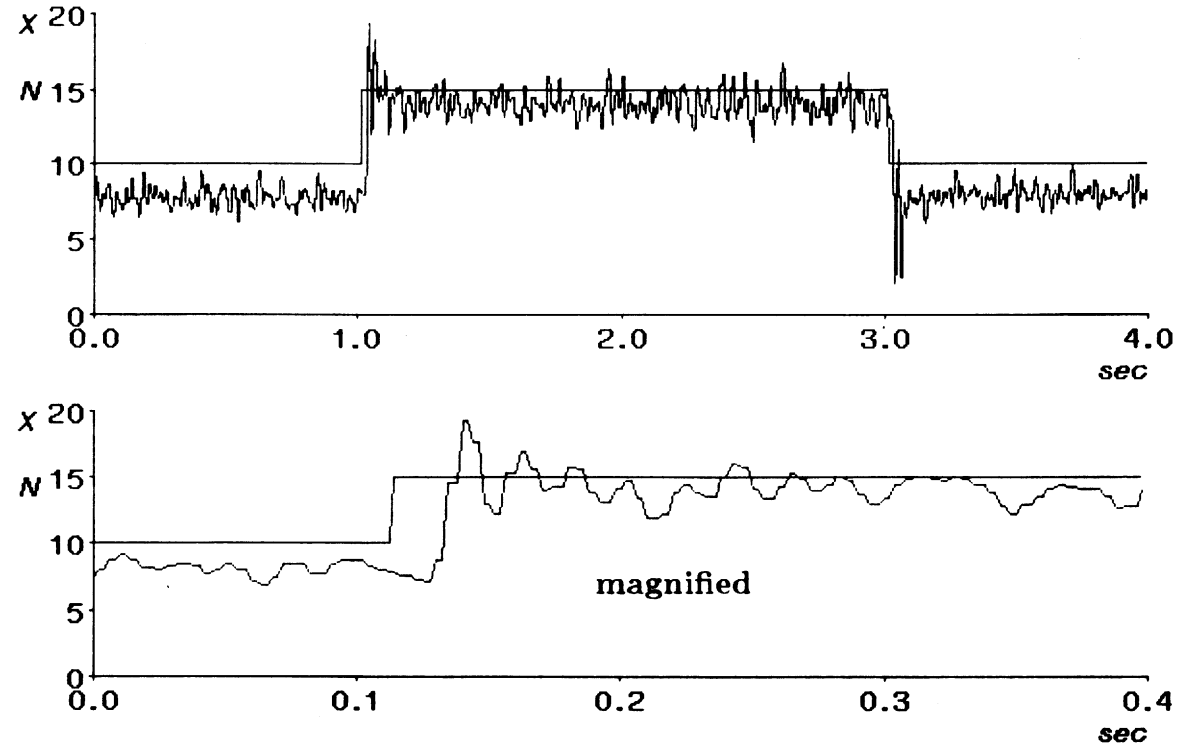
$$f_{command_x}(N) = 12.5 + 2.5\sin(wt)$$

The force and position responses to the $1Hz$ sine force input are shown in Figure 7.4. Both the force and the position traces have some offset errors, but the controller performs well dynamically. However, the manipulator did not respond as well to higher frequency inputs, although the one joint experiments showed good responses up to $20Hz$ inputs. This lower bandwidth is due to lower sampling frequencies ($100\,Hz$ outer force loop vs. $500\,Hz$) and less rigidity in the manipulator structure caused by the flexibilities at the joints connecting the links.

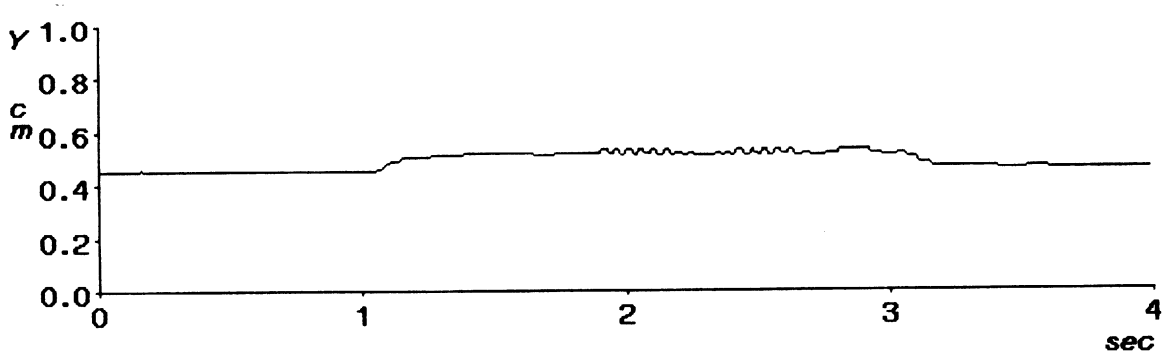
7.2.3 Following the Eccentric Cam

The manipulator is commanded to exert a constant force of $12.5\,N$ against the surface of the eccentric cam (Fig. 5.20) in the x-axis and stay stationary in the y-axis at $y = -0.669\,m$. As the cam turns, it produces a varying positional disturbance whose shape is approximately that of a sine wave.

The performance under these conditions is shown in Figure 7.5 for the same cam speed used for the one joint experiment. The plot of the x-axis position shows movements of the the manipulator in the x direction, while maintaining a constant force

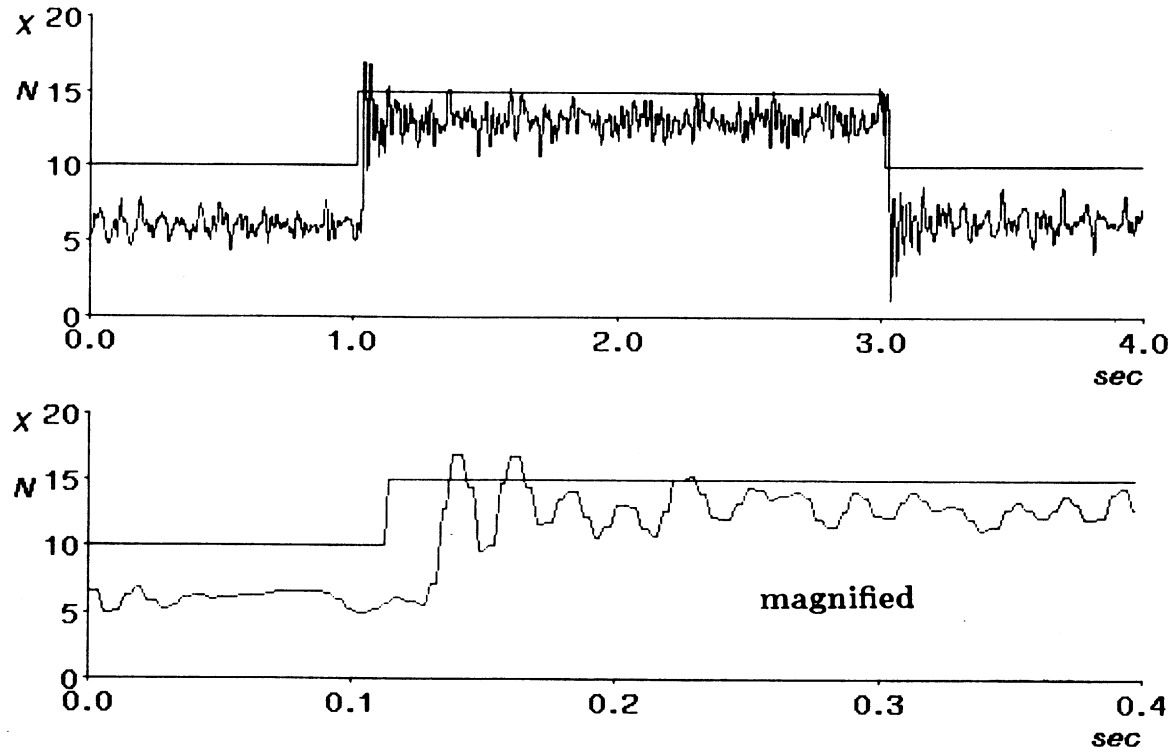


FORCE and TORQUE SENSING: X-AXIS STEP FORCE

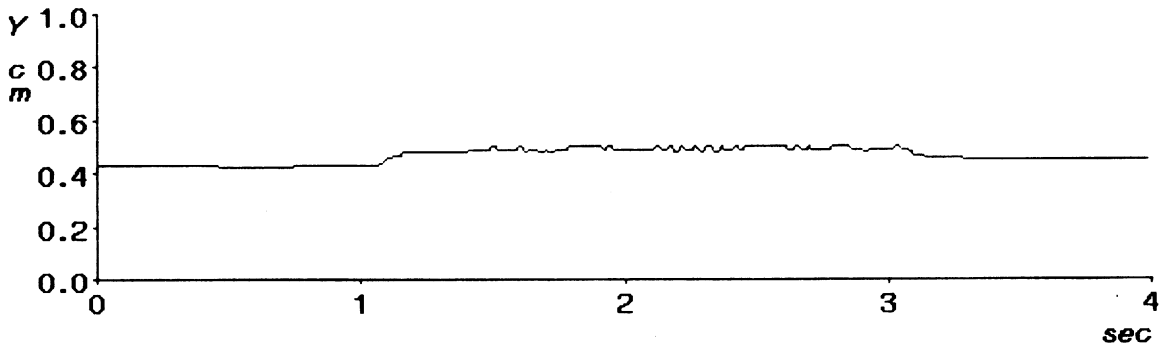


Y-AXIS POSITION CONTROL

Figure 7.2: Resolved acceleration: x-axis=force steps, y-axis=constant position



TORQUE SENSING ONLY: X-AXIS STEP FORCE



Y-AXIS POSITION CONTROL

Figure 7.3: Resolved acceleration without wrist force sensing: x-axis=force steps, y-axis=constant position

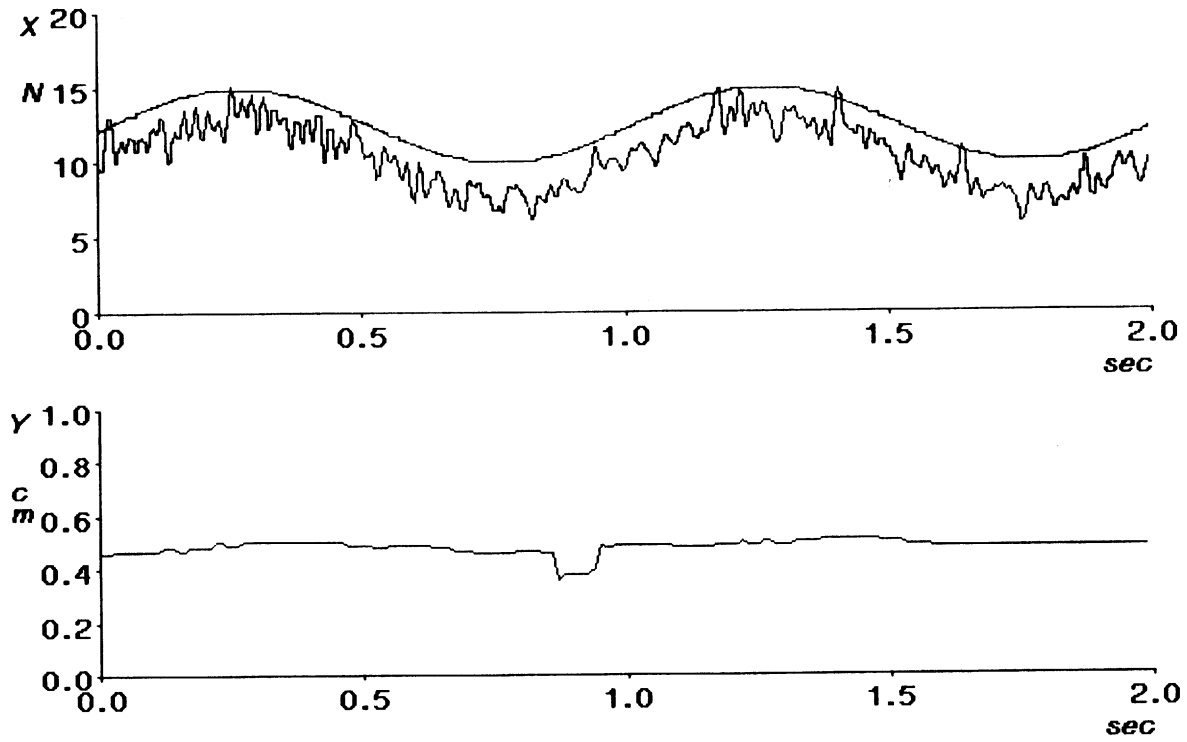


Figure 7.4: Resolved acceleration: x-axis=1 Hz sine wave force, y-axis=constant position

against the moving cam. The position trace in the y-axis shows that the resolved acceleration controller is able to maintain constant y position while following the cam in the x direction. Disregarding the high frequency noise, the variation of the measured force is slightly larger than the 10% limit that I defined as a disturbance rejection measure. Therefore, the frequency limit of the two link system measured in this way is slightly lower than 0.8 Hz, which was the limit of the one joint system.

7.3 Some Results with Stiffness Control

The stiffness method by Salisbury (1980) was also implemented on the direct drive arm in order to compare its performance against the resolved acceleration method while the manipulator is in contact with its environment. The gains for this controller were:

$$\mathbf{K}_p = \begin{bmatrix} 0 & 0 \\ 0 & 400 \end{bmatrix}, \quad \mathbf{K}_v = \begin{bmatrix} 40 & 0 \\ 0 & 40 \end{bmatrix}.$$

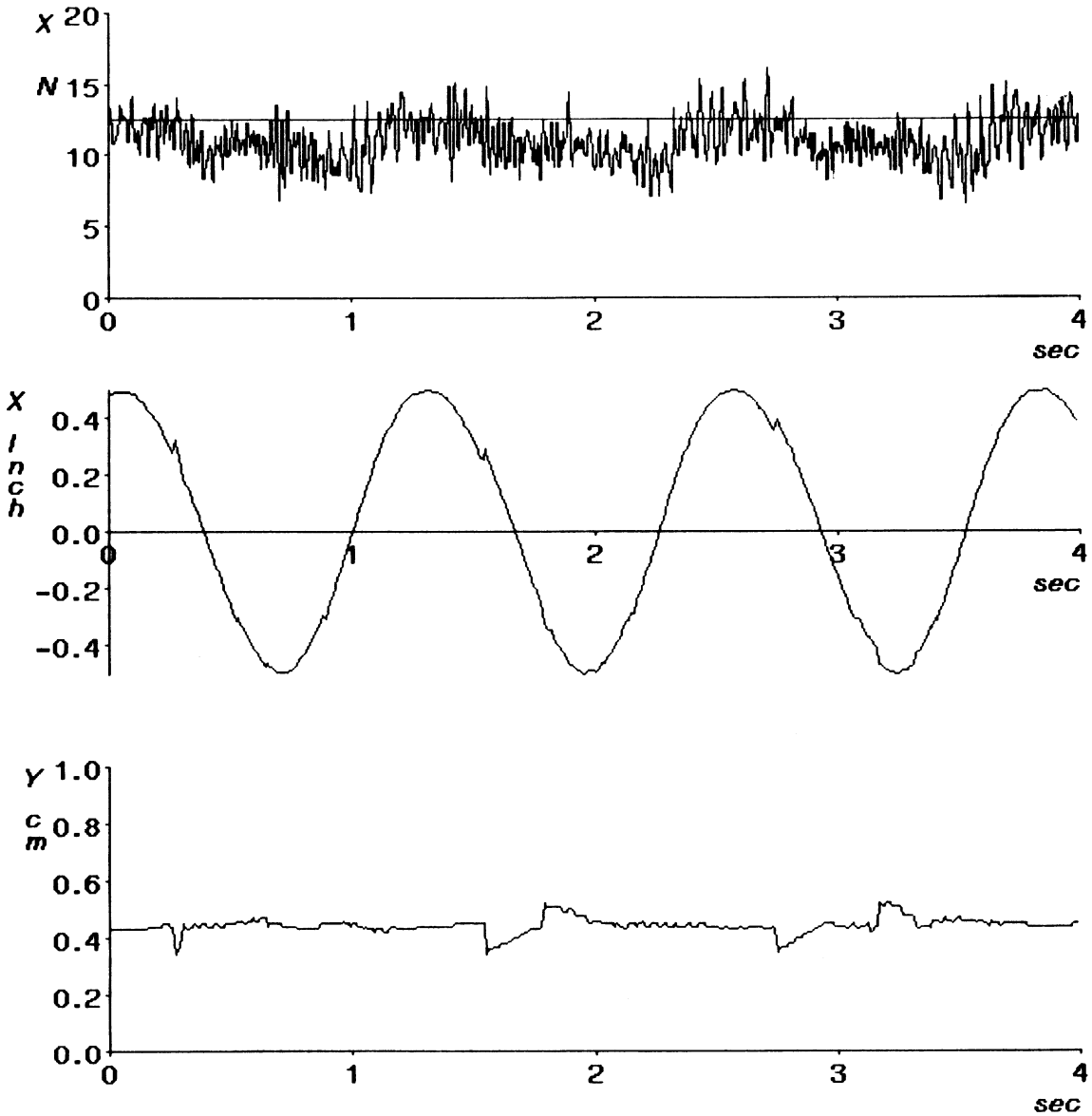


Figure 7.5: Resolved acceleration: x-axis=12.5 N force and positional disturbance, y-axis=constant position

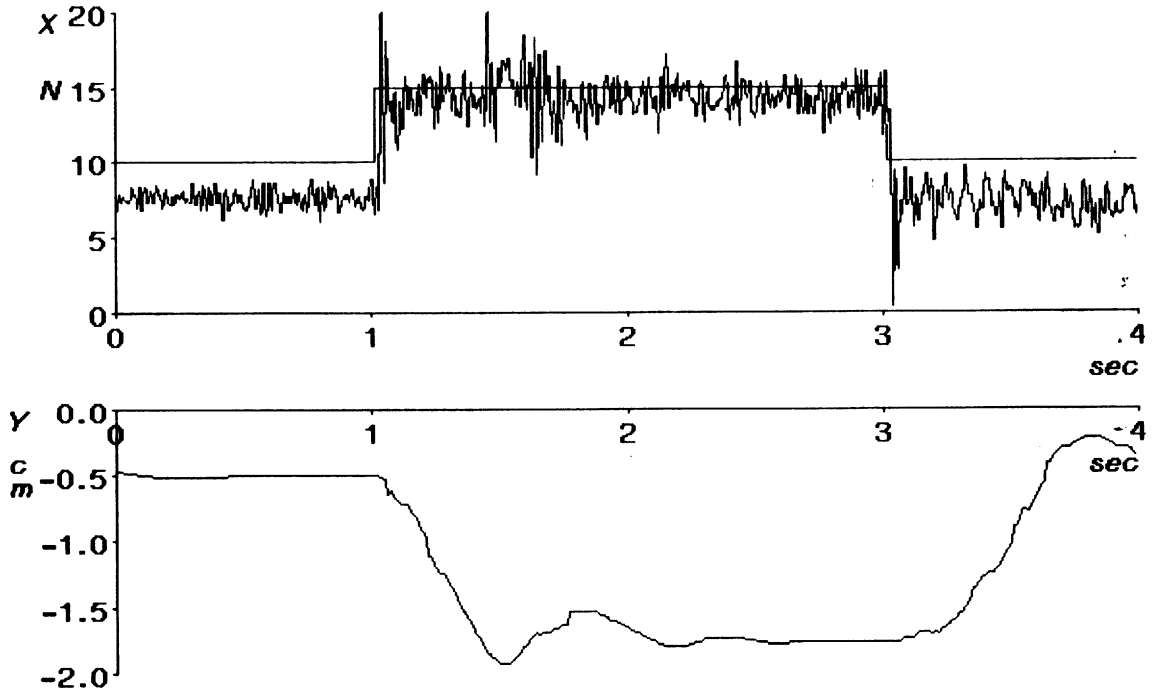
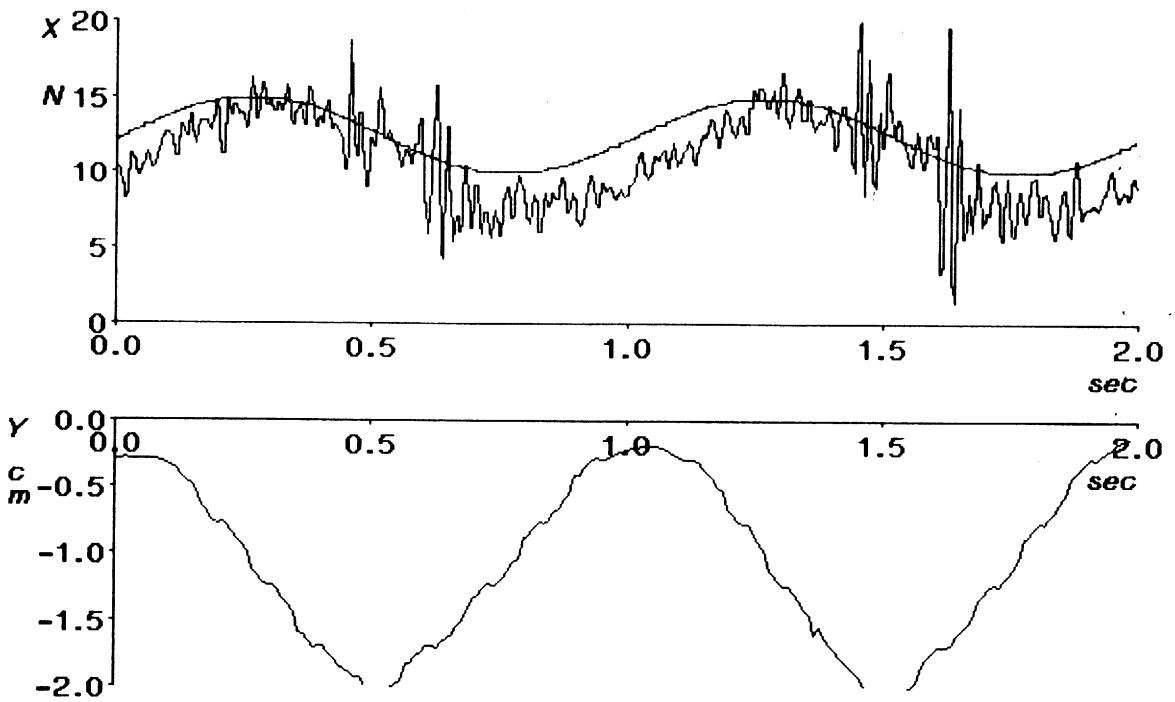


Figure 7.6: Stiffness Control: x-axis=force steps, y-axis=constant position

The stiffness in the x-axis is set to 0 so that pure force can be controlled in this direction in a similar way as the resolved acceleration method. This is one of two modes of operation for the stiffness control. For the other mode, one should set the stiffness in the x-axis to some desired value and then command a position set point beyond the contact surface. Since only one mode is evaluated, the comparisons in this section may not be the fairest.

As before, the manipulator is commanded to maintain constant y position, and square wave and sine wave force commands are given along x-axis. The results are presented in Figures 7.6 and 7.7. Although the dynamic force responses are slightly different from the results of the resolved acceleration, the main differences are that there are significant movements in the y direction with the stiffness control. Since the manipulator dynamics are not included in the stiffness control, the coupling between the two axes are much more severe in this case. The results of experiments using the cam positional disturbance are not presented because the stiffness controller could not keep the manipulator tip on the cam surface. The undesirable movement in the y direction was greater than the width of the cam.



STIFFNESS CONTROL: 1 Hz SINE RESPONSE

Figure 7.7: Stiffness Control: x-axis=1 Hz sine wave force, y-axis=constant position

Chapter 8

Conclusion and Future Research

8.1 Conclusion

There are two themes that are present throughout this thesis. One is the importance of accurate modelling of the manipulator system to be controlled and the second is the importance of experiments. Accurate modelling of a manipulator, either by estimation or by other methods, is important for both trajectory and force control. Experiments can not only verify theoretical results but also give new insights into the problems. For example, through the implementation results, I discovered that some of the link inertial parameters cannot be estimated and also that they do not affect joint torques. Chapter 6 on kinematic stability was motivated by my initial observation of instability when I tried to implement hybrid control on the DDARM. These issues were not apparent at first before the experiments. Therefore, these two ingredients, modelling and experiments, are essential in designing a high performance controller for a manipulator. Actually, they are important for designing a controller for any system.

Following those themes, the work presented in this thesis tried to address some of the problems in the area of robot control using the high performance MIT Serial Link Direct Drive Arm as the main experimental device. First, the issues of estimation of the dynamic models of a manipulator and its loads were studied since accurate modelling is the first step in any control design. Using the algorithms

presented, the inertial parameters of a manipulator and its loads were estimated very accurately. The estimated dynamic model was then used in evaluating several well known trajectory control algorithms on the DDARM. The experimental evaluations showed that the manipulator exhibits significantly improved trajectory following accuracy when its controller uses dynamic compensation.

The issues of force control of a manipulator were addressed next. The study started with a very simple model of a manipulator in order to understand some of the dynamic instability problems that occur when a manipulator interacts with a stiff environment. After understanding the stability problems, several methods of improving the stability were suggested and evaluated. It was shown both analytically and experimentally that there are also kinematic instability problems that can occur even when the manipulator is in free space. The problems are caused by the interaction of the inertia matrix with the Jacobian inverses used for kinematic coordinate transformations. However, the kinematic instabilities disappear when an accurate dynamic compensation of the manipulator is included in the feedback path by the resolved acceleration algorithm. This, then, brings us back to the importance of estimating the inertial parameters of a manipulator accurately.

8.2 Future Research

The work of this thesis is only the beginning of growing interests in high performance robot manipulation. I tried to address only some of the more imminent problems as accurate trajectory following and force control stability. There are vast numbers of difficult problems that have to be answered in the future, and many researchers have already begun addressing them. Three problems that are directly related to this thesis work are discussed below.

8.2.1 Manipulator

Conventional highly geared robots can be used successfully for applications that do not require high performance. It is not able to produce accurate high speed movements, nor interact with environments stably. Direct drive arms are high performance robots that eliminated or reduced many of the problems with the conventional robots. For the next several years, because of their speed, power, and

almost ideal dynamic characteristics, they will continue to provide researchers with many new opportunities to study effectively various topics such as trajectory control, force control, robot learning, and hand eye coordination. However, as I discussed in Chapter 1, they also have many shortcomings. It is, therefore, necessary to consider new approaches to high performance manipulator design. One of the new approaches is a lightweight flexible arm (Book, 1984; Cannon and Schmitz, 1984). This technology embodies a completely opposite philosophy from the technology of a heavy and rigid direct drive arm. Another very exciting approach is a light but rigid tendon driven arm. Recently, this approach was demonstrated successfully in designing multi-fingered robot hands (Salisbury, 1982; Jacobsen et al., 1986).

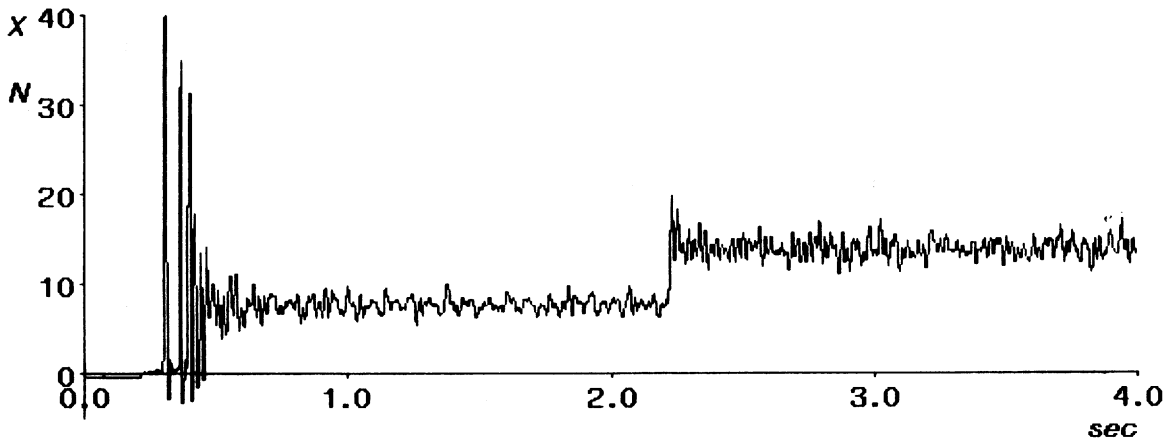
8.2.2 Collision

In my study of force control, I avoided the issues of collision that occurs at the initial contact phase. Especially for heavy manipulators, even at low speeds, the impact forces may be very large such that the environment and also the manipulator itself are damaged. During the force control experiments that I performed, although the manipulator remained stable, the impact forces were typically greater than 400% of the commanded force levels (Fig. 8.1).

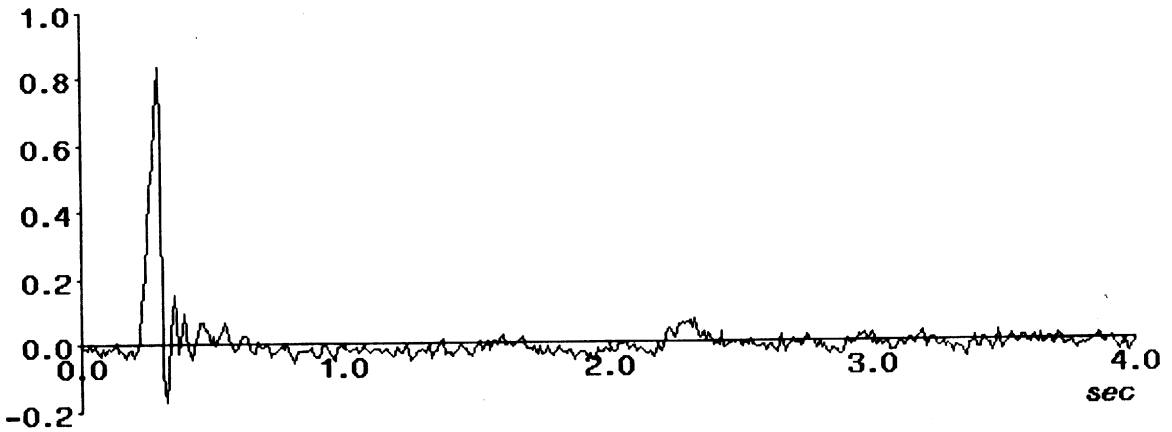
Featherstone (1984) developed a dynamic model of impact between a robot and its environment for simulation purposes. Khatib and Burdick (1986) introduced a transition phase of pure damping control to reduce the impact forces at collision. As discussed in Chapter 5, adding a compliant skin to the end effector will also reduce the impact forces. Redundant degrees of freedom may also be used effectively in handling collision. As humans bend knees when, for example, jumping down from a high platform, the robot can configure the redundant links so that they are compliant in the direction of collision and also transfer its momentum to other parts of the manipulator structure.

8.2.3 Fundamental Concepts of Compliant Behavior

A very important and difficult problem that needs to be answered in order to improve significantly the compliance capability of a manipulator is the fundamental behavior we want the robot to have in interacting with its uncertain environment. It is not clear whether we want the manipulator to behave as a spring (Salisbury,



(a) x-axis force(N) at collision.



(b) x-axis velocity(m/s) at collision.

Figure 8.1: Responses at collision

1980), a damper (Whitney, 1977), an impedance (Hogan, 1985a, 1985b, 1985c), or a hybrid of pure position/force source (Raibert and Craig, 1981, Khatib, 1983). It may be that none of the above approaches is the best one, and a completely different behavior is desired.

Appendices

Appendix 1: Closed Form Dynamics

The customized closed form equations of the dynamics of the MIT Serial Link Direct Drive Arm are presented in this appendix. To simplify the equations, the following notation is used:

$$s_2 = \sin\theta_2, \quad s_3 = \sin\theta_3, \quad c_2 = \cos\theta_2, \quad c_3 = \cos\theta_3.$$

Also, g = 9.80665 m/sec^2. The closed form equations are:

$$\begin{aligned}
n_1 = \; & \mathrm{SP}_4(\ddot{\theta}_1) + \mathrm{SP}_5(2\dot{\theta}_1\dot{\theta}_2 s_2 c_2 - \ddot{\theta}_1 c_2^2) + I_{xy_2}(2\ddot{\theta}_1 s_2 c_2 - 2\dot{\theta}_1\dot{\theta}_2 + 4\dot{\theta}_1\dot{\theta}_2 c_2^2) \\
& + I_{xz_2}(\ddot{\theta}_2 s_2 + v_2^2 c_2) + \mathrm{SP}_1(\ddot{\theta}_2 c_2 - v_2^2 s_2) \\
& + m_3 c_{x_3}(l_2)(2\ddot{\theta}_1 s_3 + 2\dot{\theta}_2\dot{\theta}_3 s_2 s_3 - \ddot{\theta}_3 s_3 c_2 + 2\dot{\theta}_1\dot{\theta}_3 c_3 - \ddot{\theta}_2 s_2 c_3 - v_2^2 c_2 c_3 - v_3^2 c_2 c_3) \\
& + m_3 c_{y_3}(l_2)(\ddot{\theta}_2 s_2 s_3 - 2\dot{\theta}_1\dot{\theta}_3 s_3 + v_2^2 s_3 c_2 + v_3^2 s_3 c_2 + 2\ddot{\theta}_1 c_3 + 2\dot{\theta}_2\dot{\theta}_3 s_2 c_3 - \ddot{\theta}_3 c_2 c_3) \\
& + \mathrm{SP}_2(\ddot{\theta}_1 c_2^2 - \ddot{\theta}_1 - \dot{\theta}_2\dot{\theta}_3 s_2 - 2\dot{\theta}_1\dot{\theta}_2 s_2 c_2 - 2\dot{\theta}_1\dot{\theta}_3 s_3 c_3 + \ddot{\theta}_2 s_2 s_3 c_3 \\
& \quad + v_2^2 s_3 c_2 c_3 + 2\dot{\theta}_1\dot{\theta}_3 s_3 c_2^2 c_3 + \ddot{\theta}_1 c_3^2 + 2\dot{\theta}_2\dot{\theta}_3 s_2 c_3^2 + 2\dot{\theta}_1\dot{\theta}_2 s_2 c_2 c_3 c_3 - \ddot{\theta}_1 c_2^2 c_3 c_3) \\
& + I_{xy_3}(2\dot{\theta}_1\dot{\theta}_3 - \ddot{\theta}_2 s_2 - v_2^2 c_2 - 2\dot{\theta}_1\dot{\theta}_3 c_2^2 - 2\ddot{\theta}_1 s_3 c_3 - 4\dot{\theta}_2\dot{\theta}_3 s_2 s_3 c_3 \\
& \quad - 4\dot{\theta}_1\dot{\theta}_2 s_2 s_3 c_2 c_3 + 2\ddot{\theta}_1 s_3 c_2^2 c_3 - 4\dot{\theta}_1\dot{\theta}_3 c_3^2 + 2\ddot{\theta}_2 s_2 c_3^2 + 2\dot{\theta}_2\dot{\theta}_2 c_2 c_3^2 + 4\dot{\theta}_1\dot{\theta}_3 c_2 c_2 c_3^2) \\
& + I_{xz_3}(\dot{\theta}_2\dot{\theta}_2 s_2 s_3 - v_3^2 s_2 s_3 - \ddot{\theta}_2 s_3 c_2 + 2\dot{\theta}_1\dot{\theta}_3 s_2 s_3 c_2 + 2\dot{\theta}_1\dot{\theta}_2 c_3 + \ddot{\theta}_3 s_2 c_3 - 2\ddot{\theta}_1 s_2 c_2 c_3 \\
& \quad - 4\dot{\theta}_1\dot{\theta}_2 c_2^2 c_3) \\
& + I_{yz_3}(2\ddot{\theta}_1 s_2 s_3 c_2 - 2\dot{\theta}_1\dot{\theta}_2 s_3 - \ddot{\theta}_3 s_2 s_3 + 4\dot{\theta}_1\dot{\theta}_2 s_3 c_2^2 + v_2^2 s_2 c_3 - v_3^2 s_2 c_3 - \ddot{\theta}_2 c_2 c_3 \\
& \quad + 2\dot{\theta}_1\dot{\theta}_3 s_2 c_2 c_3) \\
& + I_{zz_3}(\dot{\theta}_2\dot{\theta}_3 s_2 - \ddot{\theta}_3 c_2 - 2\dot{\theta}_1\dot{\theta}_2 s_2 c_2 + \ddot{\theta}_1 c_2^2)
\end{aligned}$$

$$
\begin{aligned}
n_2 = \; & m_2 c_{x_2}(\mathrm{g}c_2) + \mathrm{SP}_6(\mathrm{g}s_2) - \mathrm{SP}_5(v_1^2 s_2 c_2) + I_{xy_2}(v_1^2 - 2v_1^2 c_2^2) + I_{xz_2}(\ddot{\theta}_1 s_2) \\
& + \mathrm{SP}_1(\ddot{\theta}_1 c_2) + \mathrm{SP}_3(\ddot{\theta}_2) + m_3 c_{x_3}(\mathrm{g}c_2 c_3 - l_2 \ddot{\theta}_1 s_2 c_3) + m_3 c_{y_3}(l_2 \ddot{\theta}_1 s_2 s_3 - \mathrm{g}s_3 c_2) \\
& + \mathrm{SP}_2(\dot{\theta}_1 \dot{\theta}_1 s_2 c_2 - \dot{\theta}_1 \dot{\theta}_3 s_2 + 2\dot{\theta}_2 \dot{\theta}_3 s_3 c_3 + \ddot{\theta}_1 s_2 s_3 c_3 - \ddot{\theta}_2 c_3^2 + 2\dot{\theta}_1 \dot{\theta}_3 s_2 c_3^2 - \dot{\theta}_1 \dot{\theta}_1 s_2 c_2 c_3^2) \\
& + I_{xy_3}(2\ddot{\theta}_2 s_3 c_3 - 2\dot{\theta}_2 \dot{\theta}_3 - \ddot{\theta}_1 s_2 - 4\dot{\theta}_1 \dot{\theta}_3 s_2 s_3 c_3 + 2v_1^2 s_2 s_3 c_2 c_3 + 4\dot{\theta}_2 \dot{\theta}_3 c_3^2 + 2\ddot{\theta}_1 s_2 c_3^2) \\
& + I_{xz_3}(\ddot{\theta}_3 s_3 - \ddot{\theta}_1 s_3 c_2 - v_1^2 c_3 + v_3^2 c_3 - 2\dot{\theta}_1 \dot{\theta}_3 c_2 c_3 + 2v_1^2 c_2^2 c_3) \\
& + I_{yz_3}(v_1^2 s_3 - v_3^2 s_3 + 2\dot{\theta}_1 \dot{\theta}_3 s_3 c_2 - 2v_1^2 s_3 c_2^2 + \ddot{\theta}_3 c_3 - \ddot{\theta}_1 c_2 c_3) \\
& + I_{zz_3}(v_1^2 s_2 c_2 - \dot{\theta}_1 \dot{\theta}_3 s_2)
\end{aligned}
$$

$$
\begin{aligned}
n_3 = \; & m_3 c_{x_3}(-\mathrm{g}s_2 s_3 - l_2 \ddot{\theta}_1 s_3 c_2 - l_2 v_1^2 c_3) + m_3 c_{y_3}(l_2 v_1^2 s_3 - l_2 \ddot{\theta}_1 c_2 c_3 - \mathrm{g}s_2 c_3) \\
& + \mathrm{SP}_2(\dot{\theta}_1 \dot{\theta}_2 s_2 + v_1^2 s_3 c_3 - v_2^2 s_3 c_3 - v_1^2 s_3 c_3 c_2^2 - 2\dot{\theta}_1 \dot{\theta}_2 s_2 c_3^2) \\
& + I_{xy_3}(v_2^2 - v_1^2 + v_1^2 c_2^2 + 4\dot{\theta}_1 \dot{\theta}_2 s_2 s_3 c_3 + 2v_1^2 c_3^2 - 2v_2^2 c_3^2 - 2v_1^2 c_2^2 c_3^2) \\
& + I_{xz_3}(\ddot{\theta}_2 s_3 - v_1^2 s_2 s_3 c_2 + \ddot{\theta}_1 s_2 c_3 + 2\dot{\theta}_1 \dot{\theta}_2 c_2 c_3) \\
& + I_{yz_3}(\ddot{\theta}_2 c_3 - \ddot{\theta}_1 s_2 s_3 - 2\dot{\theta}_1 \dot{\theta}_2 s_3 c_2 - v_1^2 s_2 c_2 c_3) \\
& + I_{zz_3}(\ddot{\theta}_3 + \dot{\theta}_1 \dot{\theta}_2 s_2 - \ddot{\theta}_1 c_2)
\end{aligned}
$$

In these equations, there are 15 reduced inertial parameters: $m_2 c_{x_2}$, I_{xy_2}, I_{xz_2}, $m_3 c_{x_3}$, $m_3 c_{y_3}$, I_{xy_3}, I_{xz_3}, I_{yz_3}, I_{zz_3}, SP_1, SP_2, SP_3, SP_4, SP_5, SP_6. The SP_i variables are abbreviations for the following linear combinations:

$$
\begin{aligned}
\mathrm{SP}_1 &= m_3 c_{z_3} l_2 + I_{yz_2} \\
\mathrm{SP}_2 &= I_{xx_3} - I_{yy_3} \\
\mathrm{SP}_3 &= I_{zz_2} + I_{xz_3} \\
\mathrm{SP}_4 &= I_{zz_1} + I_{xz_2} + I_{xx_3} + m_3 l_2^2 \\
\mathrm{SP}_5 &= I_{xz_2} + I_{xz_3} - I_{yy_2} \\
\mathrm{SP}_6 &= m_3 c_{z_3} - m_2 c_{y_2}
\end{aligned}
$$

Eleven inertial parameters do not appear in these equations at all, and are completely unidentifiable: m_1, $m_1 c_{x_1}$, $m_1 c_{y_1}$, $m_1 c_{z_1}$, I_{xx_1}, I_{xy_1}, I_{xz_1}, I_{yy_1}, I_{yz_1}, m_2, $m_2 c_{z_2}$.

Appendix 2: Stability Robustness

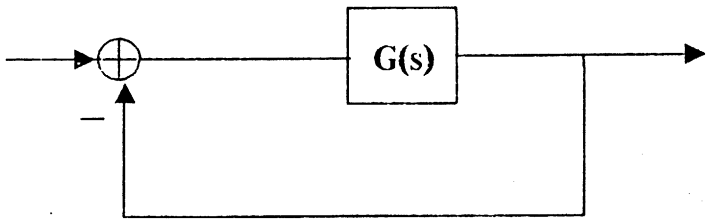


Figure A.1: Nominal SISO system

This appendix is a summary of the discussion in Lehtomaki (1982) on the stability robustness using the Nyquist criterion for a single input single output (SISO) system,

Let's consider a SISO system described in Figure A.1 with nominal loop transfer function $G(s)$. If we have a modelling error, the actual loop transfer function is represented as the perturbed $\tilde{G}(s)$. The stability robustness is determined by the distance that the Nyquist plot of $\tilde{G}(s)$ avoids the (-1 ,0) point in the complex plane (Fig. A.2). The situation in Figure A.2 shows that if the nominal closed-loop system with $G(s)$ were stable, then the perturbed system with $\tilde{G}(s)$ would also be stable since the number of encirclements of the (-1, 0) point has not changed.

For any ω, the distance between the (-1, 0) point and $G(jw)$ is given by

$$d(w) = |1 + G(j\omega)|, \tag{A.1}$$

and the distance between $\tilde{G}(j\omega)$ and $G(j\omega)$ is,

$$p(w) = |\tilde{G}(j\omega) - G(j\omega)|. \tag{A.2}$$

Then, from the Nyquist plot, it is clear that the perturbed closed-loop system is stable if

$$|1 + G(j\omega)| > |\tilde{G}(j\omega) - G(j\omega)|. \tag{A.3}$$

If we define the modelling error as in Figure 5.2, then

$$\tilde{G}(s) = (1 + E(s))G(s)$$

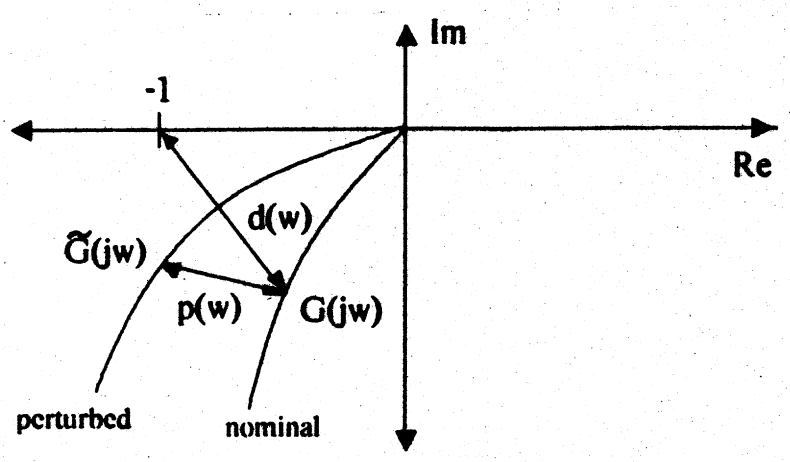


Figure A.2: Nyquist plot of the nominal and the perturbed (actual) system

and

$$E(s) = \frac{\tilde{G}(s) - G(s)}{G(s)}.$$

The relation for stability robustness is obtained for this error model by dividing (A.3) by $G(j\omega)$:

$$\left|\frac{1 + G(j\omega)}{G(j\omega)}\right| = |1 + G^{-1}(j\omega)| > \left|\frac{\tilde{G}(j\omega) - G(j\omega)}{G(j\omega)}\right| = |E(j\omega)|. \qquad \text{(A.4)}$$

Appendix 3: Adaptive Observer

The new plant equation (5.20) can be derived from the original equation (5.18) in the following way. This derivation is slightly simpler than the method presented by Narendra and Kudva (1974). (5.18) and the filter equations (5.19) are repeated below:

$$Df_s = \ddot{x}_E + \beta\dot{x}_E + \Omega x_E \tag{5.18}$$

$$\dot{y} = -\lambda y + x_E, \quad \dot{w} = -\lambda w + f_s \tag{5.19}$$

The above equations can be written in the frequency domain as,

$$Df_s(s) = (s^2 + \beta s + \Omega)x_E(s) \tag{A.5}$$

$$(s+\lambda)y(s) = x_E(s), \quad (s+\lambda)w(s) = f_s(s). \tag{A.6}$$

Then by substituting (A.6) into (A.5),

$$(s^2 + \beta s + \Omega)(s+\lambda)y(s) = D(s+\lambda)w(s). \tag{A.7}$$

Cancelling $(s+\lambda)$ from both sides of (A.7), we get

$$(s^2 + \beta s + \Omega)y(s) = Dw(s). \tag{A.8}$$

Since $sy(s) = x_E(s) - \lambda y(s)$ from (A.6), we can substitute for $sy(s)$ in (A.8) so that,

$$x_E(s) = -\beta x_E(s) - \Omega y(s) + Dw(s) + \lambda x_E(s) + \lambda(\beta - \lambda)y(s) \tag{A.9}$$

In time domain, (A.9) is the desired equation (5.20):

$$\dot{x}_E = -\beta x_E - \Omega y + Dw + \lambda x_E + \lambda(\beta - \lambda)y \tag{5.20}$$

Appendix 4: Operational Space and Resolved Acceleration

In the reports by Khatib (Khatib, 1983; Khatib and Burdick, 1986), the dynamics of a manipulator in operational space or end effector cartesian space are described by

$$\mathbf{\Lambda}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}(x) = \mathbf{f}. \qquad [1]^1$$

In joint coordinate system, the same dynamics are described by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}. \qquad (A.10)$$

$\mathbf{M}(\mathbf{q})$ and $\mathbf{\Lambda}(\mathbf{x})$ are related by

$$\mathbf{M}(\mathbf{q}) = \mathbf{J}^T(\mathbf{q})\mathbf{\Lambda}(\mathbf{x})\mathbf{J}(\mathbf{q}) \qquad [3]$$

or

$$\mathbf{J}^{-T}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}^{-1}(\mathbf{q}) = \mathbf{\Lambda}(\mathbf{x}) = \mathbf{\Lambda}(\mathbf{q}). \qquad (A.11)$$

Also, the operational space force vector $\mathbf{f}$ and the joint torque vector $\boldsymbol{\tau}$ are related by

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{f}. \qquad [4]$$

For a decoupled end effector motion commanded by $\mathbf{f}_m^*$,

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{J}^T(\mathbf{q})\mathbf{\Lambda}(\mathbf{q})\mathbf{f}_m^* + \mathbf{J}^T\boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{J}^T\mathbf{p}(\mathbf{x}) \\ &= \mathbf{J}^T(\mathbf{q})\mathbf{\Lambda}(\mathbf{q})\mathbf{f}_m^* + \tilde{\mathbf{b}}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}). \end{aligned} \qquad [7]$$

where

$$\tilde{\mathbf{b}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{J}^T\mathbf{\Lambda}(\mathbf{q})\mathbf{h}(\mathbf{q},\dot{\mathbf{q}}), \qquad [8]$$

$$\mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) = \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}. \qquad [9]$$

Typically, for position or trajectory control, a linear second order behavior is commanded so that

$$\mathbf{f}_m^* = \ddot{\mathbf{x}}_d + \mathbf{K}_v(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) \qquad (A.12)$$

[1] In this appendix, the equation numbers in brackets refer to the equation numbers in (Khatib and Burdick, 1986)

Then, for a hybrid force/position control, the joint command vector with the operational space method is

$$\tau = \mathbf{J}^T(\mathbf{q})[\mathbf{\Lambda}(\mathbf{q})\mathbf{S}\mathbf{f}_m^* + (\mathbf{I} - \mathbf{S})\mathbf{f}_a^*] + \tilde{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \qquad [18]$$

where $\mathbf{f}_a^*$ is the command vector for force control.

Substituting (A.11) and [8] into [18],

$$\begin{aligned} \tau &= \mathbf{J}^T(\mathbf{q})\mathbf{J}^{-T}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J}^{-1}(\mathbf{q})\mathbf{S}\mathbf{f}_m^* + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}^T\mathbf{J}^{-T}(\mathbf{q})M(\mathbf{q})\mathbf{J}^{-1}(\mathbf{q})\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \\ & \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})(\mathbf{I} - \mathbf{S})\mathbf{f}_a^* \end{aligned} \qquad (A.13)$$

Simplifying the above equation,

$$\tau = \mathbf{M}(\mathbf{q})\mathbf{J}^{-1}(\mathbf{q})[\mathbf{S}\mathbf{f}_m^* - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})] + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})(\mathbf{I} - \mathbf{S})\mathbf{f}_a^*. \qquad (A.14)$$

This equation is identical to the equation (6.2) for the modified resolved acceleration controller presented in Chapters 6 and 7.

References

An, C.H., Atkeson, C.G., and Hollerbach, J.M., 1985, "Estimation of inertial parameters of rigid body links of manipulators," *Proc. 24th IEEE Conf. Decision and Control*, Fort Lauderdale, Dec. 11-13, pp. 990-995.

An, C.H., Atkeson, C.G., and Hollerbach, J.M., 1986, "Experimental determination of the effect of feedforward control on trajectory tracking errors," *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, April 7-10, pp. 55-60.

An, C.H., Atkeson, C.G., Griffiths, J.D., and Hollerbach, J.M., 1986, "Experimental evaluation of feedforward and computed torque control," *Proc. Sixth CISM-IFToMM Sym. of Theory and Practice of Robots and Manipulators*, Cracow, Poland, Sept. 9-12, in press.

Armstrong, B., Khatib, O. and Burdick, J., 1986, "The explicit dynamic model and inertial parameters of the PUMA 560 arm," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 510-518.

Asada, H., and Kanade, T., 1981, "Design of Direct-Drive Mechanical Arms," CMU-RI-TR-81-1, Carnegie-Mellon Univ., Robotics Institute, April 28.

Asada, H., Kanade, K., and Takeyama, I., 1983, "Control of a direct-drive arm," *Trans. of ASME*, 105, pp. 136-142.

Asada, H., and Youcef-Toumi, K., 1984, "Analysis and design of a direct-drive arm with a five-bar-link parallel drive mechanism," *ASME J. Dynamic Systems, Meas., Control*, 106, pp. 225-230.

Asada, H., Youcef-Toumi, K., and Lim, S.K., 1984, "Joint torque measurement of a direct-drive arm," *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, Dec. 12-14, pp. 1332-1337.

Asada, H., and Ro, I.H., 1985, "A linkage design for direct-drive robot arms," *J. Mechanisms, Transmissions, and Automation in Design*, 107, pp. 536-540.

Atkeson, C.G., An, C.H., and Hollerbach, J.M., 1985a, "Estimation of inertial parameters of manipulator loads and links," *Preprints of 3rd Int. Symp. Robotics Research*, Gouvieux (Chantilly), France, October 7-11, pp. 32-39.

Atkeson, C.G., An, C.H., and Hollerbach, J.M., 1985b, "Rigid body load identification for manipulators," *Proc. 24th Conf. Decision and Control*, Fort Lauderdale, Dec. 11-13, pp. 996-1002.

Atkeson, C.G., 1986, Roles of Knowledge in Motor Learning, Ph.D. Thesis, MIT, Dept. of Brain and Cognitive Sciences, Sept..

Bejczy, A.K., 1974, "Robot arm dynamics and control," 33-669, Jet Propulsion Laboratory, Feb..

Book, W.J, 1984, "Recursive Lagrangian dynamics of flexible manipulator arms," *The International Journal of Robotics Research*, 3 no. 3, pp. 87-101.

Brady, M., Hollerbach, J.M., Johnson, T.L., Lozano-Perez, T., Mason, M., editors, 1983, *Robot Motion: Planning and Control*, Cambridge, MA, MIT Press.

Caine, M. E., 1985, Chamferless Assembly of Rectangular Parts in Two and Three Dimensions, SM Thesis, MIT, Mechanical Engineering Department, June.

Cannon, R.H., Schmitz, E., 1984, "Initial experiments on the end-point control of a flexible one-link robot," *The International Journal of Robotics Research*, 3 no. 3, pp. 62-75.

Coiffet, P., 1983, *Robot Technology: Interaction With The Environment. Vol. 2*, Englewood Cliffs, N.J., Prentice-Hall.

Curran, R., and Mayer, G., 1985, "The architecture of the AdeptOne direct-drive robot," *Proc. American Control Conf.*, Boston, June 19-21, pp. 716-721.

Drake, S.K., and Simunovic, S.N., 1977, "The use of compliance in a robot assembly system," *Preprints of IFAC Symposium on Information and Control Problems in Manufacturing Technology*, Tokyo.

Eppinger, S.D., and Seering, W.P., 1986, "On dynamic models of robot force control," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 29-34.

Featherstone, R, 1984, Robot Dynamics Algorithms, Ph.D. Thesis, University of Edinburgh.

Franklin, G.F. and Powel, J.D., 1980, *Digital Control of Dynamic Systems*, Reading, MA, Addison-Wesley.

Gilbert, E.G., and Ha, I.J., 1984, "An approach to nonlinear feedback control with applications to robotics," *IEEE Trans. Systems, Man, Cybern*, SMC-14, pp. 879-884.

Golla, D.F., Garg, S.C., and Hughes, P.C., 1981, "Linear-state feedback control of manipulators," *Mech. Machine Theory*, 16, pp. 93-103.

Golub, G.H., and Van Loan, C.F., 1983, *Matrix computations*, John Hopkins University Press.

Good, M.C., Sweet, L.M., and Strobel, K.L., 1985, "Dynamic models for control

system design of integrated robot and drive systems," *ASME J. Dynamic Systems, Meas., Control*, 107, pp. 53-59.

Goor, R.M., 1985, "A new approach to minimum time robot control," *ASME Winter Annual Meeting*, Miami Beach, Florida, Nov. 17-22, pp. 1-12.

Griffiths, John D., 1986, Experimental Evaluation of Computed Torque Control, B.S. Thesis, MIT, Mechanical Eng..

Hayati, S.A., 1983, "Robot arm geometric link parameter estimation," *Proc. 22nd IEEE Conf. Decision and Control*, San Antonio, Dec. 14-16, pp. 1477-1483.

Hogan, N, 1985a, "Impedance control: An approach to manipulation: Part I - Theory," *ASME J. of Dynamic Systems, Measurement, and Control*, 107, pp. 1-7.

Hogan, N, 1985b, "Impedance control: An approach to manipulation: Part II - Implementation," *ASME J. of Dynamic Systems, Measurement, and Control*, 107, pp. 8-16.

Hogan, N, 1985c, "Impedance control: An approach to manipulation: Part III - Applications," *ASME J. of Dynamic Systems, Measurement, and Control*, 107, pp. 17-24.

Jacobsen, S.C., Iversen, E.K., Knutti, D.F., Johnson, R.T.,and Biggers, K.B., 1986, "Design of the Utah/MIT dexterous hand," *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, April 7-10, pp. 1520-1532.

Kazerooni, H, 1985, A Robus Design Method for Impedance Control of Constrained Dynamic Systems, Ph.D. Thesis, MIT, Mechanical Engineering Department, Feb..

Kazerooni, H., Houpt, P.K., and Sheridan, T.B., 1986a, "The fundamental concepts of robust compliant motion for robot manipulators," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 418-427.

Kazerooni, H., Houpt, P.K., and Sheridan, T.B., 1986b, "Robust compliant motion for manipulators, Part I: The fundamental concepts of compliant motion," *IEEE Journal of Robotics and Automation*, RA-2, pp. 83-92.

Kazerooni, H., Houpt, P.K., and Sheridan, T.B., 1986c, "Robust compliant motion for manipulators, Part II: Design method," *IEEE Journal of Robotics and Automation*, RA-2, pp. 93-105.

Khalil, W., Gautier, M., and Kleinfinger, J.F., 1986, "Automatic Generation of Identification Models of Robots," *International Journal of Robotics and Automation*, 1 no. 1, pp. 2-6.

Khatib, O, 1983, "Dynamic control of manipulators in operational space," *Sixth IFTOMM Congress on Theory of Machines and Mechanisms*, New Delhi, Dec 15-20.

Khatib, O. and Burdick, J., 1986, "Motion and force control of robot manipulators," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp.

1381-1386.

Khosla, P.K. and Kanade, T., 1986, "Real-time implementation and evaluation of model-based controls on CMU," *Proc. IEEE Conf. on Robotics and Automation*, San Francisco, April 7-10, pp. 1546-1555.

Kudva, P., and Narendra, K.S., 1974, "The discrete adaptive observer," CT-63, Department of Engineering and Applied Science, Yale University, June.

Kuwahara, H., Ono, Y., Nikaido, M., and Matsumoto, T., 1985, "A precision direct-drive robot arm," *Proc. American Control Conf.*, Boston, June 19-21, pp. 722-727.

Lawson, C.L., and Hanson, R.J., 1974, *Solving Least Squares Problems*, Englewood, N.J., Prentice-Hall.

Lee, K., 1983, Shape Optimization of Assemblies Using Geometric Properties, Ph.D. Thesis, MIT, Mechanical Engineering Department, December.

Liégeois, A., Fournier, A. and Aldon, M., 1980, "Model Reference Control of High-Velocity Industrial Robots," *Proc. Joint Automatic Control Conf.*, San Francisco, CA, August 13-15.

Ljung, L. and Soderstrom, T., 1983, *Theory and Practice of Recursive Identification*, Cambridge, Ma., MIT Press.

Lozano-Perez, T., 1984, *MIT Course Notes on Robot Manipulation (6.867).*

Luh, J.Y.S., Fisher, W.D, and Paul, R.P., 1983, "Joint torque control by a direct feedback for industrial robots," *IEEE Trans. Automatic Control*, AC-28, pp. 153-160.

Luh, J.Y.S., Walker, M., and Paul, R.P., 1980a, "On-line computational scheme for mechanical manipulators," *J. Dynamic Systems, Meas., Control*, 102, pp. 69-76.

Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., 1980b, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. Auto. Contr.*, AC-25, pp. 468-474.

Markiewicz, B., 1973, "Analysis of the Computed Torque Drive Method and Comparison With Conventional Position Servo for a Computer-Controlled Manipulator," Jet Propulsion Laboratory, Pasadena, CA, March 15.

Marquardt, D.W., and Snee, R.D., 1975, "Ridge regression in practice," *Amer. Statistician*, 29, pp. 3-20.

Mathlab Group, 1983, "MACSYMA Reference Manual," Version 10, Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge, MA..

Mayeda, H., Osuka, K., and Kangawa, A., 1984, "A new identification method for serial manipulator arms," *Preprints IFAC 9th World Congress, Vol. VI*, Budapest, July 2-6, pp. 74-79.

Mooring, B.W., 1983, "The effect of joint axis misalignment on robot positioning accuracy," *Proceedings of the 1983 International Computers in Engineering Conf. and Exhibit*, August, pp. 151-155.

Mukerjee, A., 1984, "Adaptation in biological sensory-motor systems: A model for robotic control," *Proc. SPIE Conf. on Intelligent Robots and Computer Vision, Vol. 521*, Cambridge, Ma., November.

Mukerjee, A., and Ballard, D.H., 1985, "Self-calibration in robot manipulators," *Proc. IEEE Conf. Robotics and Automation*, St. Louis, Mar. 25-28, pp. 1050-1057.

Narendra, K.S., and Kudva, P., 1974, "Stable adaptive schemes for system identification and control - Part II," *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-4 no. 6, pp. 552-560.

Narendra, K.S., and Valavani, L.S., 1977, "Stable adaptive observers and controllers," *IEEE Proceedings*, 64, pp. 1198-1208.

Neuman, C.P., and Khosla, P.K., 1985, "Identification of robot dynamics: an application of recursive estimation," *Proc. 4th Yale Workshop on Applications of Adaptive Systems Theory*, New Haven, May 29-31, pp. 42-49.

Olsen, H.B., and Bekey, G.A., 1985, "Identification of parameters in models of robots with rotary joints," *Proc. IEEE Conf. Robotics and Automation*, St. Louis, Mar. 25-28, pp. 1045-1050.

Paul, R. C., Sept. 1972, "Modeling,trajectory calculation and servoing of a computer controlled arm," AIM-177, Stanford University Artificial Intelligence Laboratory.

Paul, R.P., 1981, *Robot Manipulators: Mathematics, Programming, and Control*, Cambridge, Mass., MIT Press.

Raibert, M.H., and Craig, J.J, 1981, "Hybrid position/force control of manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, 102, pp. 126-133.

Roberge, J.K, 1975, *Operational Amplifiers: Theory and Practice*, New York, John Wiley and Sons, Inc..

Roberts, R.K, 1984, The Compliance of End Effector Force Sensors for Robot Manipulator Control, Ph.D. Thesis, Purdue Univ., School of Electrical Engineering, Dec..

Roberts, R.K., Paul, R.P., and Hillberry, B.M., 1985, "The effect of wrist force sensor stiffness on the control of robot manipulators," *Proc. IEEE Conf. Robotics and Automation*, St. Louis, Mar. 25-28, pp. 269-274.

Salisbury, J.K., 1980, "Active stiffness control of a manipulator in Cartesian coordinates," *Proc. 19th IEEE CDC*, Dec..

Salisbury, J.K., 1982, Kinematic and Force Analysis of Articulated Hands, Ph.D. Thesis, Stanford Univ., Dept. Mechanical Eng., May.

Samson, C., 1983, "Robust nonlinear control of robotic manipulators," *Proc. 22nd IEEE Conf. Decision and Control*, San Antonio, Dec. 14-16.

Shih, S., 1985a, Reduced-order model-reference adaptive system identification of large scale systems with discrete adaptation laws, Ph.D. Thesis, MIT, Department of Electrical Engineering and Computer Science, March.

Shih, S., Lang, J. H., 1985b, "Adaptive observers in which parameters are updated infrequently," *Proc. 24th Conf. Decision and Control*, Fort Lauderdale, Florida, Dec. 11-13.

Siegel, D.M., Garabieta, I., and Hollerbach, J.M., 1986, "An integrated tactile and thermal sensor," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, April 7-10, pp. 1286-1291.

Slotine, J.-J. E., 1985, "The robust control of robot manipulators," *Int. J. Robotics Research*, 4 no. 2, pp. 49-64.

Spong, M.W., Thorp, J.S., and Kleinwaks, J.M., 1984, "The control of robot manipulators with bounded input. Part II: robustness and disturbance rejection," *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, Dec. 12-14, pp. 1047-1052.

Sugimoto, K., and Okada, T., 1985, "Compensation of positioning errors caused by geometric deviations in robot system," *Robotics Research: The Second International Symposium*, edited by H. Hanafusa and H. Inoue, Cambridge, Mass., MIT Press, pp. 231-236.

Symon, K.R., 1971, *Mechanics*, Reading, Mass., Addison-Wesley.

Whitney, D.E., 1972, "The mathematics of coordinated control of prosthetic arms and manipulators," *ASME J. Dynamic Systems, Meas., Control*, pp. 303-309.

Whitney, D. E., 1977, "Force feedback control of manipulator fine motions," *ASME J. Dyn. Sys.*, pp. 91-97.

Whitney, D.E., Lozinski, C.A., and Rourke, J.M., 1986, "Industrial robot forward calibration method and results," *J. Dynamic Systems, Meas., Control*, 108, pp. 1-8.

Whitney, D. E., 1985, "Historical perspective and state of the art in robot force control," *Proc. IEEE Conf. Robotics and Automation*, St. Louis, Mar. 25-28, pp. 262-268.

Wlassich, J.J., 1986, Nonlinear Force Feedback Impedance Control, MS Thesis, MIT, Dept. of Mechanical Engineering, Feb..

Wu, C.H., 1983, "The kinematic error model for the design of robot manipulators," *Proc. American Control Conf.*, San Francisco, June 22-24, pp. 497-502.

Wu, C.H. and Paul, R.P., 1980, "Manipulator compliance based on joint torque control," *Proc. IEEE Conf. Decision and Control*, Alburquerque, NM, Dec. 10-12, pp. 88-94.

Youcef-Toumi, K., 1985, Analysis, Design, and Control of Direct-Drive Manipulators, Ph.D. Thesis, MIT, Dept. of Mechanical Engineering, May.

CS-TR Scanning Project
Document Control Form

Date : 10/18/95

Report # AI-TR-912

Each of the following should be identified by a checkmark:
Originating Department:

- [x] Artificial Intellegence Laboratory (AI)
- [ ] Laboratory for Computer Science (LCS)

Document Type:

- [x] Technical Report (TR)
- [ ] Technical Memo (TM)
- [ ] Other:

Document Information

Number of pages: 160 (167 - IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:

- [x] Single-sided or
- [ ] Double-sided

Intended to be printed as :

- [ ] Single-sided or
- [x] Double-sided

Print type:

- [ ] Typewriter
- [ ] Offset Press
- [x] Laser Print
- [ ] InkJet Printer
- [ ] Unknown
- [ ] Other:

Check each if included with document:

- [x] DOD Form (2)
- [ ] Funding Agent Form
- [x] Cover Page
- [ ] Spine
- [ ] Printers Notes
- [ ] Photo negatives
- [ ] Other:

Page Data:

Blank Pages (by page number):

Photographs/Tonal Material (by page number): 13, 27, 100

Other (note description/page number):

Description : Page Number:

IMAGE MAP: (1-160) UN#'ED TITLE PAGE, 2-160
(161-167) SCANCONTROL, COVER, DOD (2)
TRGT'S (3)

(B) CUT&PASTE FIGS ON PAGES 13, 21, 27-28, 31, 34, 38, 45, 49, 54, 59, 62-63, 65, 69, 71-77, 79, 82-84, 90, 92-93, 95-96, 98-102, 105, 107, 110-112, 115-121, 123, 124, 126-129, 131-132, 134, 137-142, 146, 150-151. (66-TOTAL)

Scanning Agent Signoff:

Date Received: 10/18/95 Date Scanned: 10/23/95 Date Returned: 10/26/95

Scanning Agent Signature: Michael N. Cook

Rev 9/94 DS/LCS Document Control Form cstrform.vsd

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI-TR-912	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Trajectory and Force Control of a Direct Drive Arm		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Chae H. An		8. CONTRACT OR GRANT NUMBER(s) N00014-80-C-050 N00014-85-K-0124 N00014-82-K-0334
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE Sept. 1986 13. NUMBER OF PAGES 160
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Distribution is unlimited.

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Force Control, Direct Drive Arm

Trajectory Control, Link Estimation

Load Estimation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see back

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract. Using the MIT Serial Link Direct Drive Arm as the main experimental device, various issues in trajectory and force control of manipulators were studied in this thesis. Since accurate modelling is important for any controller, issues of estimating the dynamic model of a manipulator and its load were addressed first. Practical and effective algorithms were developed from the Newton-Euler equations to estimate the inertial parameters of manipulator rigid-body loads and links. Load estimation was implemented both on a PUMA 600 robot and on the MIT Serial Link Direct Drive Arm. With the link estimation algorithm, the inertial parameters of the direct drive arm were obtained. For both load and link estimation results, the estimated parameters are good models of the actual system for control purposes since torques and forces can be predicted accurately from these estimated parameters.

The estimated model of the direct drive arm was then used to evaluate trajectory following performance by feedforward and computed torque control algorithms. The experimental evaluations showed that the dynamic compensation can greatly improve trajectory following accuracy.

Various stability issues of force control were studied next. It was determined that there are two types of instability in force control. Dynamic instability, present in all of the previous force control algorithms discussed in this thesis, is caused by the interaction of a manipulator with a stiff environment. Kinematic instability is present only in the hybrid control algorithm of Raibert and Craig, and is caused by the interaction of the inertia matrix with the Jacobian inverse coordinate transformation in the feedback path. Several methods were suggested and demonstrated experimentally to solve these stability problems. The results of the stability analyses were then incorporated in implementing a stable force/position controller on the direct drive arm by the modified resolved acceleration method using both joint torque and wrist force sensor feedbacks.

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

Scanned

Date: 10/23/1995

M.I.T. Libraries
Document Services