**PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT**

PRECISIONSM RESOLUTION TARGETS

PIONEERS IN ME* NYLON BLUE TESTING SINCE 1974

10300 COUNTY ROAD 5 SUDBURY ON N6A 2G7 CAN
TEL 512 435 7887 FAX 512 435 7887 TLX 510000-000

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Investigations of Impedance Control
for a
Robot Manipulator

by

WILLIAM C. McCORMICK, B.Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Engineering in Electrical Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Ontario
June 14, 1991

©Copyright
1991. William C. McCormick

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-75984-4

Canada

The undersigned recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis
Investigations of Impedance Control for a Robot Manipulator
submitted by
William C. McCormick
in partial fulfilment of the requirements for
the degree of Master of Engineering in Electrical Engineering

A handwritten signature in black ink, appearing to read 'Hans K. ...', written over a horizontal line.

Thesis Supervisor

A handwritten signature in black ink, appearing to read 'Seetharam', written over a horizontal line.

Chairman, Department of Systems and Computer Engineering

Carleton University

June 14, 1991

Abstract

This thesis documents investigations into impedance control carried out on a direct drive robot manipulator.

Two impedance controllers are simulated numerically and implemented on Carleton University's direct drive robot. The performance of these controllers is compared.

The stability of the robot in contact with a stiff and a compliant environment is investigated. Both impedance controllers are used for this investigation. The experimental results are compared to three contact stability theories.

Acknowledgements

This thesis is the culmination of research that I found very educational and enlightening. I would like to thank my thesis supervisor, Professor H. M. Schwartz for his guidance and support throughout.

I would also like to thank my fellow students and colleagues at Carleton University, especially Dan Nahmiash, Terry —, Gabriel Warshaw and Vic Aitken, who helped with many of the hard points.

I gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council and the Canadian Space Agency which made this research possible.

Contents

1	Introduction	1
1.1	Scope	1
1.2	General Background	2
1.2.1	Environmental Constraints	2
1.2.2	Hybrid Position/Force Control	2
1.2.3	Impedance Control	3
1.2.4	Contact Stability	4
1.3	Thesis Overview	5
2	Impedance Control Theory	6
2.1	Robot Dynamics Modelling	6
2.1.1	Joint and Workspace Coordinate Systems	7
2.2	A Linear Impedance Controller	9
2.2.1	Target Dynamics	9
2.2.2	Robot Model	12
2.2.3	Controller Design	13
2.2.4	State Feedback Design	15
2.2.5	Force Feedback Gains Selection	18
2.2.6	The Control Law	19
2.2.7	Modifications to the Linear Controller	19

2.3	A Non-Linear Impedance Controller	20
2.3.1	Robot Model	20
2.3.2	Controller Design	21
2.3.3	Modifications to the Non-Linear Controller Design	22
2.3.4	Design Example	23
2.4	Contact Dynamics and Contact Stability	26
2.4.1	Contact Stability for the Linear Impedance Controller	26
2.4.2	Contact Stability for the Non-Linear Impedance Controller	28
2.4.3	A Non-Linear Stability Proof	29
3	Simulations	33
3.1	Simulation Goals	33
3.2	Robot Simulator	34
3.2.1	Robot Model	34
3.2.2	Simulation Technique	36
3.3	Controller Designs	37
3.3.1	The Linear Impedance Controller	37
3.3.2	The Non-Linear Impedance Controller	39
3.4	Simulation Results	41
3.4.1	Verification of Target Dynamics - the Linear Controller	42
3.4.2	Verification of Target Dynamics - the Non-Linear Controller.	47
3.4.3	Joint Space versus Work Space	49
3.4.4	Summary of Simulation Results	52
4	Experimental Results	55
4.1	Experimental Setup	56
4.1.1	The Direct Drive Robot	56
4.1.2	Force Sensor	56
4.1.3	Robot Properties Modelling	57

4.1.4	Environmental Contact Setup	60
4.2	Verification of Target Dynamics	60
4.2.1	Linear Controller	61
4.2.2	Non-Linear Controller	63
4.3	Comparison of Controller Performance	65
4.4	Force Trajectory Limitations	70
4.5	Contact Dynamics Experiments	76
4.5.1	Unstable Contact	78
4.5.2	Stable Contact	83
5	Conclusions and Discussions	85
5.1	Conclusions	85
5.1.1	Controller Comparisons	85
5.1.2	Force Trajectory Limitations	86
5.1.3	Contact Stability Theory	87
5.2	Future Research Directions	88
5.2.1	Force Trajectory Limitations	88
5.2.2	Contact Dynamics Characterization	89
A	Small Gains Theorem	94
B	Force Sensor Integration and Installation	97
B.1	Physical	97
B.2	Electrical	98
B.3	Calibration	98
B.3.1	Motor Torque Constants	105
B.3.2	Sample Computer Source Code for Motor Torque Constant Data Collection	105

B.3.3	Sample Computer Source Code for Motor Torque Constant Estimation	114
B.3.4	Robot Mass Properties	115
B.3.5	Sample Computer Source Code for Robot Mass Properties Data Collection	117
B.3.6	Sample Computer Source Code for Robot Mass Properties Es- timation	128

List of Tables

3.1	Summary of Simulation Parameters	43
3.2	Summary of Simulation Results	53
4.1	Summary of Target Dynamics Verification Results	62
4.2	Contact Stability Results – Stiff Environment	79
4.3	Contact Stability Results – Compliant Environment	80

List of Figures

2.1	Linear Impedance Controller Block Diagram	14
2.2	Linear One Degree of Freedom Manipulator	23
2.3	Block Diagram of Controller Design	25
2.4	Linear Model of Manipulator/Environmental Interactions.	28
2.5	Interaction of the Robot Manipulator with the Environment	29
2.6	Simplified Closed Loop System	30
3.1	Serial Link Manipulator	34
3.2	Simulation Response of Non-Linear Controller to Critically Damped Target Dynamics - Small Initial Deviation.	44
3.3	Simulation Response of Linear Controller to Critically Damped Tar- get Dynamics - Large Initial Deviation.	44
3.4	Simulation Response of Linear Controller to Underdamped Target Dynamics.	45
3.5	Simulation Response of Linear Controller to Small Applied Force. . .	46
3.6	Simulation Response of Linear Controller to Large Applied Force. . .	46
3.7	Simulation Response of Non-Linear Controller to Critically Damped Target Dynamics - Small Initial Deviation.	47
3.8	Simulation Response of Non-Linear Controller to Critically Damped Target Dynamics - Large Initial Deviation.	48

3.9	Simulation Response of Non-Linear Controller to Underdamped Target Dynamics.	48
3.10	Simulation Response of the Non-Linear Controller to a Small Applied Force.	49
3.11	Simulation Response of the Non-Linear Controller to a Large Applied Force.	50
3.12	Controller Behaviour in the Neighbourhood of a Singularity	51
4.1	Carleton's Direct Drive Robot	57
4.2	Universal Force Sensor Mounting Assembly	58
4.3	Compliant Contact Experimental Setup	61
4.4	Linear Controller: Manipulator Response to Small Initial Displacement	63
4.5	Linear Controller: Manipulator Response to Large Initial Displacement	64
4.6	Non-Linear Controller: Manipulator Response to Small Initial Displacement	64
4.7	Non-Linear Controller: Manipulator Response to Large Initial Displacement	65
4.8	Manipulator Position Trajectory for Controller Comparison	66
4.9	Force Response Comparison	68
4.10	Force Response Comparison - Actuator Output	69
4.11	Contact Maintenance Experiment	73
4.12	Contact Maintenance Experiment, Motor Torques	74
4.13	Contact Force Along the Surface	75
4.14	Robot Trajectory for Initial Contact Maintenance Experiment	75
4.15	Unstable Contact Example	81
4.16	Stable Contact Example	84

A.1	Input Output Stability	95
B.1	UFS Mounting	99
B.2	UFS Mounting Plate.	100
B.3	UFS Adaptor Plate.	101
B.4	UFS End Effector.	102
B.5	UFS Mounting Hole Detail.	103
B.6	UFS Electrical Connections	104
B.7	Robot Configuration for Torque Constant Estimation	106
B.8	Motor 1 Torque Constant	107
B.9	Five Bar Linkage Robot Model	116

Chapter 1

Introduction

1.1 Scope

Robot manipulation may be categorized into two classes; the first involves unconstrained motion in space, while the second class involves motion that is constrained in some sense by contact with the environment external to the robot manipulator [15]. The first class of manipulation has been subject to extensive research and application over the past two decades [28]. The second class, while equally important, has only been addressed by a relatively few researchers. The number of applications and amount of experimental work carried out in the second class of manipulation has been correspondingly smaller.

This thesis deals with that second class of robot manipulation where the robot is constrained by some environmental contact. The range of available research topics is wide and varied within this area, including subjects ranging from adaptive control of constrained motion through to analysis of the contact interactions of the manipulator in contact with some environment. The work presented herein is concentrated on practical examples, as opposed to theoretical expositions. This emphasis was selected as the dynamics of such environmental contact are poorly understood [8] and, as such, experimental research would seem a promising approach to increasing

this understanding.

1.2 General Background

1.2.1 Environmental Constraints

Consider first the work of Mason [21]. In this paper, Mason made the observation that for a general n -degree of freedom manipulator, each degree of freedom, in the appropriate frame of reference, will be either force¹ constrained or position constrained. Two examples are illustrative of this observation:

1. A manipulator operating in space, with no environmental contact, has the contact force constrained to zero in all degrees of freedom.
2. Consider a two degree of freedom manipulator pushing against a frictionless wall. In the normal direction to the wall, the position is constrained, while in the tangential direction along the wall, the force is constrained to zero.

The reader may note that Mason has neglected some practical considerations. For example, no allowance is made for the possibility of the environment deforming. Such deformation would result in a constraint in the relationship between the force exerted on the manipulator and the position of the manipulator. Another example makes use of item 2 above: the position of the manipulator is only constrained in that it cannot move past the wall; the manipulator is completely free to move *away* from the wall.

1.2.2 Hybrid Position/Force Control

The next important paper investigating control of a robot manipulator in contact with some environment was written by Mark Raibert and J.J. Craig in 1981 [26]. In this paper, the authors proposed a control scheme which they termed *Hybrid Position/Force Control*. This control scheme may have been based upon Mason's work

¹Throughout the body of this thesis, the term force is used to be representative of both force and torque.

described above; the robot's workspace is divided into orthogonal directions based on the environmental constraints. In those directions that are position constrained, the contact force is controlled, and in those directions that are force constrained, the manipulator position is controlled.

After reviewing [26], it becomes apparent that there are some issues that have not been addressed:

1. What are the methods for choosing constraints for a given assembly operation?

It is necessary to have a good working knowledge of the environment in order to use the appropriate type of control for each degree of freedom [31].

2. What are the limitations imposed on the controller in order to maintain stable contact with the environment? When the manipulator comes into contact with the environment, it will exhibit different dynamic characteristics [20]. Plainly it is desirable to guarantee stability under all conditions of operation.

3. When the environmental constraints change, it is necessary to design a new controller with a different structure that incorporates the new constraints [16].

4. How will frictional forces and environmental deformation affect the control strategy? Consider the ideal positioning control; it would be infinitely stiff, rejecting any position disturbance. Suppose that this positioning control were to encounter some object in the robot's trajectory; very large contact forces would result from the collision, potentially damaging the manipulator and/or the object [16]. Yet such an ideal positioning controller would exhibit the same properties as Raibert and Craig's controller in force constrained directions.

These difficulties may have led to control strategies that specified a relationship between the contact force and robot position. The most general of these strategies has been termed *impedance control*.

1.2.3 Impedance Control

Impedance control was first proposed by Neville Hogan [12]. The fundamental idea underlying impedance control is that environmental constraints impose a relation-

ship between the position of a manipulator and the contact force exerted by the environment. Hogan points out that this relationship may be modelled by a generalized, non-linear impedance, consisting of some inertial, damping and stiffness characteristics that describe the relationship. Consequently, Hogan hypothesizes that a viable control strategy for contact tasks would be to give the manipulator a complementary impedance to that of the environment. The combined manipulator-environment system should then be well behaved, much like a matched impedance electrical circuit.

The term *impedance control* typically refers to a control law that implements some target dynamics consisting of selected inertial, damping and stiffness parameters. For example, Kazerooni [15, 16] proposes a control law that implements the target dynamics of a generalized n -dimensional damped linear spring on an n -degree of freedom manipulator. Colgate and Hogan [7] propose a control methodology for implementing a generalized single degree of freedom impedance. Anderson and Spong [2] propose an inverse dynamics controller that implements a second order linear target dynamics.

Within this thesis, the controllers proposed by Kazerooni [15, 16] and Colgate and Hogan [7] were compared through numerical simulations and experiments on Carleton's direct drive robot.

1.2.4 Contact Stability

The term *contact stability* refers to stability of the robot manipulator when it is in contact with some environment. *Contact dynamics* refers to mathematical modelling of the interactions between the manipulator and the environment.

Contact stability is plainly a critical issue for control of a manipulator in contact with some environment. Early analyses of contact dynamics treated the environment as a first or second order linear system to which the manipulator was directly coupled (see, for example, [16] or [7]). At the same time, it was well known that environmental contact could not always be so simply modelled; for example, robot *chattering* when in contact with some surface has been frequently observed [1].

More recent contact dynamics analyses use input-output stability methods to obtain sufficiency conditions for contact stability [19].

Four series of contact dynamics experiments were carried out as part of the experimental work described in this thesis. Three contact stability theories were subject to systematic experimental investigation.

1.3 Thesis Overview

Chapter 2 of this document presents an overview of two methods of designing impedance controllers. Contact stability analyses are presented and discussed.

Chapter 3 contains the simulations that were carried out in order to investigate the two impedance control techniques for a 2 degree of freedom manipulator.

Chapter 4 describes experimental results for the two impedance controllers obtained on Carleton University's direct drive robot. The two impedance control techniques are compared. Additionally, systematic investigations into contact stability were carried out.

A summary of the results obtained in Chapters 3 and 4 are presented in Chapter 5. Conclusions which may be drawn from this research are stated and discussed. Promising future research directions are outlined.

Following the list of references, there are two Appendices containing:

- Theoretical details of the small gains theorem discussed in Chapter 2; and
- Details of the integration of the force sensor selected for this research into the robot system.

Chapter 2

Impedance Control Theory

2.1 Robot Dynamics Modelling

Most of the theoretical work in the body of this thesis is based on the well known model of robot dynamics [28] given by

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + F_g(\theta) = D \quad (2.1)$$

where

$M(\theta)$ defines an $n \times n$ matrix that describes the inertial properties of the robot manipulator. $M(\theta)$ is often referred to as the mass matrix. The mass matrix has the properties of being symmetrical and positive definite.

$C(\theta, \dot{\theta})$ defines an $n \times 1$ vector that describes coriolis and centripetal acceleration terms in the dynamics of the robot manipulator. Non-zero elements in C contain a product of velocities, that is, non-zero elements are of the form $k\theta_i\theta_j$, where k is some function of θ .

$F_g(\theta)$ is an $n \times 1$ vector that specifies the effects of gravitational forces acting on the manipulator.

D is the $n \times 1$ vector that defines generalized joint torques that act on the manipulator. For the purposes of this document, D is subdivided into two further components:

$$D = T + J_c^T(\theta)F. \quad (2.2)$$

Here T represents the actuator torques, F the force applied to the end effector of the manipulator, and $J_c^T F$ represents F reflected to the manipulator joints assuming rigid manipulator links. (A more detailed discussion of the reflection of this force to the manipulator joints follows shortly.)

2.1.1 Joint and Workspace Coordinate Systems

One frame of reference often used in robotic systems is the joint angle coordinate system. In this frame of reference, the position of the manipulator is specified by the joint angle measurements. Within this thesis, joint angle measurements are always denoted by θ , for a vector of all the joint angles, or θ_i , for a particular joint angle.

When dealing with environmental interactions, it is desirable to work in a frame of reference that corresponds more closely to the environment. This frame of reference is often referred to as workspace or Cartesian coordinates. (Cartesian coordinates are one specific case of the more general coordinate sets that may be appropriate to describe the robot's workspace.) Within this report, the term workspace or Cartesian coordinates are used to denote the Euclidean coordinate system shown in Figures 3.1 and 4.1. The symbol Y is used to represent a vector of Cartesian coordinates and y_i is used to denote one particular Cartesian coordinate.

In general, each set of joint angles will map to only one set of workspace coordinates. Denote this functional mapping by:

$$Y = G(\theta) \quad (2.3)$$

Workspace velocities may then be determined through the use of the *Jacobian*, or matrix derivative operator:

$$\dot{Y} = J_c(\theta)\dot{\theta} \quad (2.4)$$

where J_c is the Jacobian, defined as:

$$J_c = \begin{pmatrix} \frac{\partial y_1}{\partial \theta_1} & \frac{\partial y_1}{\partial \theta_2} & \cdots & \frac{\partial y_1}{\partial \theta_n} \\ \frac{\partial y_2}{\partial \theta_1} & \frac{\partial y_2}{\partial \theta_2} & \cdots & \frac{\partial y_2}{\partial \theta_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial y_n}{\partial \theta_1} & \frac{\partial y_n}{\partial \theta_2} & \cdots & \frac{\partial y_n}{\partial \theta_n} \end{pmatrix} \quad (2.5)$$

In a similar fashion, acceleration in workspace coordinates can be related to joint angle accelerations as

$$\ddot{Y} = J_c(\theta)\ddot{\theta} + \dot{\theta}\frac{dJ_c(\theta)}{dt} \quad (2.6)$$

$$= J_c(\theta)\ddot{\theta} + \dot{\theta}\dot{\theta}^T \frac{\partial J_c(\theta)}{\partial \theta} \quad (2.7)$$

Singularities

Observe that the transformation from joint rates to workspace velocities always exists. The converse is not true, however. When the Jacobian is singular (i.e. non-invertible), it is impossible to compute joint velocities that will give any arbitrary workspace velocities.

Such a location within the robot workspace is known as a *singularity of the mechanism* or *singularity* for short [9]. When a robot reaches a singularity, it loses one or more degrees of freedom. One example that applies to all manipulator configurations occurs when the robot is at the boundary of its workspace.

End-effector Forces and Joint Torques

End effector forces may also be reflected to joint torques through the manipulator Jacobian. This relationship may be derived using a method often called virtual work; a force causing a small displacement at the end effector must be equivalent to some torque causing a small displacement at the manipulator joints:

$$F \cdot \delta Y = T \cdot \delta \theta, \quad (2.8)$$

where $\cdot$ denotes the dot product operator. Equation 2.8 may also be written in vector multiplication form:

$$F^T \delta Y = T^T \delta \theta. \quad (2.9)$$

Noting that the Jacobian may be defined by $\delta Y = J_c(\theta)\delta\theta$ it is straightforward to obtain

$$F^T J_c \delta \theta = T^T \delta \theta \quad (2.10)$$

$$F^T J_c = T^T \quad (2.11)$$

$$T = J_c^T(\theta)F. \quad (2.12)$$

2.2 A Linear Impedance Controller

In [16] Kazerooni proposed a linear impedance controller for robotic manipulators that operates using full state feedback as well as a force feedback term. Kazerooni's method also accounts for actuator dynamics.

2.2.1 Target Dynamics

Kazerooni selects the target dynamics for the controller to be those of a generalized n dimensional second order linear system:

$$\delta F(t) = J\delta\ddot{Y}(t) + C\delta\dot{Y}(t) + K\delta Y(t) \quad (2.13)$$

where the variables are defined as:

$\delta F(t)$ represents a deviation from some force set point,

J is an $n \times n$ matrix describing the inertial parameters of the target dynamics,

C is an $n \times n$ matrix describing the damping parameters of the target dynamics,

K is an $n \times n$ matrix describing the stiffness parameters of the target dynamics,

$\delta Y(t)$ represents a deviation from some position set point, and

n is the number of degrees of freedom of the manipulator.

Observe that the target dynamics for the impedance controller have been parameterized in terms of three matrices specifying the desired inertia, damping and stiffness.

The stiffness matrix specifies the desired steady state reaction of the closed loop system to some applied force; a force perturbation will then result in some well defined deviation in position from the target dynamics. The inertia matrix specifies the controller bandwidth. The damping matrix specifies the speed of the desired response. Typically the damping matrix is chosen such as to guarantee critical damping, giving the fastest non-oscillatory response within the controller bandwidth.

It is important to note that the target dynamics matrices will often be chosen as diagonal; off-diagonal elements lead to cross coupling between workspace coordinates.

Kazerooni's design method is based on duplicating the eigenstructure of the target dynamics in the closed loop robot system; in order to carry out this duplication, it is necessary to derive the target dynamics' eigenstructure.

In state space form, the target dynamics may be written as:

$$\begin{aligned} \begin{bmatrix} \delta\dot{Y}(t) \\ \delta\ddot{Y}(t) \end{bmatrix} &= \begin{bmatrix} 0_{nn} & I_{nn} \\ -J^{-1}K & -J^{-1}C \end{bmatrix} \begin{bmatrix} \delta Y(t) \\ \delta\dot{Y}(t) \end{bmatrix} + \begin{bmatrix} 0_{nn} \\ J^{-1} \end{bmatrix} \delta F(t) \\ &= A_t \begin{bmatrix} \delta Y(t) \\ \delta\dot{Y}(t) \end{bmatrix} + B_t \delta F(t) \end{aligned} \quad (2.14)$$

$$\begin{aligned} \delta Y(t) &= \begin{bmatrix} I_{nn} & 0_{nn} \end{bmatrix} \begin{bmatrix} \delta Y(t) \\ \delta\dot{Y}(t) \end{bmatrix} \\ &= C_t \begin{bmatrix} \delta Y(t) \\ \delta\dot{Y}(t) \end{bmatrix} \end{aligned} \quad (2.15)$$

From here it follows directly that the transfer function of the target dynamics (considering the force vector as an input) is given by

$$G_t(s) = (Js^2 + Cs + K)^{-1} = C_t(sI_{2n2n} - A_t)^{-1}B_t \quad (2.16)$$

The eigenvalues (λ_i) and corresponding eigenvectors (z_i) of the target dynamics are then defined by

$$(\lambda_i I_{2n2n} - A_t)z_i = 0_{2n}, z_i \neq 0_{2n}, i = 1, 2, \dots, 2n \quad (2.17)$$

Substituting A_t from 2.14 into 2.17 gives that

$$\begin{bmatrix} \lambda_i I_{nn} & -I_{nn} \\ J^{-1}K & \lambda_i I_{nn} + J^{-1}C \end{bmatrix} z_i = 0_{2n} \quad (2.18)$$

Let the eigenvectors be denoted by

$$z_i = \begin{bmatrix} q_i \\ p_i \end{bmatrix} \quad (2.19)$$

where q_i and p_i are $1 \times n$ vectors. Substituting this representation of z_i into equation 2.18 gives two equations:

$$\lambda_i I_{nn} q_i - I_{nn} p_i = 0_n, q_i \neq 0, p_i \neq 0 \quad (2.20)$$

and

$$J^{-1} K q_i + (\lambda_i I_{nn} + J^{-1} C) p_i = 0_n, q_i \neq 0, p_i \neq 0 \quad (2.21)$$

From equation 2.20, it is apparent that

$$p_i = \lambda_i q_i. \quad (2.22)$$

Substituting this result into equation 2.21 gives that

$$(J^{-1} K + \lambda_i^2 I_{nn} + \lambda_i J^{-1} C) q_i = 0_n, q_i \neq 0_n \quad (2.23)$$

Multiplying through by J , we have

$$(\lambda_i^2 J + \lambda_i C + K) q_i = 0. \quad (2.24)$$

To summarize, the eigenstructure of the target dynamics, defined by A_t , is given by eigenvectors z_i and eigenvalues λ_i where

$$z_i = \begin{bmatrix} q_i \\ \lambda_i q_i \end{bmatrix} \quad (2.25)$$

and q_i is defined by

$$(J \lambda_i^2 + C \lambda_i + K) q_i = 0_n, q_i \neq 0_n, i = 1, 2, \dots, 2n. \quad (2.26)$$

As explained in section 2.2.2, for small deviations, the transformation from workspace to joint space coordinates is defined by

$$\delta Y(t) = J_c \delta \theta(t) \quad (2.27)$$

$$\delta \dot{Y}(t) = J_c \delta \dot{\theta}(t). \quad (2.28)$$

Using these substitutions in equation 2.25, the eigenvectors of the target dynamics in the joint angle frame are given by

$$v_i = \begin{bmatrix} J_c^{-1} q_i \\ \lambda_i J_c^{-1} q_i \end{bmatrix} \quad (2.29)$$

provided that the Jacobian, J_c is invertible (that is, the robot is not at a singularity).

Finally, let

$$\begin{aligned} V &= \begin{bmatrix} v_1 & v_2 & \cdots & v_{2n} \end{bmatrix} \\ \Lambda &= \{\lambda_i, i = 1, 2, \dots, 2n\} \end{aligned} \quad (2.30)$$

Then V and Λ define the eigenstructure of the target dynamics in the joint angle coordinate frame.

2.2.2 Robot Model

Recall from Section 2.1 that the dynamics of the manipulator are defined by

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + F_g(\theta) = D. \quad (2.31)$$

Assuming slow manipulator motion, the velocity product terms that compose $C(\theta, \dot{\theta})$ will be negligible, so we can set $C(\theta, \dot{\theta}) = 0$. Additionally, as the target dynamics are predicated on small deviations of the manipulator from some set point, the robot dynamics may be linearized about this set point, giving

$$M(\theta_0)\delta\ddot{\theta} + G_r(\theta_0)\delta\theta = \delta D \quad (2.32)$$

where $G_r(\theta_0)$ is the $n \times n$ matrix defined by

$$G_r(\theta_0) = \left[\frac{\partial F_g(\theta)}{\partial \theta_1} \frac{\partial F_g(\theta)}{\partial \theta_2} \cdots \frac{\partial F_g(\theta)}{\partial \theta_n} \right] \bigg|_{\theta=\theta_0} \quad (2.33)$$

The generalized torque perturbation, δD , can be expressed as the summation of the actuator torque vector and the effect of applied interaction force. Assuming that the joints are independently driven by the actuators (as opposed to having the actuators mechanically cross-coupled to a single joint), δD is given by the expression:

$$\delta D = \delta T + J_c^T \delta F \quad (2.34)$$

where δT represents the actuator torque perturbations from their equilibrium value and δF represents the interaction force perturbations.

The actuator dynamics are modeled by decoupled first order linear systems; the dynamics of actuator i are given by

$$\frac{\delta \dot{\tau}_i(t)}{\lambda_{ai}} + \delta \tau_i(t) = \delta u_i(t), i = 1, 2, \dots, n. \quad (2.35)$$

where δu_i and $\delta \tau_i$ are the input torque command perturbation and the output torque perturbation respectively. In state space form the actuator dynamics may be written as

$$\delta \dot{T}(t) = A_a \delta T(t) + B_a \delta U(t) \quad (2.36)$$

where A_a and B_a are diagonal matrices containing the actuator eigenvalues:

$$A_a = \text{diag}(-\lambda_{a1}, -\lambda_{a2}, \dots, -\lambda_{an}) \quad (2.37)$$

$$B_a = \text{diag}(\lambda_{a1}, \lambda_{a2}, \dots, \lambda_{an}) \quad (2.38)$$

Combining equations 2.32, 2.34, and 2.36 gives the complete dynamics for the linearized robot model and the actuators.

$$\underbrace{\begin{bmatrix} \delta \dot{\theta}(t) \\ \delta \ddot{\theta}(t) \\ \delta \dot{T}(t) \end{bmatrix}}_{\delta \dot{X}(t)} = \underbrace{\begin{bmatrix} 0_{nn} & I_{nn} & 0_{nn} \\ -M^{-1}(\theta_0)G_r(\theta_0) & 0_{nn} & M^{-1}(\theta_0) \\ 0_{nn} & 0_{nn} & A_a \end{bmatrix}}_A \underbrace{\begin{bmatrix} \delta \theta(t) \\ \delta \dot{\theta}(t) \\ \delta T(t) \end{bmatrix}}_{\delta X(t)} \quad (2.39)$$

$$+ \underbrace{\begin{bmatrix} 0_{nn} \\ 0_{nn} \\ B_a \end{bmatrix}}_B \delta U(t) + \underbrace{\begin{bmatrix} 0_{nn} \\ M^{-1}(\theta_0)J_c^T \\ 0_{nn} \end{bmatrix}}_L \delta F(t) \quad (2.40)$$

Define $H = \begin{bmatrix} I_{nn} & 0_{nn} & 0_{nn} \end{bmatrix}$ and the state space equations for the robot are given by

$$\delta \dot{X}(t) = A \delta X(t) + B \delta U(t) + L \delta F(t) \quad (2.41)$$

$$\delta \theta(t) = H \delta X(t) \quad (2.42)$$

2.2.3 Controller Design

A block diagram of the proposed controller is shown in Figure 2.1.

Figure 2.1: Linear Impedance Controller Block Diagram

In state space form, the closed loop dynamics are given by

$$\delta \dot{X}(t) = (A - BG)\delta X(t) + (L + BG_d)\delta F(t) \quad (2.43)$$

Observe that this controller uses both full state feedback and force feedback. The controller design is separated into the design of the state feedback gains and the force feedback gains. Recall that the intent of the compensator is to cause the closed loop system to approach the behaviour of the target dynamics. This behaviour is achieved by first dividing the closed loop eigenstructure into *dominant* and *non-dominant* partitions¹. The feedback gains are then chosen so that the eigenstructure of the target dynamics is then duplicated in the dominant partition of the closed loop robot system. As a result of the particular form of the controller, it is also possible to independently assign the location of the eigenvalues of the non-dominant partition.

¹The dominant partition consists of the $2n$ eigenvalues (and their associated eigenvectors) that are closest to the origin in the left half complex plane. The non-dominant partition consists of the n eigenvalues (and their associated eigenvectors) that are furthest into the left half plane. This partition is termed non-dominant because it has less effect on the overall closed loop dynamics than the dominant partition.

The next two sections discuss in detail the method used to select the feedback gains.

2.2.4 State Feedback Design

The state feedback gains matrix, G , is designed to obtain the eigenstructure represented by V and Λ . The closed loop robot eigenstructure is given by

$$s_i u_i = (A - BG)u_i, u_i \neq 0_{3n}, i = 1, 2, \dots, 3n \quad (2.44)$$

where s_i is the i th eigenvalue and u_i is the i th $3n \times 1$ eigenvector. For convenience define

$$U = \begin{bmatrix} u_1 & u_2 & \dots & u_{3n} \end{bmatrix} \quad (2.45)$$

$$= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \quad (2.46)$$

$$= \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \quad (2.47)$$

$$S = \{s_i, i = 1, 2, \dots, 3n\} \quad (2.48)$$

The objective is then to select G so that $(A - BG)$ contains the eigenstructure represented by Λ and V . This objective is carried out by selecting G so that

1. The $2n$ most dominant eigenvalues of S correspond to Λ . The remaining n eigenvalues are associated with the actuator dynamics and selected to be much further into the left half plane.
2. The dominant partition of U contains V . Some explanation is required for the term *dominant partition*. In the above paragraph, we divided U into two smaller matrices; U_1 and U_2 . In this case, U_1 is chosen to be the set of eigenvectors associated with the dominant set of eigenvalues (Λ), and, as such, we refer to U_1 as the dominant partition of U . In a complementary fashion, we could refer to U_2 as the non-dominant partition of U .

We describe first how the dominant partition eigenstructure may be constructed. Kazerooni undertakes a detailed derivation in [16], however, here the method is stated without rigor.

In equation 2.47, the matrix U was partitioned into four sub-matrices; consider the following observations about each:

U_{11} is a $2n \times 2n$ matrix that reflects the contribution of the $2n$ dominant eigenvalues to the manipulator states (that is θ and $\dot{\theta}$).

U_{21} is a $n \times 2n$ matrix that reflects the contribution of the $2n$ dominant eigenvalues to the actuator states ($T(t)$).

U_{12} is a $2n \times n$ matrix that shows the contribution of the n non-dominant eigenvalues on the manipulator states.

U_{22} is a $n \times n$ matrix that shows the contribution of the n non-dominant eigenvalues on the actuator states.

As a result of the linearized choice of robot model, U_1 may be constructed so that U_{11} is equal to V as follows.

Rewriting equation 2.44, we have

$$(s_i I_{3n3n} - A)u_i + BG u_i = 0_{3n}, u_i \neq 0_{3n}, i = 1, 2, \dots, 3n. \quad (2.49)$$

Substituting $s_i = \lambda_i$ and defining

$$m_i = -G u_i \quad (2.50)$$

gives

$$\begin{bmatrix} (\lambda_i I_{3n3n} - A) & -B \end{bmatrix} \begin{bmatrix} u_i \\ m_i \end{bmatrix} = 0_{3n}, i = 1, 2, \dots, 3n. \quad (2.51)$$

Substituting for A and B from equation 2.40 into equation 2.51 gives

$$\begin{bmatrix} \lambda_i I_{nn} & -I_{nn} & 0_{nn} & 0_{nn} \\ M^{-1}(\theta_0)G_r(\theta_0) & \lambda_i I_{nn} & -M^{-1}(\theta_0) & 0_{nn} \\ 0_{nn} & 0_{nn} & \lambda_i I_{nn} - A_a & -B_a \end{bmatrix} \begin{bmatrix} u_i \\ m_i \end{bmatrix} = 0_{4n}, u_i \neq 0, m_i \neq 0 \quad (2.52)$$

Recall that, in order to duplicate the target dynamics eigenstructure, it is necessary that the first $2n$ elements of u_i must be equal to v_i , where v_i was defined by equation 2.29 as

$$v_i = \begin{bmatrix} J_c^{-1} q_i \\ \lambda_i J_c^{-1} q_i \end{bmatrix} \quad (2.53)$$

By inspection, it is apparent that the choice of

$$u_i = \begin{bmatrix} I_{nn} \\ \lambda_i I_{nn} \\ M(\theta_0)\lambda_i^2 + G_r(\theta_0) \end{bmatrix} J_c^{-1}(\theta_0)q_i, i = 1, 2, \dots, 2n \quad (2.54)$$

and

$$m_i = B_a^{-1}(\lambda_i I_{nn} - A_a)[M(\theta_0)\lambda_i^2 + G_r(\theta_0)]J_c^{-1}(\theta_0)q_i, i = 1, 2, \dots, 2n. \quad (2.55)$$

is a solution to both equation 2.52 and the restriction that the first $2n$ elements of u_i must be the same as v_i .

Now consider the non-dominant partition of the eigenstructure (that for the actuator dynamics).

For the actuator eigenstructure, the u_i terms must still be chosen to satisfy equation 2.51. Again as a result of the linearized model chosen for the robot manipulator, it is guaranteed that there are at least n independent values that may be selected for each u_i term [16]. This guarantee allows the designer to arbitrarily select the matrix U_{22} . For the actuator dynamics, these values are selected by setting $U_{22} = I_{nn}$, ensuring that the actuators will remain decoupled in the closed loop system. As the actuator eigenvalues are non-dominant, the remaining $2n$ elements in u_i will not be significant².

Substituting the following choices for u_i and m_i into equation 2.51, shows that selecting

$$u_i = \begin{bmatrix} [M(\theta_0)s_i^2 + G_r(\theta_0)]^{-1} \\ [M(\theta_0)s_i^2 + G_r(\theta_0)]^{-1} s_i \\ I_{nn} \end{bmatrix} p_{i-2n}, i = 2n + 1, 2n + 2, \dots, 3n \quad (2.56)$$

and

$$m_i = B_a^{-1}(s_i I_{nn} - A_a)p_{i-2n}, i = 2n + 1, 2n + 2, \dots, 3n. \quad (2.57)$$

will satisfy both requirements for the actuator dynamics. Here p_i has been defined as:

$$P = \begin{bmatrix} p_1 & p_2 & \dots & p_n \end{bmatrix} = I_{nn}. \quad (2.58)$$

²Kazerooni [16] shows that as the actuator eigenvalues, $|s_i| \rightarrow \infty$ the other elements in u_i will approach zero for $i = 2n + 1, 2n + 2, \dots, 3n$.

in order to maintain actuator decoupling in the closed loop system ($U_{22} = I_{nn}$), as discussed above.

Once the m_i and u_i are known, the gains matrix, G , is computed from equation 2.50 as

$$G = - \begin{bmatrix} m_1 & m_2 & \cdots & m_{3n} \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \cdots & u_{3n} \end{bmatrix}^{-1}. \quad (2.59)$$

2.2.5 Force Feedback Gains Selection

Without proof, if the actuator closed loop eigenvalues are selected to satisfy the inequality

$$|s_i| > \rho, \text{real}(s_i) < 0, i = 2n + 1, 2n + 2, \dots, 3n, \quad (2.60)$$

and G is chosen by the design method specified in sub-section 2.2.4 then, as $\rho \rightarrow \infty$, there exists some unique G_d such that

$$\lim_{\rho \rightarrow \infty} J_c G_d(j\omega) = G_t(j\omega) \quad (2.61)$$

within the interval $\omega \in (0, \omega_b)$ where ω_b defines the bandwidth of the robot manipulator for which the linearized model is valid. G_d represents the closed loop transfer function of the system, mapping $\delta F(j\omega)$ to $\delta \theta(j\omega)$. Substituting from equation 2.41:

$$G_d(j\omega) = H(j\omega I_{3n3n} - A + BG)^{-1}(L + BG_d), 0 \leq \omega \leq \omega_b \quad (2.62)$$

G_d may then be computed at any frequency in the interval $(0, \omega_b)$. Choosing $\omega = 0$ provides a simple solution; at $\omega = 0$ equation 2.61 becomes

$$J_c G_d(0) = G_t(0) \quad (2.63)$$

or, substituting for $G_d(0) = H(-A + BG)^{-1}(L + BG_d)$ and $G_t(0) = K^{-1}$:

$$J_c H(-A + BG)^{-1}(L + BG_d) = K^{-1}. \quad (2.64)$$

Solving for G_d gives that

$$G_d = [(G_3 + I_{nn})G_r(\theta_0) + G_1]J_c^{-1}K^{-1} - (G_3 + I_{nn})J_c^T \quad (2.65)$$

where the state feedback matrix, G , has been divided into three $n \times n$ matrices; $G = [G_1 \ G_2 \ G_3]$.

2.2.6 The Control Law

Throughout this derivation, all work has been done in terms of small perturbations from some set point. For completeness, we include the entire control law as it would be implemented on a digital computer as:

$$U = G_1\delta\theta + G_2\delta\dot{\theta} + G_3\delta T + G_d\delta F + F_g(\theta_0) \quad (2.66)$$

$$= G_1(\theta - \theta_0) - G_2\dot{\theta} + G_3(T - T_0) + G_dF + F_g(\theta_0) \quad (2.67)$$

2.2.7 Modifications to the Linear Controller

Reflecting on Kazerooni's linear controller, it becomes apparent that no method has been described for specifying a non-zero force set point. How then does one specify a non-zero contact force for such an application as robotic deburring or contour following?

One solution might be to plot a trajectory that passes behind the surface of interaction: the resulting deviation from the position set point should maintain a contact force as specified by the target dynamics. Immediately, this seems somewhat awkward; it is desired that a contact force be specified, not some modified version of the position trajectory. Furthermore, observe that this method of specification would lead to the robot manipulator moving away from the set point, so that the initial assumption that robot motion will be restricted to small deviations from some set point will be violated for all but very stiff compensators³.

Instead, consider the modification of the control law in order to allow specification of a non-zero force set point. This modification was not included in [16] and represents original work for this thesis.

Returning to equation 2.31, expand the right hand side giving:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + F_g(\theta) = D \quad (2.68)$$

$$= T + J_c^T F_0 + J_c^T \delta F \quad (2.69)$$

³In a further section, it will be shown how interaction force perturbations can lead to a loss of the target dynamics due to this effect.

set point will be violated for all but very stiff compensators³.

Instead, consider the modification of the control law in order to allow specification of a non-zero force set point. This modification was not included in [16] and represents original work for this thesis.

Returning to equation 2.31, expand the right hand side giving:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + F_g(\theta) = D \quad (2.68)$$

$$= T + J_c^T F_0 + J_c^T \delta F \quad (2.69)$$

where F_0 is some force setpoint. Recall that the original definition of the robot dynamics in equation 2.31 includes a term that encompasses non-zero actuator torques required to counteract gravitational force. Modify this term to also include the non-zero actuator torques required to maintain some constant contact force. Rewriting equation 2.69 then gives that

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + F'_g(\theta) = T + J_c^T \delta F \quad (2.70)$$

where we have defined $F'_g(\theta) = F_g(\theta) - J_c^T F_0$. The entire derivation for the controller may then be repeated replacing the original definition of G_r with a definition that incorporates both the torque required to maintain the force set point and the torque required to counteract gravitational forces.

$$G_r(\theta_0) = \left[\frac{\partial F'_g(\theta)}{\partial \theta_1} \frac{\partial F'_g(\theta)}{\partial \theta_2} \dots \frac{\partial F'_g(\theta)}{\partial \theta_n} \right] \Big|_{\theta=\theta_0} \quad (2.71)$$

Making these substitutions into equation 2.67, the complete control law then becomes:

$$U = G_1(\theta - \theta_0) - G_2\dot{\theta} + G_3(T - T_0) + G_d(F - F_0) + F_g(\theta_0) - J_c^T(\theta_0)F_0. \quad (2.72)$$

2.3 A Non-Linear Impedance Controller

In [7], Colgate and Hogan propose a general non-linear approach to the design of impedance controllers. A method of achieving some force/position target dynamics

³In a further section, it will be shown how interaction force perturbations can lead to a loss of the target dynamics due to this effect.

- ϕ represents the actuator states,
- θ represents the manipulator states.
- U is the control input to the actuator,
- F is some measured contact force,
- f_1 describes the actuator dynamics, and
- f_2 describes the manipulator dynamics.

2.3.2 Controller Design

The target dynamics are assumed to have the same canonical form as the plant:

$$\ddot{\theta}_t = f_t(\theta_t, \dot{\theta}_t, \theta_0, \dot{\theta}_0, F). \quad (2.75)$$

where

- θ_t represents the *target* behaviour of the manipulator, that is the desired behaviour of the manipulator in the presence of some force F .
- θ_0 represents some *virtual* trajectory for the manipulator. The term *virtual* refers to the fact that the presence of some external force F will lead to some deviation from the trajectory. In other words, if F was uniformly zero, the target behaviour of the manipulator would be identical to the virtual trajectory.
- f_t defines the target dynamics.

In order to derive the control law, it is also required that the target dynamics be of such a form that the virtual trajectory terms are separable [7], that is:

$$f_t(\theta_t, \dot{\theta}_t, \theta_0, \dot{\theta}_0) = f_{t0}(\theta_0, \dot{\theta}_0) + f_{t1}(\theta_t, \dot{\theta}_t, F). \quad (2.76)$$

The control law is then derived by equating the manipulator motion to the motion of the target model:

$$\theta = \theta_t \quad (2.77)$$

$$\dot{\theta} = \dot{\theta}_t \quad (2.78)$$

$$f_2(\phi, \dot{\phi}, \theta, \dot{\theta}, F) = f_t(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0, F). \quad (2.79)$$

Since the virtual trajectory may be separated from the target dynamics, we can rewrite equation 2.79 as

$$f_2(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2, F) - f_{t1}(\theta_2, \dot{\theta}_2, F) = f_{t0}(\theta_0, \dot{\theta}_0). \quad (2.80)$$

Equation 2.80 is in the form describing the desired behaviour of a servo; the left hand side describes the servo output, and the right hand side describes the input trajectory for the servo. The authors do not address the complex task of performing actuator compensation for the general non-linear system assumed in equations 2.73 and 2.74. However, an example of controller design for a linear system is presented.

2.3.3 Modifications to the Non-Linear Controller Design

Considering the design method presented, it becomes apparent that there are two areas that require further consideration:

1. In the general case, a robot is not limited to one degree of freedom. Also in the general case, a robot may not be modelled by a number of decoupled degrees of freedom (i.e. one for each joint). Consequently, this controller design must be expanded to the n -degree of freedom case.
2. As with the linear controller, the authors present no direct method for specifying a non-zero force trajectory. Of, course, a deviation from the virtual trajectory would reflect some force as specified by the target behaviour of the manipulator, however, this method seems somewhat awkward.

These areas were not addressed in [7], and, as such, this section represents original work carried out for this thesis.

For the first case, the model may be expanded directly to n -degrees of freedom without any effect on the design method. The design problem then becomes analogous to performing actuator compensation on a non-linear robot model. In some cases, as demonstrated in the previous section on the linear impedance controller, it may be valid to linearize the robot model.

Figure 2.2: Linear One Degree of Freedom Manipulator

The second case may be corrected by adding a force trajectory term to the target dynamics; that is, choose the target dynamics as

$$\ddot{\theta}_t = f_t(\theta_t, \dot{\theta}_t, F_t, \theta_0, \dot{\theta}_0, F_0) \quad (2.81)$$

$$= f_{t0}(\theta_0, \dot{\theta}_0, F_0) + f_{t1}(\theta_t, \dot{\theta}_t, F_t) \quad (2.82)$$

The form of the servo dynamics equation remains unchanged:

$$f_2(\phi, \dot{\phi}, \theta, \dot{\theta}, F) - f_{t1}(\theta, \dot{\theta}, F_t) = f_{t0}(\theta_0, \dot{\theta}_0, F_0). \quad (2.83)$$

2.3.4 Design Example

The following design example is based on an example presented in [7], however it has been modified to incorporate a force set point as discussed above. Consider the linear system shown in Figure 2.2.

This system is representative of a linear robot manipulator with actuator dy-

namics. The state space equations describing this manipulator are given by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_1/M_1 & k_1/M_1 & -b_1/M_1 & 0 \\ k_1/M_2 & -(k_1 + k_2)/M_2 & 0 & -b_2/M_2 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} \quad (2.84)$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 1/M_1 \\ 0 \end{bmatrix} U + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/M_2 \end{bmatrix} F. \quad (2.85)$$

Rearranging into the form of equations 2.73 and 2.74 gives

$$\ddot{\theta} = f_2(\phi, \dot{\phi}, \theta, \dot{\theta}, F) \quad (2.86)$$

$$= \frac{k_1}{M_2} \phi - \frac{k_1 + k_2}{M_2} \theta - \frac{b_2}{M_2} \dot{\theta} + \frac{1}{M_2} F \quad (2.87)$$

and

$$\ddot{\phi} = f_1(\phi, \dot{\phi}, \theta, \dot{\theta}, U) \quad (2.88)$$

$$= \frac{k_1}{M_1} \phi + \frac{k_1}{M_1} \theta - \frac{b_1}{M_1} \dot{\phi} + \frac{1}{M_1} U. \quad (2.89)$$

Choose the target dynamics to be essentially identical to those proposed in section 2.2.1, that is a second order damped system:

$$J\ddot{\theta}_t + C(\dot{\theta}_t - \dot{\theta}_0) + K(\theta_t - \theta_0) = F_t - F_0. \quad (2.90)$$

Rearranging equation 2.90 into the form of equation 2.82 gives f_{10} and f_{11} .

$$\ddot{\theta}_t = f_{11}(\theta_t, \dot{\theta}_t, F_t) + f_{10}(\theta_0, \dot{\theta}_0, F_0) \quad (2.91)$$

$$= \left(-\frac{K}{J} \theta_t - \frac{C}{J} \dot{\theta}_t + \frac{1}{J} F_t \right) + \left(\frac{K}{J} \theta_0 + \frac{C}{J} \dot{\theta}_0 - \frac{1}{J} F_0 \right) \quad (2.92)$$

Setting $f_2 - f_{11} = f_{10}$ gives

$$\begin{aligned} \left(\frac{k_1}{M_2} \phi - \frac{k_1 + k_2}{M_2} \theta - \frac{b_2}{M_2} \dot{\theta} + \frac{1}{M_2} F \right) &= \left(\frac{K}{J} \theta_t - \frac{C}{J} \dot{\theta}_t + \frac{1}{J} F_t \right) \\ &= \left(\frac{K}{J} \theta_0 + \frac{C}{J} \dot{\theta}_0 - \frac{1}{J} F_0 \right) \end{aligned}$$

Figure 2.3: Block Diagram of Controller Design

and then with $\theta = \theta_t$, $\dot{\theta} = \dot{\theta}_t$ and $F_t = F$ finally gives that

$$\frac{k_1}{M_2} \phi - \left(\frac{k_1 + k_2}{M_2} - \frac{K}{J} \right) \theta - \left(\frac{b_2}{M_2} - \frac{C}{J} \right) \dot{\theta} \quad (2.93)$$

$$+ \left(\frac{1}{M_2} - \frac{1}{J} \right) F = \frac{K}{J} \theta_0 + \frac{C}{J} \dot{\theta}_0 - \frac{1}{J} F_0. \quad (2.94)$$

Colgate and Hogan build a proportional controller of the form shown in Figure 2.3.

In this example, we select

$$G_1 = \begin{bmatrix} k_1/M_2 \\ 0 \\ -(k_1 + k_2)/M_2 + K/J \\ -b_2/M_2 + C/J \\ 1/M_2 - 1/J \end{bmatrix} \quad (2.95)$$

and

$$G_2 = \begin{bmatrix} k/J \\ C/J \\ -1/J \end{bmatrix} \quad (2.96)$$

in order to satisfy equation 2.94. The scalar gain G then acts as a parameter that determines how closely the system is able to track the desired target dynamics.

2.4 Contact Dynamics and Contact Stability

The term *contact dynamics* refers to the modelling and analysis of interactions between the robot and the environment. *Contact stability* refers to stability when the robot is in contact with the environment. For contact tasks it is necessary to show not only stability of the robot manipulator in unconstrained motion, but stability when the manipulator is in contact with the environment.

After some thought, it becomes apparent that contact dynamics are not necessarily a linear form of interaction. For example, consider the following qualitative observations:

- A typical contact situation will be one where the manipulator and the environment can only exert a force in one direction. For example, one contact interaction could be a robot mounted grinder used to deburr a machined part [18]. Plainly the manipulator and the environment may *push* against each other, however it is impossible for the environment to *pull* on the robot manipulator.
- The deformation of the two objects in contact may itself be non-linear, especially when the contact is initiated rapidly. For example, a large contact force may exceed the elastic limit of the material or large interaction forces may actually affect the material properties; consider how "silly putty" will shatter when hit with a hammer.

The remainder of this section presents an overview of some approaches to contact stability analysis. In the interests of brevity, results are stated, not derived. The reader is referred to the citations for further details.

2.4.1 Contact Stability for the Linear Impedance Controller

In [16], Kazerooni, Sheridan and Houpt show stability of the target dynamics specified for the linear controller in contact with an environment, assuming that the

environment is treated as a second order linear system to which the manipulator is directly coupled. Specifically, Theorem 2 from [16] states that:

If the closed loop dynamic behaviour of the manipulator is given by
(2.97)

$$\begin{aligned} J\delta\ddot{Y}(t) + C\delta\dot{Y}(t) + K\delta Y(t) &= \delta D(t), \\ \delta F(t) \text{ and } \delta Y(t) &\in R^n, J = J^T > 0, \\ K = K^T > 0, C = C^T > 0 \end{aligned} \quad (2.97)$$

and if the environment is a system with the dynamics behaviour represented by (2.98)

$$\begin{aligned} J_e\delta\ddot{Y}_e(t) + C_e\delta\dot{Y}_e(t) + K_e\delta Y_e(t) &= \delta D_e(t) + \delta D_e^0(t), \\ \delta D_e(t), \delta D_e^0(t), \text{ and } \delta Y_e(t) &\in R^m, J_e = J_e^T > 0, \\ K_e = K_e^T > 0, C_e = C_e^T > 0 \end{aligned} \quad (2.98)$$

where

$\delta D_e(t)$ force that the manipulator exerts on the environment

$\delta D_e^0(t)$ all other forces on the environment (uncorrelated with the manipulator states and environment states)

$\delta D(t)$ environmental force on the manipulator

then the overall system (manipulator and environment) is stable.

The authors recognize that the manipulator will, in general, not achieve the target dynamics exactly, but note that this theorem is significant because it shows the desirability of the target dynamics.

They continue to show a stability condition with an additional constraint on the deviation from the target dynamics. (The reader is referred to [16] for the constraint and stability proof.) However, at no point do the authors recognize the fact that in many cases contact dynamics may not be described by a directly coupled second order linear system in this fashion⁴.

⁴In a further section, it will be shown how a robot implementing the specified target dynamics may become unstable during environmental contact.

Figure 2.4: Linear Model of Manipulator/Environmental Interactions.

2.4.2 Contact Stability for the Non-Linear Impedance Controller

In 1988, Colgate and Hogan [7] described the more general controller design method outlined in section 2.3. The authors also included a more general stability proof appropriate to this controller.

Again, the authors assume that the manipulator and the environment may be modelled as two coupled linear systems, as shown in Figure 2.4.

The stability result is not stated as a theorem, however the highlights are presented here. The environment is restricted to passive systems. Define the *driving point impedance* of the closed loop robot system (while not in contact with the environment) as the ratio

$$sY(s)/F(s) \text{ at } s = j\omega. \quad (2.99)$$

The authors then use the Nyquist criteria to show that the combined system of the environment and manipulator will be stable provided that the driving point impedance presented to the environment is strictly positive real for all ω .

Figure 2.5: Interaction of the Robot Manipulator with the Environment

Once again, this proof has the inherent limitation of assuming that contact interactions are linear in nature.

2.4.3 A Non-Linear Stability Proof

More recently, Kazerooni [17, 19, 20] has developed a continuous time stability proof that makes no assumptions about the linearity of the contact dynamics. Within this sub-section, we restrict the notation to be identical to that used in the above references.

Model the system of the robot and the environment using the block diagram given in Figure 2.5. The variables shown in this figure are defined by:

- r A virtual input trajectory command, that is the desired trajectory in the absence of any force disturbance.
- e The input trajectory command.
- y The actual robot position.
- f The interaction force with the environment.
- d The force acting on the robot manipulator.
- x_0 The initial location of the point of contact on the environment before any

Figure 2.6: Simplified Closed Loop System

deformation occurs.

x A vector defining the displacement of the environment.

$G(e)$ A non-linear operator defining the robot's response to the trajectory command.

$E(x)$ A non-linear operator defining the force generated by the environment in response to some deformation x .

$S(d)$ A non-linear operator defining the robot's response to the externally applied force f .

$H(f)$ Some non-linear force feedback compensator.

Also define a mapping from e to f :

$$f = V(e) \quad (2.100)$$

Using this mapping, the closed loop system may be simplified to Figure 2.6.

Prior to stating the stability condition, it is necessary to introduce some concepts from input-output stability theory. The reader is referred to [10] for more information on bounded input-bounded output stability. Consider the definition of the property L taken directly from [10]:

Definition: For any fixed $p \in [1, \infty)$, $f : \mathcal{R}^+ \rightarrow \mathcal{R}$ is said to belong to L_p if and

only if f is locally integrable and

$$\|f\|_p = \left(\int_0^\infty |f(t)|^p dt \right)^{1/p} \leq \infty.$$

Any system that may be represented by some non-linear operator is said to be L_p stable if all inputs, $u \in L_p$, produce outputs $y \in L_p$.

The stability condition for the closed loop system is then given as follows:

If the following three conditions hold:

1. V is an L_p stable operator, that is

$$V(e) : L_p^n \rightarrow L_p^n \text{ and} \quad (2.101)$$

$$\|V(e)\|_p \leq \alpha_4 \|e\|_p + \beta_4 \quad (2.102)$$

2. H is chosen such that the mapping $H(f)$ is L_p stable, that is

$$H(f) : L_p^n \rightarrow L_p^n \text{ and} \quad (2.103)$$

$$\|H(f)\|_p \leq \alpha_5 \|f\|_p + \beta_5 \quad (2.104)$$

3. and

$$\alpha_4 \alpha_5 < 1 \quad (2.105)$$

then the closed loop system is also L_p stable⁵.

The term $\alpha_4 \alpha_5$ represents a bound on the loop gain from e to $H(V(e))$. Consequently, equation 2.105 specifies that the coupled system will be stable provided that the loop gain from the trajectory command is linearly bounded with a slope less than one. The implications of this result are that there must be some initial compliance present in either the environment or the manipulator; contact instability will always result if both the environment and the manipulator are infinitely stiff [8]. Furthermore, α_5 defines a limit on the amount of compliance that may be introduced by force feedback; increasingly large force feedback gains will violate the stability condition and eventually lead to contact instability. (Note that the three conditions are sufficient, but not necessary for stability, so that a system may violate the conditions yet remain stable.)

⁵This proposition is a special case of the small gains theorem. The reader is referred to Appendix A for the definition of the small gains theorem.

Bound on Environmental Stiffness

The stability proposition requires that the loop gain, $H(V(e))$, be uniformly bounded by some constant less than 1. The designer may freely specify the gain of the force feedback component.

Determination of the bound on the $V(e)$ term requires more care. First note that condition 1 requires that $V(e)$ be a stable function when the manipulator is in contact with the environment. It is not clear how condition 1 may be guaranteed, other than experimentally. Nevertheless, once $V(e)$ is determined to be stable, it is not clear how to determine the bound α_4 in order to allow for a force feedback selection based on the limiting values of α_5 .

In short, this stability theory appears reasonable⁶, however further research is needed in order to apply it to real-world design problems.

⁶The experimental results discussed in Chapter 4 will highlight some issues involved in the implementation of this theory.

Chapter 3

Simulations

3.1 Simulation Goals

The principal goals of the robot simulations were to verify that the controllers matched the selected target dynamics and to investigate the effect of deviations from the initial assumptions on the two impedance controllers.

Recall that the target dynamics for the linear impedance controller were selected to be those of a generalized second order linear system. In order to effect a direct comparison, the target dynamics for the non-linear controller were selected identically. This selection also allows for a ready evaluation of performance; compare the simulated response to the response of the equivalent second order linear systems.

The robot model selected for the simulations was that of a serial link manipulator, shown in Figure 3.1. The reader may note that this manipulator is of a different configuration from Carleton University's direct drive robot. Note also that both manipulators are representative of the same class of manipulators whose dynamics may be modelled by equation 2.1, and thus the simulation results form a valid basis for an assessment of expected results on the experimental hardware.

The main argument behind the choice of a serial link manipulator for simulations lies in the relative degree of non-linearity of the five bar linkage configuration of the direct drive robot; the experimental hardware is a nearly decoupled two degree of freedom manipulator. By contrast, the serial link manipulator has a high degree of coupling between joints which generates a relatively larger set of non-linearities.

Figure 3.1: Serial Link Manipulator

Thus the serial link manipulator should provide a more rigorous environment for initial testing of controller performance.

3.2 Robot Simulator

3.2.1 Robot Model

The robot model used for simulations consists of the two degree of freedom linkage shown in Figure 3.1. The dynamic equations for this manipulator are derived in [28] using Lagrangian methods. Specifically, the parameters in equation 2.1 are given by

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad (3.1)$$

where

$$m_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) + I_1 + I_2$$

$$m_{12} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(\theta_2)) + I_2$$

$$m_{21} = m_{12}$$

$$m_{22} = m_2 l_{c2}^2 + I_2$$

and

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -m_2 l_1 l_{c2} \sin(\theta_2) (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \\ m_2 l_1 l_{c2} \sin(\theta_2) \dot{\theta}_1^2 \end{bmatrix} \quad (3.2)$$

and

$$F_g(\theta) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (3.3)$$

The various parameters of the robot model are defined as follows:

- l_1 specifies the length of link 1, in metres,
- l_{c1} specifies the location of the centre of gravity of link 1, in metres from the origin,
- m_1 specifies the mass of link 1,
- I_1 specifies the moment of inertia of link 1 about it's centre of gravity,
- l_2 specifies the length of link 2, in metres,
- l_{c2} specifies the location of the centre of gravity of link 2, in metres from the origin,
- m_2 specifies the mass of link 2, and
- I_2 specifies the moment of inertia of link 2 about it's centre of gravity.

The transformation from joint space to workspace coordinates is given by

$$Y = G(\theta) \quad (3.4)$$

$$= \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (3.5)$$

and the robot Jacobian is given by

$$J_c(\theta) = \begin{bmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix}. \quad (3.6)$$

The values selected for the linkage parameters are given by

$$l_1 = 0.5000 \text{ metres}$$

$$l_{1c} = 0.2500 \text{ metres}$$

$$m_1 = 2.0000 \text{ kg}$$

$$I_1 = 0.0417 \text{ kg} \cdot \text{m}^2$$

$$l_2 = 0.2500 \text{ metres}$$

$$l_{2c} = 0.1250 \text{ metres}$$

$$m_2 = 1.0000 \text{ kg}$$

$$I_2 = 0.0052 \text{ kg} \cdot \text{m}^2.$$

These values lead to numerical values of

$$M(\theta) = \begin{bmatrix} 0.3438 + 0.1250 \cos(\theta_2) & 0.0208 + 0.0625 \cos(\theta_2) \\ 0.0208 + 0.0625 \cos(\theta_2) & 0.0208 \end{bmatrix} \quad (3.7)$$

for the mass matrix,

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -0.0625 \sin(\theta_2)(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ 0.0625 \sin(\theta_2)\dot{\theta}_1^2 \end{bmatrix} \quad (3.8)$$

for the coriolis and centripetal terms, and

$$J_c(\theta) = \begin{bmatrix} -0.50 \sin(\theta_1) - 0.25 \sin(\theta_1 + \theta_2) & -0.25 \sin(\theta_1 + \theta_2) \\ 0.50 \cos(\theta_1) + 0.25 \cos(\theta_1 + \theta_2) & 0.25 \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (3.9)$$

for the manipulator Jacobian.

3.2.2 Simulation Technique

The robot model was simulated numerically using the fourth order Runge-Kutta method for the solution of differential equations [4]. Note that the robot dynamics equation 2.1 must be rearranged into standard form:

$$\ddot{\theta} = M^{-1}(\theta)(D - C(\theta, \dot{\theta}) - F_g(\theta)). \quad (3.10)$$

Implementation of the numerical algorithm is then straight-forward. Computer source code for the robot simulations may be found in [22].

3.3 Controller Designs

The actuators on Carleton's direct drive robot have bandwidths on the order of 2.5 kHz [23]. Consequently, it was reasonable to neglect the actuator dynamics, as the controller bandwidths implemented are typically on the order of 1-10 Hertz, much slower than the actuator bandwidths.

The following subsections describe the two controller design methods neglecting actuator dynamics. These methods were used to design controllers both for robot simulations and to be run on the experimental hardware, taking advantage of the common form of the robot dynamics model shared by both manipulators.

3.3.1 The Linear Impedance Controller

Neglecting actuator dynamics, design of the linear controller is significantly simplified. Maintain the assumptions:

1. The manipulator moves slowly, so that $C(\theta, \dot{\theta}) \simeq 0$, and
2. The manipulator only makes small deviations about the set point, θ_0 , so that joint coordinates can be transformed to Cartesian coordinates by:

$$\delta Y = J_c(\theta_0) \delta \theta \quad (3.11)$$

$$\delta \dot{Y} = J_c(\theta_0) \delta \dot{\theta} \quad (3.12)$$

$$\delta \ddot{Y} = J_c(\theta_0) \delta \ddot{\theta} \quad (3.13)$$

The simplified robot dynamics then become:

$$M(\theta_0) \delta \ddot{\theta} + G_r(\theta_0) = \delta T + J_c^T(\theta_0) \delta F \quad (3.14)$$

Maintain the target dynamics specified in equation 2.13, namely:

$$\delta F = K \delta Y + C \delta \dot{Y} + J \delta \ddot{Y} \quad (3.15)$$

In state space form the target dynamics may be written as

$$\begin{bmatrix} \delta \dot{Y}(t) \\ \delta \ddot{Y}(t) \end{bmatrix} = \begin{bmatrix} 0_{nn} & I_{nn} \\ -J^{-1}K & -J^{-1}C \end{bmatrix} \begin{bmatrix} \delta Y(t) \\ \delta \dot{Y}(t) \end{bmatrix} +$$

$$\begin{bmatrix} 0_{nn} \\ J^{-1} \end{bmatrix} \delta F(t) \quad (3.16)$$

The linearized robot dynamics in state space form are given by:

$$\begin{bmatrix} \delta \dot{\theta} \\ \delta \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0_{nn} & I_{nn} \\ 0_{nn} & 0_{nn} \end{bmatrix} \begin{bmatrix} \delta \theta \\ \delta \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0_{nn} \\ M^{-1}(\theta_0) \end{bmatrix} \delta T(t) + \begin{bmatrix} 0_{nn} \\ M^{-1}(\theta_0) J_c^T \end{bmatrix} \delta F(t) \quad (3.17)$$

Recall that the control law for the linear controller is given by

$$\delta T = -G_1 \delta \theta - G_2 \delta \dot{\theta} + G_d \delta F \quad (3.18)$$

so that the closed loop controller dynamics are given by:

$$\begin{bmatrix} \delta \dot{\theta} \\ \delta \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0_{nn} & I_{nn} \\ -M^{-1}(\theta_0) G_1 & -M^{-1}(\theta_0) G_2 \end{bmatrix} \begin{bmatrix} \delta \theta \\ \delta \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0_{nn} \\ M^{-1}(\theta_0) (J_c^T + G_d) \end{bmatrix} \delta F \quad (3.19)$$

where the state feedback gains matrix has been written as $G = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$.

The value of G_1 , G_2 and G_d can be obtained by equating equations 3.16 and 3.19 as

$$\begin{aligned} G_1 &= M(\theta_0) J_c^{-1} J^{-1} K J_c \\ G_2 &= M(\theta_0) J_c^{-1} J^{-1} C J_c \\ G_d &= M(\theta_0) J_c^{-1} J^{-1} - J_c^T \end{aligned} \quad (3.20)$$

Once again, this derivation of controller gains applies for small deviations about the set point, θ_0 . Making the following substitutions:

$$\begin{aligned} \delta \theta &= \theta - \theta_0 \\ \delta \dot{\theta} &= \dot{\theta} \\ \delta F &= F - F_0 \\ \delta T &= T - T_0 \\ T_0 &= -J_c^T(\theta_0) F_0 - G_r(\theta_0) \end{aligned}$$

the control law can be written as

$$T - T_0 = -G_1(\theta - \theta_0) - G_2\dot{\theta} + G_d(F - F_0) \quad (3.21)$$

or

$$T = -G_1(\theta - \theta_0) - G_2\dot{\theta} + G_d(F - F_0) - J_c^T(\theta_0)F_0 - G_r(\theta_0) \quad (3.22)$$

3.3.2 The Non-Linear Impedance Controller

Similar results may be obtained for the non-linear impedance controller; neglecting actuator dynamics, the robot model simplifies to

$$\dot{V} = f(X, V, E, F) \quad (3.23)$$

where

- V - represents a vector of velocity terms,
- X - represents a vector of position terms,
- E - represents a vector of inputs,
- F - represents the external forces acting on the system, and
- $f(X, V, E, F)$ - defines the system dynamics.

and the target model simplifies to

$$\dot{V}_t = f_t(X_t, V_t, F_t, X_0, V_0, F_0) = f_{t0}(X_0, V_0, F_0) + f_{t1}(X_t, V_t, F_t). \quad (3.24)$$

For the purposes of comparison, choose the same generalized second order linear target dynamics as for the linear controller:

$$F - F_0 = K(Y - Y_0) + C\dot{Y} + J\ddot{Y} \quad (3.25)$$

Note that these target dynamics are no longer restricted to small deviations about some set point. Furthermore, if desired, a non-zero velocity could be chosen, allowing for the generation of a moving robot trajectory.

Also note that the driving point impedance of the closed loop robot manipulator, as defined by equation 2.99, is given by

$$\begin{aligned} Z(s) &= \left. \frac{sY(s)}{F(s)} \right|_{s=j\omega} \\ &= \frac{j\omega}{K + Cj\omega - J\omega^2} \\ &= \frac{j\omega(K - J\omega^2) + C\omega^2}{(K - J\omega^2)^2 + (C\omega)^2} \end{aligned} \quad (3.26)$$

Then

$$\text{real}(Z) = \frac{C\omega^2}{(K - J\omega^2)^2 + (C\omega)^2}. \quad (3.27)$$

Plainly the driving point impedance is strictly positive real for any real valued J , C , and K . Consequently, the target dynamics satisfy the stability condition of section 2.4.2, and the closed loop robot system should be stable in contact with any directly coupled linear environment.

Again assume that the manipulator moves relatively slowly, so that the coriolis and centripetal terms can be neglected. The robot dynamics are then given by

$$M(\theta)\ddot{\theta} + G_r(\theta) = T + J_c^T(\theta)F \quad (3.28)$$

Since we are no longer restricted to small deviations from some set point, the mapping from joint space to Cartesian space now becomes

$$Y = G(\theta) \quad (3.29)$$

$$\dot{Y} = J_c(\theta)\dot{\theta} \quad (3.30)$$

$$\ddot{Y} = J_c(\theta)\ddot{\theta} + \dot{\theta} \frac{dJ_c(\theta)}{dt} \quad (3.31)$$

$$= J_c(\theta)\ddot{\theta} + \dot{\theta}\dot{\theta}^T \frac{\partial J_c(\theta)}{\partial \theta} \quad (3.32)$$

$$\simeq J_c(\theta)\ddot{\theta} \quad (3.33)$$

where, in equation (3.33) we have used assumption 2 from section 3.3.1 that the robot will be moving slowly, so that terms involving $\dot{\theta}_i\dot{\theta}_j$ will be negligible ¹

¹In this example, it would be simple enough to include the $\dot{\theta}_i\dot{\theta}_j$ and coriolis and centripetal terms to achieve the target dynamics perfectly. However, inclusion of these terms would lead to violation of the separability condition required in equation 3.24, which would make it impossible to correct for actuator dynamics.

Writing the target dynamics in joint space form gives:

$$JJ_c(\theta)\ddot{\theta} + CJ_c(\theta)\dot{\theta} + K(G(\theta) - G(\theta_0)) = F - F_0. \quad (3.34)$$

The most direct way in which to arrive at the control law is to substitute for $\ddot{\theta}$ from the robot dynamics (equation 3.28):

$$\ddot{\theta} = M^{-1}(\theta)T + M^{-1}(\theta)J_c^T(\theta)F - M^{-1}(\theta)G_r(\theta) \quad (3.35)$$

Substituting 3.35 into 3.34 gives the control law

$$T = (M(\theta)J_c^{-1}(\theta)J^{-1} - J_c^T)F - M(\theta)J_c^{-1}(\theta)J^{-1}F_0 - \quad (3.36)$$

$$M(\theta)J_c^{-1}(\theta)J^{-1}CJ_c(\theta)\dot{\theta} + \quad (3.37)$$

$$M(\theta)J_c^{-1}(\theta)J^{-1}K(G(\theta_0) - G(\theta)) + M(\theta)J_c^{-1}(\theta)G_r(\theta) \quad (3.38)$$

$$= G_1F - G_2F_0 - G_3\dot{\theta} + G_4(G(\theta_0) - G(\theta)) + G_5G_r(\theta). \quad (3.39)$$

where the gain matrices are set to be:

$$\begin{aligned} G_1 &= M(\theta)J_c^{-1}(\theta)J^{-1} - J_c^T \\ G_2 &= M(\theta)J_c^{-1}(\theta)J^{-1} \\ G_3 &= G_2CJ_c(\theta) \\ G_4 &= G_2KJ_c(\theta) \\ G_5 &= M(\theta)J_c^{-1}(\theta). \end{aligned} \quad (3.40)$$

Note that the gain matrices are time varying for this controller, while the gain matrices for the linear controller are constant.

3.4 Simulation Results

The direction of the simulations evolved as the research described in this thesis progressed. Simulations were used to investigate a total of three different areas:

1. Verification of the target dynamics in the closed loop robot system.

2. Determining the potential effects of a loss of linearization (for the linear impedance controller) resulting from a displacement from the specified set point.
3. Illustration of the different behaviour between joint space controllers (the linear controller) and workspace controllers (the non-linear controller).

3.4.1 Verification of Target Dynamics - the Linear Controller

The first simulations with the linear controller consist of giving the simulation some initial condition different from the set point and releasing it. The manipulator response can then be compared to the ideal response of the equivalent second order linear system of the specified target dynamics.

Simulation parameters are given in Table 3.1.

The manipulator trajectories for critically damped target dynamics are shown in Figure 3.2. Note that the controller bandwidth is limited to $10 \text{ rad/sec} = 1.59 \text{ Hertz}$, and the initial condition is 0.02 metres from the set point, so that the assumptions regarding slow manipulator motion in the neighbourhood of the set point are followed. By inspection, the manipulator response follows the ideal trajectory quite closely.

In Figure 3.3, the manipulator has been given an initial condition of a somewhat larger distance of 0.20 metres from the set point, so that the assumption about motion in the neighbourhood of the set point has been stretched. Again by inspection, the manipulator response behaves quite well, showing a bit more deviation.

In Figure 3.4, the target dynamics were chosen to be underdamped with a relatively small damping factor of $\zeta = 0.25$. The manipulator response obtained is again essentially the expected response.

From the results of these three simulations, it seems reasonable to conclude that the controller performs as expected in an unconstrained environment.

Now consider the results shown in Figure 3.5. A force of 10.0 Newtons is being applied to the manipulator tip. As the manipulator is moved away from the set

Figure	Target Dynamics	Position Parameters (metres)	Force Parameters (Newtons)
3.2, 3.7	$J = \begin{pmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{pmatrix}$ $C = \begin{pmatrix} 200.0 & 0.0 \\ 0.0 & 200.0 \end{pmatrix}$ $K = \begin{pmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{pmatrix}$	Set Point (0.40, 0.40)	Set Point (0.00, 0.00)
		Initial Point (0.38, 0.38)	Simulated Force (0.00, 0.00)
3.3, 3.8	$J = \begin{pmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{pmatrix}$ $C = \begin{pmatrix} 200.0 & 0.0 \\ 0.0 & 200.0 \end{pmatrix}$ $K = \begin{pmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{pmatrix}$	Set Point (0.40, 0.40)	Set Point (0.00, 0.00)
		Initial Point (0.20, 0.20)	Simulated Force (0.00, 0.00)
3.4, 3.9	$J = \begin{pmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{pmatrix}$ $C = \begin{pmatrix} 50.0 & 0.0 \\ 0.0 & 50.0 \end{pmatrix}$ $K = \begin{pmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{pmatrix}$	Set Point (0.40, 0.40)	Set Point (0.00, 0.00)
		Initial Point (0.30, 0.30)	Simulated Force (0.00, 0.00)
3.5, 3.10	$J = \begin{pmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{pmatrix}$ $C = \begin{pmatrix} 200.0 & 0.0 \\ 0.0 & 200.0 \end{pmatrix}$ $K = \begin{pmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{pmatrix}$	Set Point (0.40, 0.40)	Set Point (0.00, 0.00)
		Initial Point (0.40, 0.40)	Simulated Force (10.00, 10.00)
3.6, 3.11	$J = \begin{pmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{pmatrix}$ $C = \begin{pmatrix} 200.0 & 0.0 \\ 0.0 & 200.0 \end{pmatrix}$ $K = \begin{pmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{pmatrix}$	Set Point (0.40, 0.40)	Set Point (0.00, 0.00)
		Initial Point (0.40, 0.40)	Simulated Force (100.00, 100.00)

Table 3.1: Summary of Simulation Parameters

- Simulated robot response.
--- Target dynamics response.

Figure 3.2: Simulation Response of Non-Linear Controller to Critically Damped Target Dynamics - Small Initial Deviation.

- Simulated robot response.
--- Target dynamics response.

Figure 3.3: Simulation Response of Linear Controller to Critically Damped Target Dynamics - Large Initial Deviation.

Figure 3.4: Simulation Response of Linear Controller to Underdamped Target Dynamics.

point, the manipulator dynamics tend to deviate somewhat from the target dynamics, however they still appear reasonable. There is a noticeable steady state error resulting.

However, the results shown in Figure 3.6 do not appear so reasonable. In this simulation, a larger force of 100.0 Newtons has been applied to the manipulator tip. This time, as the manipulator moves farther from the set point, the manipulator dynamics deviate significantly, leading to a very large steady state error.

Informally, the phenomena just observed can be described as follows:

1. A force is applied to the manipulator, moving it away from the set point.
2. As the manipulator moves away from the set point, the linearization about the set point is lost, leading to a deviation from the target dynamics.
3. As increasingly large forces are applied, the deviation from the target dynamics grows.

Figure 3.5: Simulation Response of Linear Controller to Small Applied Force.

Figure 3.6: Simulation Response of Linear Controller to Large Applied Force.

Figure 3.7: Simulation Response of Non-Linear Controller to Critically Damped Target Dynamics - Small Initial Deviation.

3.4.2 Verification of Target Dynamics - the Non-Linear Controller.

Having examined results for the linear controller, we can make some observations about how the non-linear controller might be expected to behave. Recall that this controller is not subject to the linearization assumptions in the neighbourhood of some set point, but only to the assumption that manipulator motion will be slow enough that coriolis and centripetal acceleration terms can be neglected. Consequently, we would expect this controller to exhibit similar behaviour in unconstrained conditions as the linear controller. However, when the manipulator has been subjected to an applied force, there should not be any steady state error.

Through inspection of Figures 3.7, 3.8, and 3.9 it is apparent that the unconstrained behaviour of the manipulator is effectively the same as it was with the linear controller.

Furthermore, inspecting Figures 3.10 and 3.11 it is apparent that there is no

—— Simulated response of robot manipulator.
--- Target dynamics response.

Figure 3.8: Simulation Response of Non-Linear Controller to Critically Damped Target Dynamics - Large Initial Deviation.

—— Simulated response of robot manipulator.
--- Target dynamics response.

Figure 3.9: Simulation Response of Non-Linear Controller to Underdamped Target Dynamics.

Figure 3.10: Simulation Response of the Non-Linear Controller to a Small Applied Force.

steady state error when a force is applied to the manipulator end point.

3.4.3 Joint Space versus Work Space

Recall from chapter 2 that the linear impedance controller is a joint space controller; the selected states are the robot joints. The non-linear controller designed is a workspace controller in that the selected states were the end-effector positions and velocities.

This difference between the two controllers is not readily apparent; further discussion will clarify specific items.

With the linear controller, the work space to joint space transformation occurs once — at the set point when the gain matrices are computed. Onwards from this initial computation, all calculations are done in terms of joint coordinates.

The non-linear controller requires that the gain matrices be continuously recomputed. Referring back to equation 3.40 it is apparent that the work space to joint space transformation is being continuously recomputed. In effect, all calculations

Figure 3.11: Simulation Response of the Non-Linear Controller to a Large Applied Force.

are being done in terms of workspace coordinates and then transformed into the joint space.

In the previous section, we found that switching the controller from joint space to work space eliminated problems associated with the loss of the point of linearization. One tradeoff was the increase in complexity in the non-linear controller over the linear controller.

Now consider a more serious tradeoff; what happens when the manipulator is pushed towards a singularity by some unanticipated force? The results are shown in Figure 3.12. The linear controller loses the point of linearization and deviates from the target dynamics before the manipulator reaches the singularity. However, the non-linear controller moves directly to the singularity. As the manipulator reaches the singularity, the workspace to joint space transformation approaches a discontinuity (where the Jacobian inverse, J_c^{-1} , no longer exists) and the system becomes unstable.

In unconstrained motion, this problem could be addressed by planning a trajec-

a) Linear impedance controller approaching singularity.

b) Non-linear impedance controller approaching singularity.

— Simulation results.

--- Target dynamics response.

Figure 3.12: Controller Behaviour in the Neighbourhood of a Singularity

tory that avoids all such singularities within the robot workspace. In constrained motion, more care must be taken; it is conceivable that the robot could come into unexpected contact with the environment. If the position trajectory was not planned correctly, this contact could push the manipulator into a singularity, leading to instability.

3.4.4 Summary of Simulation Results

Quantitative results for the simulations presented in this chapter are shown in Table 3.2. The definition of root mean square error used in this table was taken from [4]:

$$E_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_{di})^2} \quad (3.41)$$

where

E_{rms} is the root mean square error;

n is the number of data points collected;

y_i is the measured data, in this case the workspace coordinates of the simulated robot, and

y_{id} is the desired response, in this case the workspace coordinates of the target dynamics.

Simulation results showed that the performance of the two controllers was essentially equivalent for unconstrained motion. This observation is reflected in the data shown in Table 3.2 for the first six rows of data; there is no dramatic difference apparent in the error terms.

The non-linear controller dramatically outperformed the linear controller when some external force was applied to the manipulator; this effect is shown in the last four rows of Table 3.2. The errors in the linear controller are two orders of magnitude greater than the errors in the non-linear controller.

The final result of the simulations is that the non-linear controller exhibits severe instability at robot singularities. The linear controller is not subject to these instabilities.

	Linear Controller		Non-Linear Controller	
	Figure	RMS Error (mm)	Figure	RMS Error (mm)
X	3.2	0.386	3.7	0.085
Y	3.2	0.286	3.7	0.134
X	3.3	13.203	3.8	4.870
Y	3.3	9.035	3.8	14.069
X	3.4	6.994	3.9	6.363
Y	3.4	4.459	3.9	6.092
X	3.5	1.844	3.10	0.050
Y	3.5	3.220	3.10	0.062
X	3.6	61.941	3.11	1.152
Y	3.6	82.402	3.11	0.692

Table 3.2: Summary of Simulation Results

This page intentionally left blank.

Chapter 4

Experimental Results

The purpose of this chapter is to provide a description of the experimental investigation carried out for the two impedance controllers.

Experiments were carried out in order to investigate three different areas:

1. Verification that each controller achieved some reasonable approximation of the specified target dynamics;
2. A detailed comparison of the performance of the two controllers, taking into account the differences that would be expected from the simulation results outlined in chapter 3; and
3. A systematic investigation into contact stability.

The results presented in this chapter show verification of previously published contact stability results and some new observations into difficulties inherent in implementing an impedance controller.

Specifically, it is shown that the controllers successfully reproduced the target dynamics for unconstrained motion (force constrained to zero, no position constraints) on the experimental hardware. The differences observed in Chapter 3 as a result of the loss of the linearization point in the linear controller were not present in the experimental setup due to actuator output limitations.

Some original results with regard to limitations inherent in controlling contact force have been observed. Cross coupling of actuator motor torques can lead to severe actuator saturation while maintaining reasonable force trajectories.

Systematic experiments verify that contact stability cannot be guaranteed for a high level of force feedback.

The remainder of this chapter is divided into five sections, each describing, respectively, the experimental equipment, the target dynamics verification, the controller comparison, some observations into force trajectory limitations and the investigation into contact stability.

4.1 Experimental Setup

4.1.1 The Direct Drive Robot

The direct drive robot available in the robotics laboratory at Carleton University was used to obtain all experimental results. This robot is a five bar linkage configuration [28] operating in the horizontal plane. A schematic illustration of the robot is shown in Figure 4.1.

This robot was designed and manufactured by the Department of Systems and Computer Engineering at Carleton University. Details of the mechanical design may be found in [27]. Details of the actuator and other electrical design are documented in [29].

The direct drive robot is controlled by an IBM PC-AT(r) compatible computer equipped with data acquisition peripherals and an 80287 numeric coprocessor extension. All control software was developed in a combination of Turbo Pascal [6] and 80286/80287 assembler. The developed software has been documented separately [22] due to its lengthy nature.

4.1.2 Force Sensor

The JR³ Universal Force Moment Sensor System, model number UFS-3012A50-U560 [14] was selected and purchased for use with the direct drive robot. This force sensor provides force and torque measurements in all 6 degrees of freedom using strain gauge technology. Force measurements were digitized using spare channels

Figure 4.1: Carleton's Direct Drive Robot

on the 12 bit analog to digital converter present on the existing data acquisition equipment within the PC-AT compatible computer.

Adaptor plates and a special rigid, light weight end-effector were manufactured in order to mount the force sensor on the robot. A top level drawing of this assembly is shown in Figure 4.2.

Complete details of the physical and electrical installation and integration of the force sensor into the robot system are given in Appendix B.

4.1.3 Robot Properties Modelling

Referring to Chapter 2, it is plain that a reasonable model of the robot's mass and kinematic properties is required in order to implement either of the impedance controllers.

A thorough description of the manipulator linkage lengths was included in [27]. These measurements specify the joint lengths shown in Figure 4.1 as

$$l_1 = 0.3048$$

$$l_2 = 0.4699$$

From these measurements, the transformations from joint coordinates to workspace

Figure 4.2: Universal Force Sensor Mounting Assembly

coordinates may be easily obtained.

Workspace coordinates are related to joint angles by the following transformation:

$$Y_1 = l_1 \cos(\theta_1) - l_4 \cos(\theta_2) \quad (4.1)$$

$$= 0.3048 \cos(\theta_1) - 0.4699 \cos(\theta_2) \quad (4.2)$$

$$Y_2 = l_1 \sin(\theta_1) - l_4 \sin(\theta_2) \quad (4.3)$$

$$= 0.3048 \sin(\theta_1) - 0.4699 \sin(\theta_2) \quad (4.4)$$

The robot Jacobian may then be derived from equation 2.5 as:

$$J_c = \begin{bmatrix} -l_1 \sin(\theta_1) & l_4 \sin(\theta_2) \\ l_1 \cos(\theta_1) & -l_4 \cos(\theta_2) \end{bmatrix} \quad (4.5)$$

$$= \begin{bmatrix} -0.3048 \sin(\theta_1) & 0.4699 \sin(\theta_2) \\ 0.3048 \cos(\theta_1) & -0.4699 \cos(\theta_2) \end{bmatrix} \quad (4.6)$$

No existing estimate for the mass properties of the direct drive robot was available. As a result, it was necessary to estimate the mass properties using a least squares model.

Recall from equations 3.14 and 3.28 that slow manipulator motion was assumed, so that the coriolis and centripetal acceleration terms represented by $C(\theta, \dot{\theta})$ in equation 2.1 may be neglected. It may be shown [28] that the mass matrix for the five bar linkage configuration is of the form

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} \cos(\theta_2 - \theta_1) \\ m_{12} \cos(\theta_2 - \theta_1) & m_{22} \end{bmatrix} \quad (4.7)$$

Consequently, it was decided that it would be sufficient to estimate the terms m_{11} , m_{12} , and m_{22} while neglecting $C(\theta, \dot{\theta})$ in order to determine the mass properties of the direct drive robot.

Details of the least squares estimation are included in Appendix B. The properties determined for the mass matrix were:

$$m_{11} = 0.995 \text{ kg} \cdot \text{m}^2$$

$$m_{12} = 0.0012 \text{ kg} \cdot \text{m}^2$$

$$m_{22} = 0.861 \text{ kg} \cdot \text{m}^2.$$

4.1.4 Environmental Contact Setup

For the experimental investigation into contact stability, it was necessary to provide some type of environment that would give consistent results for contact experiments. For the purposes of this research, a one half inch steel plate was used as the environment. Additionally, a strip of one quarter inch rubber was bonded to the plate with contact cement. The steel plate was then clamped securely to a table. The plate could be raised or lowered to provide contact with either the steel surface or the rubber surface.

Thus two surfaces were available for contact stability experiments; the relatively stiff surface comprised of the steel plate, and the more compliant surface consisting of the rubber backed by the steel plate.

The motivation for choosing both a stiff and compliant environment was twofold:

1. It is well known (see, for example [9]) that unstable contact may be stabilized by introducing some compliancy in the form of foam or rubber padding.
2. From the contact stability theory presented in Section 2.4.3, it is apparent that the stiffness of the environment should have some effect on the sufficiency condition for contact stability.

A schematic of the contact setup showing the steel plate and rubber padding is given as Figure 4.3.

4.2 Verification of Target Dynamics

In order to verify that each controller achieved the specified target dynamics, it was necessary to proceed in much the same way as outlined in chapter 3; the manipulator is moved to some initial displacement from the set point and released. The resulting behaviour can then be compared to the ideal second order response of the target dynamics.

Recall from Chapter 3 that simulations showed both controllers approximated the target dynamics quite well in an unconstrained situation. Some differences

Figure 4.3: Compliant Contact Experimental Setup

can be anticipated between the experimental results and the simulation results. In particular, the experimental results would be expected to show measurement noise not included in the simulations and some steady state error as a result of static friction in the manipulator joints and dead zones in the actuators.

The selected target dynamics and summarized results are shown in Table 4.1. The following sub-sections include plots of the experimental results and further discussion.

4.2.1 Linear Controller

Results for the critically damped target dynamics are shown in Figures 4.4 and 4.5. Qualitatively, the manipulator appears to follow the target response quite well. Additionally, examination of the error terms in Table 4.1 shows that the RMS errors are of the same order as those shown for the simulations in Table 3.2 for the large initial deviation. (The large initial deviation simulations correspond to Figures 3.3 and 3.8.) For the smaller initial deviation of 0.02 metres, the experimental errors are an order of magnitude larger than the simulation errors. This difference may be

Linear Controller						
	Figure	Target Dynamics	Initial Point (metres)	Set Point (metres)	RMS Error (mm)	Steady State Error (mm)
X	4.4	$J_x = 10$ $C_x = 200$ $K_x = 1000$	0.38	0.40	4.24	-4.59
Y	-	$J_y = 10$ $C_y = 200$ $K_y = 1000$	0.38	0.40	3.45	3.77
X	4.5	$J_x = 10$ $C_x = 200$ $K_x = 1000$	0.20	0.40	19.78	-5.68
Y	-	$J_y = 10$ $C_y = 200$ $K_y = 1000$	0.20	0.40	32.83	3.45

Non-Linear Controller						
	Figure	Target Dynamics	Initial Point (metres)	Set Point (metres)	RMS Error (mm)	Steady State Error (mm)
X	4.6	$J_x = 10$ $C_x = 200$ $K_x = 1000$	0.38	0.40	4.97	-4.85
Y	-	$J_y = 10$ $C_y = 200$ $K_y = 1000$	0.38	0.40	3.57	3.92
X	4.7	$J_x = 10$ $C_x = 200$ $K_x = 1000$	0.20	0.40	49.69	-5.50
Y	-	$J_y = 10$ $C_y = 200$ $K_y = 1000$	0.20	0.40	27.03	3.29

Table 4.1: Summary of Target Dynamics Verification Results

—— Experimental measurements --- Target dynamics response.

Figure 4.4: Linear Controller: Manipulator Response to Small Initial Displacement

attributed to the steady state error caused by static friction and motor dead zones.

Inspecting Figures 4.4 and 4.5, observe that the robot trajectories are both subject to a steady state error of approximately five millimetres. (Precise values are shown in Table 4.1.) From the discussion in Section 4.2, this behaviour is exactly that to be expected from experimental hardware subject to static friction and/or motor dead zones. Thus the proportionately larger RMS error for the small initial position deviation is a direct result of the steady state error being included in the calculation of RMS error.

4.2.2 Non-Linear Controller

Results for the non-linear controller are shown in Figures 4.6 and 4.7. Again, observe that the manipulator appears to follow the target response quite well. Note also the similarity between the results for the linear controller and the non-linear controller. Comparing the results shown for the non-linear controller in Table 4.1 to the simulation results for the non-linear controller in Table 3.2, it is apparent that the errors are of exactly the same form as those of the linear controller.

— Experimental measurements. - - - Target dynamics response.

Figure 4.5: Linear Controller: Manipulator Response to Large Initial Displacement

— Experimental measurements. - - - Target dynamics response.

Figure 4.6: Non-Linear Controller: Manipulator Response to Small Initial Displacement

— Experimental measurements. - - - Target dynamics response.

Figure 4.7: Non-Linear Controller: Manipulator Response to Large Initial Displacement

4.3 Comparison of Controller Performance

At this point, it seems reasonable to conclude that the controllers are implementing the specified target dynamics on the experimental hardware. It is now of interest to perform a comparison between the two controllers. The basis for this comparison will be the loss of the point of linearization observed in the simulation results for the linear impedance controller.

Recall from Chapter 3 that the difference in controller performance became apparent when a force disturbance was applied to the manipulator; the linear controller was subject to significant steady state error as a result of deviation from the point of linearization, while no effect was observed on the non-linear controller.

In the simulations, this effect was observed by applying a constant force to the manipulator. It would be difficult to maintain such a force constraint on the experimental hardware. Instead, the manipulator was given a trajectory that passed through the environment, similar to that shown in Figure 4.8.

Some explanation is required for the manipulator trajectory; both controllers

Figure 4.8: Manipulator Position Trajectory for Controller Comparison

were designed about a set point, not a trajectory. The trajectory was implemented by recomputing the gain matrices for a series of set points along the trajectory.

An illustrative way of viewing this type of trajectory generation is to consider an equivalent single degree of freedom case based on the target dynamics. The target dynamics represent a damped mass spring combination. The mass can be made to follow some trajectory by moving the opposite end of the spring along the desired path: the mass will then also follow the desired path with dynamics determined by the mass, spring and damping parameters.

Analogously, the robot manipulator will follow the specified trajectory with some delay as determined by the target dynamics. For slow motion, the delay should be negligible. Exact determination of this delay could be determined by analysing the transient response of the target dynamics.

The results of running the manipulator into the environment in this fashion are shown in Figure 4.9. The target dynamics are given by:

$$J = \begin{bmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{bmatrix}, C = \begin{bmatrix} 200.0 & 0.0 \\ 0.0 & 200.0 \end{bmatrix}, K = \begin{bmatrix} 1000.0 & 0.0 \\ 0.0 & 1000.0 \end{bmatrix}.$$

Observe the measured force; as the trajectory receded behind the environment, the setpoint was displaced an increasing distance from the manipulator tip. Consequently, the measured force ramped up according to the stiffness specified in the target dynamics. Some oscillation was observed during the contact, however the contact remained stable.

In the experiments with both the linear and the non-linear manipulator, the robot actuators reached their output limits before significant variation in performance could be observed. This actuator output saturation may be seen plainly in Figure 4.10.

Recall that the stiffness of the target dynamics was relatively high, namely 1000 N/m. Presumably, if more compliant target dynamics were selected, a more significant deviation from the target dynamics could be achieved for the linear controller.

However, as will be discussed in Section 4.5, increasing the compliance of the target dynamics tends to lead to contact instability. As can be seen plainly in Figure 4.9, the contact force showed a tendency to oscillate. In fact, these target dynamics were chosen in an ad hoc manner as being the best to illustrate the results of the controller comparison; more compliant target dynamics led to instability, while less compliant results gave less indication of the measured force response.

These results suggest that the additional complexity of the non-linear controller provides little advantage; before the manipulator could be perturbed significantly from the set point, the motor outputs saturated. In any event, these results show an engineering tradeoff in controller selection; some analysis of the various operating parameters would be required prior to selection of either controller. This analysis would include such factors as:

- Peak actuator outputs.
- Robot configuration and the joint space to workspace transformations.
- Available processing time in the computer controlling the robot.

a. Linear controller force measurements.

b. Non-linear controller force measurements.

—— Measured force.

--- Ideal force response.

Figure 4.9: Force Response Comparison

a. Linear controller motor 1 output.

b. Non-linear controller motor 1 output.

Figure 4.10: Force Response Comparison - Actuator Output

4.4 Force Trajectory Limitations

During these experiments, another effect was observed. The contact force that the robot was capable of maintaining was somewhat dependant on the manipulator configuration.

This effect is tied to the static configuration of the robot. Recall that the actuator torque required to maintain a contact force is given by equation 2.12:

$$T = J_c^T(\theta)F. \quad (4.8)$$

The contact force is thereby limited by both the peak actuator torque and the specified robot configuration; the best case for maintaining contact forces independently occurs when the robot Jacobian is diagonal; in this configuration the actuators are decoupled with respect to the workspace coordinates and each actuator can independently apply a force in a single workspace coordinate.

As the configuration moves from this best case, an increasing amount of coupling is obtained between the manipulator joints with respect to the workspace coordinates. Consequently, the locus of reachable applied forces changes. Plainly the worst case occurs at robot singularities; here the robot has lost a degree of freedom, and no realizable actuator torque can generate a contact force in the newly constrained direction.

Consider the experimental results presented in Figure 4.11. Here, the robot has been given a position trajectory moving along a wall, and a force trajectory against the wall. The robot trajectory is shown schematically in Figure 4.14. The target dynamics were chosen to be

$$J = \begin{bmatrix} 5.0 & 0.0 \\ 0.0 & 10.0 \end{bmatrix}, \quad C = \begin{bmatrix} 1000.0 & 0.0 \\ 0.0 & 200.0 \end{bmatrix}, \quad K = \begin{bmatrix} 10000.0 & 0.0 \\ 0.0 & 200.0 \end{bmatrix}$$

Note that the target dynamics were chosen to be stiff along the wall, to ensure good tracking of the position trajectory, and compliant normal to the wall, to ensure good tracking of the force trajectory. Damping was chosen so as to specify target dynamics with no overshoot in both directions. The specified contact force was

10 Newtons in the direction normal to the wall, and 0 Newtons in the direction tangential to the surface of the wall.

Examine the results for the force trajectory; initially, the tracking is relatively good. As the manipulator moves along the wall, the tracking performance degrades, until large deviations from the desired contact force are observed.

Consider a numerical example of this effect. Examine two robot configurations: one fully decoupled into workspace coordinates and a more highly coupled case. These configurations correspond approximately to the robot configuration at the beginning of the trajectory and at the end of the trajectory.

For the first case, take $\theta_1 = 0^\circ$ and $\theta_2 = -90^\circ$. The robot Jacobian is given by

$$J_c = \begin{bmatrix} 0.0 & -0.4699 \\ 0.3048 & 0.0 \end{bmatrix}$$

In order to maintain the specified contact force of $F = [0.0 \quad 10.0]^T$ the required actuator torques are

$$\begin{aligned} T &= J_c^T(\theta)F \\ &= \begin{bmatrix} -3.05 \\ 0.0 \end{bmatrix} \end{aligned} \quad (4.9)$$

For the second case, choose $\theta_1 = 30.0^\circ$ and $\theta_2 = -55.0^\circ$. For this case, the robot Jacobian is given by

$$J_c = \begin{bmatrix} -0.1524 & -0.3849 \\ 0.2640 & -0.2695 \end{bmatrix}$$

and the actuator torques to maintain the same contact force are given by

$$T = \begin{bmatrix} 2.64 \\ -2.70 \end{bmatrix} \quad (4.10)$$

These are quite reasonable torque values when compared with the values specified in decoupled case. However, examining the motor torques shown in Figure 4.12, it is apparent that some saturation has occurred in motor 2. Note also, that the torques for motor 2 are significantly larger than the value obtained in equation 4.10.

To determine the cause of this deviation, it is necessary to examine the measured force generated along the surface: this force is shown in Figure 4.13. By inspection, this frictional force reaches values as high as 30 Newtons¹. The actuator torques required to maintain a force vector of $F = [30 \ 10]^T$ Newtons are

$$T = \begin{bmatrix} -1.93 \\ -14.24 \end{bmatrix}. \quad (4.11)$$

Consider what is happening physically: the actuators are *fighting* each other in order to maintain the contact dynamics. As actuator 2 saturates, its output is not sufficient to maintain the specified contact force while acting in opposition to actuator 1. Consequently, the contact force deviates from the specified trajectory.

This observation is important for three reasons:

1. The manipulator configuration can potentially place very strict limits on the attainable contact force.
2. These limits are difficult to predict, because frictional forces must also be considered.
3. Two large actuator torques are being subtracted to generate a smaller contact force: this has significant potential as a source of error in maintaining the specified contact force.

A comprehensive analysis of these effects is beyond the scope of this report. However, consider one figure of merit for a given manipulator configuration: the condition number of the Jacobian.

Refer again to the joint space to workspace transformation:

$$T = J^T(\theta)F. \quad (4.12)$$

This matrix equation is a linear set of equations for any given manipulator joint angles, θ . Consider only the problem of relating some force to some torque; it is

¹ This large frictional force is generated during contact with the compliant rubber padding. As the manipulator pushes against the environment, it deforms, leading to large frictional forces.

a. Position Trajectory

b. Force Trajectory

— Experimental Results - - - Desired Trajectory

Figure 4.11: Contact Maintenance Experiment

a. Motor 1 Torque

b. Motor 2 Torque

Figure 4.12: Contact Maintenance Experiment. Motor Torques

— Measured Contact Force

Figure 4.13: Contact Force Along the Surface

Figure 4.14: Robot Trajectory for Initial Contact Maintenance Experiment

plainly of interest to examine the sensitivity of the actuator torques to a change in the contact forces, and vice versa. Sensitivity of such a set of linear equations may be quantified through the condition number of the matrix [4]. For simplicity, consider the condition number based on the infinity norm of the Jacobian [4].

The definition of the infinity norm of some $n \times n$ matrix A is

$$\|A\|_{\infty} = \|A\| = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| \quad (4.13)$$

The condition number is then given by²

$$\text{cond}(A) = \|A\| \|A^{-1}\| \quad (4.14)$$

Consider the range of values that the condition number may achieve: for the identity matrix the condition number is one. For a sequence of matrices that are increasingly singular, the condition number becomes unbounded.

Plainly the greatest limits on attainable force trajectories will be obtained when the robot Jacobian is non-invertible; i.e. at the robot singularities. The least restricted case will occur when the Jacobian is diagonal, and there is no coupling between the joints in the joint space to workspace transformation. Thus the condition number of the robot Jacobian could be used as a figure of merit for the achievability of a force trajectory.

4.5 Contact Dynamics Experiments

The last segment of the experimental work was to perform a systematic investigation into contact stability results, bearing in mind the bounded input bounded output stability proposition of section 2.1.3.

This stability proposition states that the product of the bounds on the closed loop gain of the position controller in contact with the environment and the force feedback gains must be less than one to guarantee contact stability. It would be difficult to systematically vary the bound on the gain of the positioning controller in contact

²The 1 norm or the 2 norm could equivalently be used to obtain a similar condition number using equation 4.14. The infinity norm has been chosen only for convenience.

with the environment, as any precise variation would require the determination of the non-linear operators $G(e)$, $E(x)$, and $S(d)$ shown in figure 2.5. However, it is much more simple to vary the bound on the force feedback operator $H(f)$ as this operator has been implemented as a force feedback gains matrix. Consequently, by varying the target dynamics, the bound α_5 in equation 2.105 may be controlled.

Refer to the definition of the gains matrices in equations 3.20 and 3.40; for the linear controller the gains matrices are given by:

$$\begin{aligned} G_1 &= M(\theta_0)J_c^{-1}J^{-1}KJ_c \quad \text{Position Feedback} \\ G_2 &= M(\theta_0)J_c^{-1}J^{-1}CJ_c \quad \text{Velocity Feedback} \\ G_d &= M(\theta_0)J_c^{-1}J^{-1} - J_c^T \quad \text{Force Feedback} \end{aligned} \quad (4.15)$$

and the feedback gains matrices for the non-linear controller are

$$\begin{aligned} G_1 &= M(\theta)J_c^{-1}(\theta)J^{-1} - J_c^T \quad \text{Force Feedback} \\ G_3 &= M(\theta)J_c^{-1}(\theta)J^{-1}CJ_c(\theta) \quad \text{Velocity Feedback} \\ G_4 &= M(\theta)J_c^{-1}(\theta)J^{-1}KJ_c(\theta) \quad \text{Position Feedback} \end{aligned} \quad (4.16)$$

Observe that the force feedback gains can be varied while maintaining the same closed loop position controller by varying J^{-1} while holding $J^{-1}C$ and $J^{-1}K$ constant.

Using this technique, a total of four series of contact stability experiments were carried out. The stability of each controller in contact with both the stiff environment and the compliant environment was investigated for an identical range of target dynamics. The force feedback was varied throughout the target dynamics using the technique described above.

The contact experiments were carried out by moving the manipulator along a position trajectory along a wall, as shown in Figure 4.14. A force trajectory of 10 Newtons, pushing against the wall, was specified.

The results of the contact experiments are summarised in Tables 4.2 and 4.3. These results are summarised in terms of root mean square errors of the actual robot trajectory with respect to the specified robot trajectory. Recall that the RMS

error is defined by equation 3.41 as

$$E_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_{di})^2} \quad (4.17)$$

where

E_{rms} is the root mean square error;

n is the number of data points collected;

y_i is the measured data, in this case the measured workspace coordinates of the robot or the measured contact force, and

y_{id} is the desired response, in this case the specified position or force trajectory.

As the experiments were carried out, it became apparent that as the level of the force feedback gains was increased, the contact became unstable. As the level of force feedback gains was decreased, the controller exhibited poorer performance in tracking the specified force trajectory.

4.5.1 Unstable Contact

When the force feedback gains were relatively large, the contact rapidly became unstable. This contact instability was characterized by the robot manipulator *bouncing* on the wall. This instability could be quite violent.

Figure 4.15 shows experimental results for the non-linear controller in contact with the compliant rubber strip with the target dynamics:

$$J = \begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 2.00 \end{bmatrix}, C = \begin{bmatrix} 56.57 & 0.00 \\ 0.00 & 56.57 \end{bmatrix}, K = \begin{bmatrix} 400.00 & 0.00 \\ 0.00 & 40.00 \end{bmatrix}.$$

Figure 4.15-a shows the position trajectory along the wall. Qualitatively, the manipulator tracks the position trajectory in a satisfactory manner. Figure 4.15-b shows the robot motion normal to the wall³. In this experiment, the manipulator contacts the wall vigorously, and then 'bounces', losing contact with the surface. Figure 4.15-c shows the force normal to the wall. Each time that the robot bounced on the wall, a large impulsive force measurement was recorded.

³The desired position trajectory against the wall changes slowly as the wall was not lined up perfectly with the robot workspace axes.

				Linear Controller		Non-Linear Controller	
				RMS Position Error (mm)	RMS Force Error (Newtons)	RMS Position Error (mm)	RMS Force Error (Newtons)
X	2.00	56.57	400.00	** 7.17	25.80	** 20.05	23.52
Y	2.00	17.89	40.00				
X	5.00	141.42	1000.00	* 7.15	8.94	* 7.65	10.36
Y	5.00	44.72	100.00				
X	7.00	197.99	1400.00	8.40	4.34	5.50	4.87
Y	7.00	62.61	140.00				
X	10.00	282.84	2000.00	6.07	5.04	4.85	5.42
Y	10.00	89.44	200.00				
X	15.00	424.26	3000.00	5.66	7.10	5.05	9.57
Y	15.00	134.16	300.00				
X	20.00	565.68	4000.00	8.50	14.70	4.45	9.28
Y	20.00	178.88	400.00				
X	50.00	1414.20	10000.00	6.73	13.84	6.78	14.24
Y	50.00	447.20	1000.00				

* - Contact instability evident at some point(s) on the manipulator trajectory.

** - Contact instability evident throughout the manipulator trajectory.

Table 4.2: Contact Stability Results Stiff Environment

				Linear Controller		Non-Linear Controller	
				RMS Position Error (mm)	RMS Force Error (Newtons)	RMS Position Error (mm)	RMS Force Error (Newtons)
X	2.00	56.57	400.00	** 8.11	22.81	** 10.22	29.14
Y	2.00	17.89	40.00				
X	5.00	141.42	1000.00	* 3.98	13.30	* 4.99	17.88
Y	5.00	44.72	100.00				
X	7.00	197.99	1400.00	12.10	3.81	11.19	4.86
Y	7.00	62.61	140.00				
X	10.00	282.84	2000.00	10.18	4.09	10.58	3.84
Y	10.00	89.44	200.00				
X	15.00	424.26	3000.00	9.38	7.89	9.45	6.68
Y	15.00	134.16	300.00				
X	20.00	565.68	4000.00	8.79	9.82	8.46	12.19
Y	20.00	178.88	400.00				
X	50.00	1414.20	10000.00	7.62	10.56	5.91	11.40
Y	50.00	447.20	1000.00				

* - Contact instability evident at some point(s) on the manipulator trajectory.

** - Contact instability evident throughout the manipulator trajectory.

Table 4.3: Contact Stability Results - Compliant Environment

a) Position along the wall.

b) Position normal to the wall.

c) Normal contact force.

— Experimental Results - - - Specified Trajectory

Figure 4.15: Unstable Contact Example

Recall that Kazerooni's stability proposition shows that there will be some upper limit on the size of the force feedback gains for which the stability of the system will no longer be guaranteed. However, examine carefully the results shown in Tables 4.2 and 4.3. Experiments where contact became unstable are marked with a single or double asterisk (*). The contact became unstable for both controllers with the same target dynamics, regardless of whether the environment was the stiff steel plate or the more compliant rubber padding.

It has been observed in the literature that unstable contact can often be stabilized by adding compliance to the system [9]⁴. Compliance is usually added in the form of foam or rubber padding. As a result, it would be expected that the rubber padding should change the loop gain bound α_4 specified in equation 2.105 and consequently it might then reasonably be expected that the instability would occur at a different level of force feedback gains for the rubber padding and the steel plate. The results shown in Tables 4.2 and 4.3 indicate that there is little change in the contact stability with the environment. This observation does not indicate that the stability proposition is invalid; rather it indicates a weakness inherent to a sufficiency proof of stability. Hypothesize that, for either environment, the product $\alpha_4\alpha_5 > 1$, yet the system was stable. This condition is possible as the stability proposition is only a sufficiency condition for stability. Since $\alpha_4\alpha_5 > 1$, no conclusions may be drawn about the stability of the system. Under these circumstances, it is entirely plausible that instability would occur at a similar level of force feedback for both environments, as the input-output stability provides no information about the stability of the system.

Consequently, it would appear that the sufficiency condition for contact stability, while valid, is a conservative guideline. This input-output stability analysis does not serve as a detailed model of the contact dynamics.

⁴Such stabilized systems do not include force feedback loops.

4.5.2 Stable Contact

As the force feedback gains were decreased, the contact became stable, and the manipulator followed the specified trajectories reasonably. Experimental results for the non-linear controller in contact with the compliant rubber strip with target dynamics

$$J = \begin{bmatrix} 7.00 & 0.00 \\ 0.00 & 7.00 \end{bmatrix}, C = \begin{bmatrix} 197.99 & 0.00 \\ 0.00 & 62.61 \end{bmatrix}, K = \begin{bmatrix} 1400.00 & 0.00 \\ 0.00 & 140.00 \end{bmatrix}$$

are shown in Figure 4.16.

Plainly there exists some range of force feedback gains which provide both contact stability and good trajectory tracking. Examining the results shown in Tables 4.2 and 4.3, it is apparent that the force tracking error increased as the force feedback gains decreased. This effect is caused by the low level of force feedback; the feedback level becomes insufficient to effectively reject disturbances in the system.

2

of/de

2

PM-1 3 1/2"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQU /ALENT

PRECISIONSM RESOLUTION TARGETS

PIONEERS IN METHYLENE BLUE TESTING SINCE 1974

1980 COUNTY ROAD 5 BURNINGWELL NEW BEDFORD MASS
TEL: 617/426-7887 FAX: 617/426-7887 TUL: 5188888888

a) Position along the wall.

b) Position normal to the wall.

c) Normal contact force.

— Experimental Results - - - Specified Trajectory

Figure 4.16: Stable Contact Example

Chapter 5

Conclusions and Discussions

5.1 Conclusions

5.1.1 Controller Comparisons

In the simulations presented in Chapter 3, it was shown that both controllers performed reasonably in the situation where no position constraints were imposed on the robot manipulator. When position constraints were imposed, however, the behaviour of the linear controller showed significant degradation.

This particular result is of some importance; if the robot unexpectedly comes into contact with the environment, the linear controller will no longer maintain the target dynamics when the manipulator is moved from the point of linearization. This example illustrates a difficulty inherent in the use of linearized models. A controller based on such a linearized models assumes that the plant under control can be linearized about some operating point. If the plant is moved away from this operating point, the linearization may be lost. The designer of the controller must be aware of this limitation while carrying out the design.

When the two impedance controllers were implemented on the experimental hardware, it was determined that other limitations over-shadowed this loss of the operating point. In particular, the following two limitations were encountered:

- For stiff target dynamics, the robot motors saturated before significant deviation from the target dynamics was observed.
- As the target dynamics were made more compliant by increasing the force feedback gains, contact instability resulted.

These experimental results do not indicate that the loss of the operating point should be ignored. Instead, they indicate further factors that a designer must consider.

5.1.2 Force Trajectory Limitations

Further experiments in Chapter 4 indicated that there are very significant limitations that may be imposed on achievable force trajectories for a robot. These limits are imposed by the manipulator position, peak actuator outputs, and the surface with which the manipulator is to maintain contact.

When a manipulator is maintaining a contact force along some surface, there will be some finite frictional force generated in the direction opposing motion. Plainly this frictional force will be dependant on the nature of the surface and of the manipulator end effector. In the experiments performed, frictional forces prevented the manipulator from following the specified target dynamics.

Depending on the robot position, there may be a certain amount of cross coupling between actuators necessary in order to maintain some specified force trajectory. The degree of cross coupling may be quantified by the condition number of the robot Jacobian. The force trajectories specified were, in general, achievable for the robot manipulator operating in contact with a frictionless environment. When such contact friction was included, however, the actuators were required to “fight” each other in order to meet the specified target dynamics; relatively large actuator torques would be subtracted in order to maintain a relatively small contact force. When one actuator entered saturation, it could no longer generate sufficient torque to “fight” the other actuator, and deviations from the force trajectory occurred.

These results indicate that the controller designer now has further factors to consider:

- What subset of the robot workspace will be used for contact tasks? This subset will impose corresponding limitations on force trajectories.
- What are the frictional characteristics of the surfaces with which the robot will be interacting? How does the surface interact with the end effector?
- What are the resulting torque requirements of the actuators to be used in the robot?

5.1.3 Contact Stability Theory

The last set of experiments in Chapter 4 was directed towards a systematic verification of some of the contact stability research presented in the literature.

First note that the linear contact stability analyses proposed by Kazerooni [16] and Colgate [7] did not apply well to the experimental setup. According to these analyses, the specified target dynamics should have been stable in contact with any environment. Recall from Chapter 2 that the drawback to both of these analyses was the modelling of manipulator environment interactions as directly coupled linear systems. Plainly the contact dynamics observed became unstable as the level of force feedback was increased. Perhaps this instability is indicative that such a model of interactions is overly restrictive.

The more sophisticated bounded input bounded output stability analysis [19] did apply reasonably to the experimental setup. Recall that this analysis showed that contact stability could only be guaranteed for some bounded level of force feedback. As the force feedback gains were increased, the stability proposition would no longer hold, and stability of the system could not be guaranteed.

Even this input output method is of limited use, however. In order to estimate a bound on the force feedback, it is first necessary to characterize the environmental contact without force feedback. The stability proposition provides no means to guarantee stability without determining a number of characteristics of both the manipulator and the environment. It is not clear how to approach this characterization.

Furthermore, the input output stability proposition is a sufficient condition for stability. In order to achieve a high level of performance, it is desirable to determine

necessary conditions for stability. In short, the designer would ideally be able to analyze the entire system and then select gain margins much as would be done for a linear positioning controller. This factor is illustrated by the results of the contact dynamics experiments: instability occurred at the same level of force feedback for both the stiff and the compliant environment. It is well known [9] that unstable contact¹ can be stabilized by placing a compliant material between the manipulator and the contact surface. Consequently, it might be expected that the compliant environment would support stable contact with a greater level of force feedback than the stiff environment.

To summarize, it is desirable to further characterize the contact dynamics and the contact stability condition.

5.2 Future Research Directions

From this research, it is apparent that there are two distinct areas for further research into force feedback control for contact tasks; effective characterization of force trajectory limitations, and a more detailed understanding of contact dynamics.

5.2.1 Force Trajectory Limitations

In order to effectively perform contact tasks, it will be necessary to thoroughly analyze the manipulator limitations with respect to force trajectories. As described in Section 5.1.2, the factors that impose constraints on the achievable force trajectories are:

1. The manipulator position.
2. The manipulator end-effector.
3. The environment with which contact is to be maintained.
4. The peak actuator torques.

¹These contact experiments have no force feedback term.

5.2.2 Contact Dynamics Characterization

In order to maintain contact stability in a controlled robot, it is plainly of interest to continue to increase the understanding of contact dynamics and contact stability. Kazerooni [19, 20] has been publishing a number of articles dealing with further characterization of contact stability using his input output stability proposition.

In this thesis, it has been shown that contact dynamics are an inherently non-linear process. It would consequently be of interest to investigate the use of other non-linear analysis techniques for their application to contact stability.

Bibliography

- [1] An, C. and Hollerbach, J., *The Role of Dynamic Models in Cartesian Force Control of Manipulators*, The International Journal of Robotics Research, Vo. 8, No. 4, August 1989.
- [2] Anderson, R. and Spong, M., *Hybrid Impedance Control of Robotic Manipulators*, Proceedings of the 1987 IEEE International Conference on Robotics and Automation, March 31-April 3, 1987, Raleigh, North Carolina.
- [3] Astrom, K. and Wittenmark, B., *Adaptive Control*, Addison-Wesley Publishing Company, 1989.
- [4] Atkinson, K., *Elementary Numerical Analysis*, John Wiley and Sons, 1985.
- [5] Blauer, M. and Bélanger, P., *State and Parameter Estimation for Robotic Manipulators Using Force Measurements*, IEEE Transactions on Automatic Control, December 1987.
- [6] Borland International, *Turbo Pascal Reference Guide, Version 5.0*, 1800 Green Hills Road, Scotts Valley, California, 1989.
- [7] Colgate, E. and Hogan, N., *Robust control of dynamically interacting systems*, Int. J. Control, 1988, Vol. 48, No. 1, pp 65-88.
- [8] Colgate, E., *On the Intrinsic Limitations of Force Feedback Compliance Controllers*, Robotics Research — 1989, DSC-Vol. 14, presented at the Winter Annual Meeting of the ASME, December 10-15, 1989.

- [9] Craig, J., *Introduction to Robotics, Mechanics and Control*, second edition, Addison Welsy Publishing Company, 1989.
- [10] Desoer, C. and Vidyasagar, M., *Feedback Systems: Input-Output Properties*, Academic Press, 1975.
- [11] Goldenberg, A., *Implementation of Force and Impedance Control in Robot Manipulators*, Proceedings of the 1988 IEEE International Conference on Robotics and Automation, April 24-28, 1988, Philadelphia, Pennsylvania.
- [12] Hogan, N., *Impedance Control - An Approach to Manipulation, Parts I, II and III*, Transactions of the ASME, March 1985.
- [13] Hogan, N. and Colgate, E., *Stability Problems in Contact Tasks*, published in "The Robotics Review", O. Katib, J. Craig, T. Lozan-Perez, editors, MIT Press, 1988.
- [14] *Universal Force-Moment Sensor System Operation Manual, Rev. 2.0a - 5/88* JR3, Inc., 22 Harter Avenue, Woodland, CA 95695, 1988.
- [15] Kazerooni, H., *A Robust Design Method for Impedance Control of Constrained Dynamics Systems*, Ph.D. Thesis, Massachusetts Institute of Technology, 1985.
- [16] Kazerooni, H., Sheridan, T. and Houpt, P., *Robust Compliant Motion for Manipulators*, IEEE Journal of Robotics and Automation, June 1986.
- [17] Kazerooni, H., *Robust, Non-Linear Impedance Control for Robot Manipulators*, Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, North Carolina.
- [18] Kazerooni, H., Bausch, J. and Kramer, B., *An Approach to Automated Deburring by Robot Manipulators*, Transactions of the ASME, Vol. 108, December 1986.
- [19] Kazerooni, H., *On the Robot Compliant Motion Control*, Transactions of the ASME, Vol 111, September 1989.

- [20] Kazerooni, H., *Contact Instability of the Direct Drive Robot when Constrained by a Rigid Environment*, IEEE Transactions on Automatic Control, Vol. 35, No. 6, June 1990.
- [21] Mason, M., *Compliance and Force Control for Computer Controller Manipulators*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 11, No. 6, June 1981.
- [22] McCormick, W., *Control Software for the Direct Drive Robot*, Technical Report SCE-91-1, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada.
- [23] MCS #1400640, *SCM-150/300 Operation Manual*, Motion Control Systems, P.O. Box 2190, Christianburg, VA 24068.
- [24] Paul, R. and Shimano, B., *Compliance and Control*, Proceedings of the 1976 Joint Automatic Control Conference, West Lafayette, Indiana, July 27-30, 1976.
- [25] Polarikas, A. and Seely, S., *Signals and Systems*, PWS Publishers, Boston, MA, 1985.
- [26] Raibert, M. and Craig, J., *Hybrid Position/Force Control of Manipulators*, Transactions of the ASME, June 1981.
- [27] Rajguru, S., *Analysis and Design of a Direct Drive Robot*, Master's Thesis, Carleton University, Ottawa, Ontario, 1989.
- [28] Spong, M. and Vidyasagar, M., *Robot Dynamics and Control*, John Wiley and Sons, Inc., 1989.
- [29] Warshaw, G., *Investigations of Adaptive Control for a Direct Drive Robotic Manipulator*, Master's Thesis, Carleton University, Ottawa, Ontario, 1990.
- [30] Whitney, D., *Force Feedback Control of Manipulator Fine Motions*, Transactions of the ASME, June 1977.

- [31] Whitney, D., *Historical Perspective and State of the Art in Robot Force Control*,
The International Journal of Robotics Research, Vol. 6, No. 1, Spring 1987.

Appendix A

Small Gains Theorem

This appendix presents a statement and proof of the small gains theorem. This theorem is based on a more general form of the small gains theorem presented in [10].

Consider the system shown in Figure A.1. Here H_1 and H_2 are some operators (not necessarily linear). We state the small gains theorem based on this figure.

Theorem: Define u_1 and u_2 as

$$u_1 = e_1 + H_2(e_2) \quad (\text{A.1})$$

$$u_2 = e_2 - H_1(e_1). \quad (\text{A.2})$$

Let $u_1, u_2 \in L_p$.

If H_1 and H_2 are L_p stable, that is, if $e_1 \in L_p$ then $H_1(e_1) \in L_p$ and if $e_2 \in L_p$ then $H_2(e_2) \in L_p$, then there exists some $\beta_1, \beta_2, \alpha_1 \geq 0, \alpha_2 \geq 0$ such that

$$\|H_1(e_1)\|_p \leq \alpha_1 \|e_1\|_p + \beta_1 \quad (\text{A.3})$$

$$\|H_2(e_2)\|_p \leq \alpha_2 \|e_2\|_p + \beta_2. \quad (\text{A.4})$$

For the system shown in Figure A.1, if $\alpha_1 \alpha_2 < 1$ then

$$\|e_1\|_p \leq (1 - \alpha_1 \alpha_2)^{-1} (\|u_1\|_p + \alpha_2 \|u_2\|_p + \beta_2 + \alpha_2 \beta_1) \quad (\text{A.5})$$

$$\|e_2\|_p \leq (1 - \alpha_1 \alpha_2)^{-1} (\|u_2\|_p + \alpha_1 \|u_1\|_p + \beta_1 + \alpha_1 \beta_2) \quad (\text{A.6})$$

and $e_1, e_2, y_1, y_2 \in L_p$.

Figure A.1: Input Output Stability

Proof: From equation A.1,

$$e_1 = u_1 - H_2(e_2) \quad (\text{A.7})$$

then

$$\|e_1\|_p \leq \|u_1\|_p + \|H_2(e_2)\|_p \quad (\text{A.8})$$

$$\leq \|u_1\|_p + \alpha_2 \|e_2\|_p + \beta_2 \quad (\text{A.9})$$

Similarly, using equation A.2,

$$e_2 = u_2 + H_1(e_1) \quad (\text{A.10})$$

and, again,

$$\|e_2\|_p \leq \|u_2\|_p + \|H_1(e_1)\|_p \quad (\text{A.11})$$

$$\leq \|u_2\|_p + \alpha_1 \|e_1\|_p + \beta_1 \quad (\text{A.12})$$

Combining equations A.9 and A.12, and using the fact that $\alpha_2 \geq 0$ we have that

$$\|e_1\|_p \leq \|u_1\|_p + \alpha_2 (\|u_2\|_p + \alpha_1 \|e_1\|_p + \beta_1) + \beta_2. \quad (\text{A.13})$$

Expanding terms,

$$\|e_1\|_p \leq \|u_1\|_p + \alpha_2 \|u_2\|_p + \alpha_1 \alpha_2 \|e_1\|_p + \alpha_2 \beta_1 + \beta_2 \quad (\text{A.14})$$

Since $\alpha_1\alpha_2 < 1$,

$$(1 - \alpha_1\alpha_2)\|e_1\|_p \leq \|u_1\| + \alpha_2\|u_2\|_p + \alpha_2\beta_1 + \beta_2. \quad (\text{A.15})$$

Rearranging gives equation A.5:

$$\|e_1\|_p \leq (1 - \alpha_1\alpha_2)^{-1}(\|u_1\|_p + \alpha_2\|u_2\|_p + \alpha_2\beta_1 + \beta_2). \quad (\text{A.16})$$

Equation A.6 may be derived in a similar fashion.

Since $u_1, u_2 \in L_p$, then, from equations A.5 and A.6, $e_1, e_2 \in L_p$. Furthermore, since $y_1 = H_1(e_1)$ and $y_2 = H_2(e_2)$, we have that $y_1, y_2 \in L_p$ which completes the proof.

In order to apply the small gains theorem to the contact stability proposition of section 2.4.3, set $u_2 = 0$, $u_1 = r$, and $y_1 = f$ and the stability result follows directly.

Appendix B

Force Sensor Integration and Installation

This appendix describes the integration and installation of the six degree of freedom force sensor into Carleton's direct drive robot.

B.1 Physical

The *JR³* Universal Force Moment Sensor System, model number UFS-3012A50-U560 [14] was selected and purchased for Carleton's direct drive robot. This force sensor provides force and torque measurements in all 6 degrees of freedom using strain gauge technology.

The sensor system consists of the force sensor and an electronics chassis that provides signal conditioning and amplification. The final output of the sensor system consists of six analog voltage levels representing the force and torque along each degree of freedom.

The sensor load ratings are 50 lbs force and 150 in-lbs torque. (Load ratings along the axis of mounting are double these values.) Measurement resolution is limited by the 12 bit analog to digital converter used in Carleton's robot system and the system noise level.

The force sensor was mounted on the direct drive robot using two adaptor plates as shown in Figure B.1. A special end effector, consisting of a rigid finger, was

manufactured for use in compliant motion experiments. Drawings of these three parts are shown in Figures B.2, B.3, B.4 and B.5.

The force sensor was electrically isolated from the robot body using a nylon gasket and nylon mounting screws to bolt the UFS mounting plate on to the robot arm. This isolation was necessary due to the large surface currents generated in the robot body by the brushless DC motor actuators.

B.2 Electrical

The six output channels from the UFS electronics were connected to six differential input channels on the A/D converter already in use on the robot system as shown in Figure B.6. Hewlett-Packard power supplies were used to supply the ± 15 volts required by the UFS electronics.

The UFS electronics produces output signals in the range ± 10 volts, while the A/D converter accepts input signals in the range ± 5 volts. Voltage dividers made of two 1K precision resistors were used to scale the voltage range.

Use of differential input channels and careful grounding techniques minimized noise levels within the sensor system. Ad hoc experimental results indicate that the noise is limited to the four least significant bits on the A/D converter. This resolution corresponds to approximately 1.0 Newtons.

B.3 Calibration

The UFS arrives factory calibrated to a linear scale. However, it was necessary to estimate torque constants for the robot actuators and the robot mass properties, as this work had not been carried out previously. Previous experiments with the direct drive robot had used units of Volts (instead of Newton-metres) for the motor inputs, and no mass property estimation or measurement had been performed.

Figure B.1: UFS Mounting

Figure B.2: UFS Mounting Plate.

Figure B.3: UFS Adaptor Plate.

Figure B.4: UFS End Effector.

Figure B.5: UFS Mounting Hole Detail.

Figure B.6: UFS Electrical Connections

B.3.1 Motor Torque Constants

The robot joints were clamped into a rectangular configuration as shown in Figure B.7. A staircase was then applied to each actuator input separately, while the robot arm was prevented from moving. A least squares estimation¹ was performed in order to estimate preload offset levels and torque constants. The torque constants were determined to be:

$$k_{T1} = 1.95Nm/V \quad (B.1)$$

$$k_{T2} = 1.56Nm/V \quad (B.2)$$

(The offset levels have not been tabulated here as they change whenever the force sensor is mounted on the robot.)

A sample graph, showing the input commands and resultant joint torque for motor 1 is given as Figure B.8.

B.3.2 Sample Computer Source Code for Motor Torque Constant Data Collection

```
{ MOTOR1.PAS - Program that gets data to estimate motor 1
torque constant. }
```

```
program motor1;
```

```
{-----
```

```
Name: motor1
```

```
Purpose: motor1 should be used to calibrate the torque constant
       for motor1 on the robot. It produces a data file motor1.dat
       that should be fed to motor1.m ( a matlab source file ),
       which absorbs the measured data and does a least squares
```

¹A thorough discussion of least squares estimation techniques is presented in [3].

Figure B.7. Robot Configuration for Torque Constant Estimation

a. Measured Torque

b. Motor Input Voltage

Figure B.8: Motor 1 Torque Constant

estimate for the torque constant.

Here are some notes:

When running this program, the robot should be clamped to a rectangular configuration, so that the joints are effectively coupled, and motor1 sees only a moment arm.

The manipulator tip should be pushing against something solid, and preferably slightly compliant, to ensure contact stability.

If the robot arm moves more than arm_movement units on the digital data interface, the program aborts (to prevent the robot from smashing against anything).

}

uses dataacq, store, robcont;

const

pi = 3.1415926535897932;

freq = 100.0;	{ Sample frequency }
dt = 1.0 / freq;	{ and the timestep }
outputfile1 = 'motor1.dat	';
measurements = 4;	{ how many measurements to save }
iterations = 2000;	{ how many iterations. }
cyclesperstep = 100;	{ How many loops at each torque step. }

```

stepsize = 0.25;           { Voltage stepsize. }
start_save = 0;            { where to start saving
                             data. }
arm_movement = 0.2;        { amount of movement
                             permitted }

```

type

```

measurement_vector = record
    t: single;      { current time. }
    p1: single;     { joint positions }
    f1: single;     { measured forces }
    c1: single;     { control inputs }
end;

```

var

```

current: measurement_vector;
p: pointer;

d: datastore;

i, j: integer;      { Loop counter. }

p1, p2: single;     { Measured joint positions. }
oldp1, oldp2: single; { Previous measurements. }
t1, t2: single;     { Control inputs. }
fx, fz: single;     { Measured forces. }

j1, j2: single;     { Desired joint positions. }
dv1, dv2: single;   { Desired joint velocities. }

old_exit_handler: pointer;

```

```
status: boolean;    { Boolean variable used to check status. }
```

```
sat1, sat2: boolean; { More boolean status variables. }
```

```
{-----
```

Name: exit_handler

Purpose:

exit_handler is called when this program exits. It cleans up
and exits gracefully.

```
}
```

```
{$F+}
```

```
procedure exit_handler;
```

```
begin
```

```
    exitproc := old_exit_handler;
```

```
    disable_robot( status );
```

```
    save_datastore( d );
```

```
end;
```

```
{$F-}
```

```
{-----
```

Name: GetRawForce

Purpose: GetRawForce acts identically to GetForce, with
the exception that the measured forces are not converted
to the standard x-y frame of reference.

}

```
procedure GetRawForce( var fx, fz: single;  
                      var status: boolean );
```

```
const
```

```
    { Channel full scale values. }
```

```
    fxscale = 50.0;
```

```
    fzsacle = 100.0;
```

```
    { Channel offset voltage values. }
```

```
    fxoffset = 0.0;
```

```
    fzoffset = 0.0;
```

```
    lbs2newtons = 4.45;
```

```
    inlbs2newtonmetres = 0.113;
```

```
    ascale = 2.0;
```

```
    fx_channel = 2;    { A/D Channel to get fx on. }
```

```
    fz_channel = 4;    { A/D channel to get fy on. }
```

```
var
```

```
    fx1, fz1: single;
```

```
    overrun: boolean;
```

```
begin
```

```

    { Get the current set of measurements. }
    get_ad( fx_channel, fx1, overrun );
    status := not overrun;

    get_ad( fz_channel, fz1, overrun );
    status := status and ( not overrun );

    fx := fx1 * fxscale / ad_range * lbs2newtons;
    fz := fz1 * fzscale / ad_range * lbs2newtons;

    end;

{-----
}
begin

    old_exit_handler := exitproc;
    exitproc := @exit_handler;

    init_datastore( outputfile1, iterations, measurements,
                    d, status );
    if not status then begin
        writeln( 'Failed to initialize data store. ' );
        exit;
        end;
    p := @current;

    initialize_robot( freq, measureforces, p1, p2, status );
    if not status then begin
        writeln('Failed robot intialization.');
```



```

        exit;
    end;

waitfor( 5, status );
if not status then begin
    writeln(' User abort. ');
    exit;
end;

getjoint( oldp1, oldp2 );
getrawforce( fx, fz, status );

for i := 1 to round ( iterations / cyclesperstep ) do begin

    getjoint( p1, p2 );

    t1 := i * stepsize;
    t2 := 0.0;

    settorque( t1, t2, sat1, sat2, status );

    for j := 1 to cyclesperstep do begin

        getrawforce( fx, fz, status );
        if not status then begin
            writeln('Error, sampling rate too high. ');
            exit;
        end;

        current.t := dt * ( j + i * cyclesperstep );
    end;
end;

```

```

        current.p1 := p1;
        current.f1 := -fz;
        current.c1 := t1;
        addto_datastore( p, d, status );
        if not status then begin
            writeln('Failed to save data. ');
            exit;
            end;

        getjoint( p1, p2 );

        if ( abs( p1 - oldp1 ) > arm_movement ) or
            ( abs( p2 - oldp2 ) > arm_movement ) then begin
            writeln('Aborting, the robot''s moving. ');
            exit;
            end;

        end;

    end;

{ Let the exit handler clean up. }
end.

```

B.3.3 Sample Computer Source Code for Motor Torque Constant Estimation

% MOTOR1.M - Matlab source file to estimate motor 1 torque constant.

```
load motor1.dat
```

```

armLength = 0.3048;

tau1 = armLength * motor1(:,3);

v1 = motor1(:,4);

[ n, 1 ] = size(motor1);

one = ones(n,1);
sigmav1 = one' * v1;
sigmataul = one' * tau1;

a1 = [ v1'*v1 sigmav1; sigmav1 n ];
b1 = [ tau1'*v1 sigmataul ]';

thetahat1 = inv(a1) * b1

```

B.3.4 Robot Mass Properties

The robot mass properties were estimated using the linearized model:

$$T = M(\Theta_0)\ddot{\Theta}$$

Each motor was subjected to sinusoidal trajectories about various set points within the robot workspace. Using the linearized model and all the collected data, least squares estimation [3] was used to obtain an estimate of the robot mass matrix as:

$$M(\Theta) = \begin{bmatrix} 0.995 & 0.0012 \cos(\theta_1 - \theta_2) \\ 0.0012 \cos(\theta_1 - \theta_2) & 0.861 \end{bmatrix}$$

Refer to figure B.9. Consider a theoretical model for the manipulator with the following reasonable values:

$$l_1 = 0.3048 \text{ m}$$

Figure B.9: Five Bar Linkage Robot Model

$$l_{1c} = 0.1524 \text{ m}$$

$$m_1 = 4 \text{ kg}$$

$$I_1 = m_1 l_1^2 / 12 = 0.03096 \text{ kg} \cdot \text{m}^2$$

$$l_2 = 0.2540 \text{ m}$$

$$l_{2c} = 0.1270 \text{ m}$$

$$m_2 = 3 \text{ kg}$$

$$I_2 = m_2 l_2^2 / 12 = 0.01612 \text{ kg} \cdot \text{m}^2$$

$$l_3 = 0.3048 \text{ m}$$

$$l_{3c} = 0.1524 \text{ m}$$

$$m_3 = 4 \text{ kg}$$

$$I_3 = m_3 l_3^2 / 12 = 0.03096 \text{ kg} \cdot \text{m}^2$$

$$l_4 = 0.4556 \text{ m}$$

$$l_{4c} = 0.1500 \text{ m}$$

$$m_4 = 3 \text{ kg}$$

$$I_4 = 0.1 \text{ kg} \cdot \text{m}^2$$

The resulting theoretical mass matrix is given by [28]:

$$M_{Theoretical}(\Theta) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad (B.3)$$

where

$$\begin{aligned} m_{11} &= m_1 l_{c1}^2 + m_3 l_{c3}^2 + m_4 l_1^2 + I_1 + I_3 \\ m_{12} &= m_{21} = (m_3 l_2 l_{c3} - m_4 l_1 l_{c4}) \cos(\theta_2 - \theta_1) \\ m_{22} &= m_2 l_{c2}^2 + m_3 l_2^2 + m_4 l_{c4}^2 + I_2 + I_4. \end{aligned}$$

Substituting numerical values gives the result:

$$M_{Theoretical}(\theta) = \begin{bmatrix} 0.5265 & 0.01768 \cos(\theta_1 - \theta_2) \\ 0.01768 \cos(\theta_1 - \theta_2) & 0.4901 \end{bmatrix} \quad (B.4)$$

Thus the theoretical values compare quite reasonably to the experimentally determine mass properties.

B.3.5 Sample Computer Source Code for Robot Mass Properties Data Collection

{ GETMASS.PAS - Program that gets data to estimate the mass matrix. }

{-----}

Name: getmass

Purpose:

getmass gets some (persistently exciting) data about the robot's behaviour, by moving each joint in a sinusoidal trajectory about a set point.

This data is then to be used to model the robot as a second order system with no damping in order to obtain an estimate

```

    for the mass matrix using a least squares linear regression.
}

program getmass;

uses robcont, pd, store, traject, filter;

const
    pi = 3.1415926535897932;

    freq = 100.0;           { Sample frequency }
    dt = 1.0 / freq;        { and the timestep }
    outputfile1 = 'mass1.dat';
    outputfile2 = 'mass2.dat';
    outputfile3 = 'mass3.dat';
    measurements = 7;       { 7 measurements to save }
    iterations = 1100;      { run for 1000 iterations. }
    start_save = 100;       { and start saving data
                             after 100. }

    j1nought1 = pi / 6.0;   { offset for joint1 }
    j2nought1 = -pi / 6.0;  { offset for joint2 }
    j1nought2 = pi / 4.0;   { offset for joint1 }
    j2nought2 = -pi / 4.0;  { offset for joint2 }
    j1nought3 = pi / 3.0;   { offset for joint1 }
    j2nought3 = -pi / 3.0;  { offset for joint2 }
    j1freq = 1.5;           { frequency for joint 1 }
    j2freq = 1.0;           { frequency for joint 2 }
    j1amp = 0.1;            { amplitude for joint 1 }
    j2amp = 0.2;            { amplitude for joint 2 }

    { Kalman Observer Terms. }

```

```

a1 = 0.49;
b1 = 120.0;
a2 = 0.35;
b2 = 194.0;

```

```

{ PD feedback terms. }
kp1 = 200.0;
kv1 = 40.0;
kp2 = 200.0;
kv2 = 40.0;

```

type

```

measurement_vector = record
    t: single;      { current time. }
    p1, p2: single; { joint positions }
    v1, v2: single; { joint velocities }
    c1, c2: single; { control inputs }
end;

```

var

```

current: measurement_vector;
p: pointer;

d: datastore;

xk1, xk2: state;    { Kalman predictions for joints 1 and 2. }
kal1, kal2: state; { Kalman gains for joints 1 and 2. }
x1, x2: state;      { Kalman filter results for joints
                     1 and 2. }

t: sine;            { Trajectory variable used to generate the

```

joint trajectories. }

i: integer; { Loop counter. }

p1, p2: single; { Measured joint positions. }

v1, v2: single; { Measured joint velocities. }

c1, c2: single; { Control inputs. }

j1, j2: single; { Desired joint positions. }

dv1, dv2: single; { Desired joint velocities. }

old_exit_handler: pointer;

status, sat1, sat2: boolean;

{-----}

Name: exit_handler

Purpose:

exit_handler is called when this program exits. It cleans up
and exits gracefully.

}

{FF+}

procedure exit_handler;

begin

exitproc := old_exit_handler;


```

    disable_robot( status );
    save_datastore( d );

    end;
{$F-}

{-----}
}

begin

    old_exit_handler := exitproc;
    exitproc := @exit_handler;

    init_datastore( outputfile1, iterations, measurements,
                    d, status );
    if not status then begin
        writeln( 'Failed to initialize data store. ' );
        exit;
    end;
    p := @current;

    initialize_robot( freq, DontMeasureForces, p1, p2, status );
    if not status then begin
        writeln( 'Failed robot initialization.' );
        exit;
    end;

    kalman_gains( a1, b1, dt, kal1 );
    kalman_gains( a2, b2, dt, kal2 );

    writeln( ' Starting run in 5 seconds. ' );

```

```

WaitFor( 5, status );
if not status then begin
    writeln( 'User abort. ' );
    exit;
end;

```

```

init_sine( j1freq, j1amp, j1nought1,
           j2freq, j2amp, j2nought1,
           dt, t );

```

```

{-----
}

```

```

writeln( ' Starting run 1. ' );
gotojoint( j1nought1, j2nought1, status );
getjoint( p1, p2 );
if not status then begin
    writeln( 'Failed to position the manipulator.' );
    exit;
end;

```

```

getjoint( p1, p2 );
init_kalman( p1, 0.0, xk1 );
init_kalman( p2, 0.0, xk2 );
filter_states( p1, xk1, kal1, x1 );
filter_states( p2, xk2, kal2, x2 );

```

```

for i := 1 to iterations do begin

```

```

    step_sine( t, j1, j2, dv1, dv2 );

```

```

    run_pd( j1, j2, dv1, dv2, kp1, kv1, kp2, kv2,

```

```

        x1[1], x2[1], x1[2], x2[2], c1, c2 );

if i > start_save then begin
    current.t := ( i - 1 ) * dt;
    current.p1 := x1[1];
    current.p2 := x2[1];
    current.v1 := x1[2];
    current.v2 := x2[2];
    current.c1 := c1;
    current.c2 := c2;
    addto_datastore( p, d, status );
    if not status then begin
        writeln('Failed attempt to save data. ');
        exit;
        end;
    end;

SetTorque( c1, c2, sat1, sat2, status );

predict_next_states( x1, a1, b1, c1, dt, xk1 );
predict_next_states( x2, a2, b2, c2, dt, xk2 );

GetJoint( p1, p2 );
GetJointVelocities( v1, v2, status );
if not status then begin
    writeln(' Error, sampling frequency to high. ' );
    exit;
    end;

filter_states( p1, xk1, kal1, x1 );
filter_states( p2, xk2, kal2, x2 );

```

```

        end;

c1 := 0.0;
c2 := 0.0;
SetTorque( c1, c2, sat1, sat2, status );

save_datastore( d );

{-----
}

writeln( ' Starting run 2. ' );

clear_datastore( d );
set_filename( outputfile2, d );

init_sine( j1freq, j1amp, j1nought2,
           j2freq, j2amp, j2nought2,
           dt, t );

gotojoint( j1nought2, j2nought2, status );
getjoint( p1, p2 );
if not status then begin
    writeln( 'Failed to position the manipulator.' );
    exit;
end;

getjoint( p1, p2 );
init_kalman( p1, 0.0, xk1 );
init_kalman( p2, 0.0, xk2 );
filter_states( p1, xk1, kal1, x1 );

```

```

filter_states( p2, xk2, kal2, x2 );

for i := 1 to iterations do begin

    step_sine( t, j1, j2, dv1, dv2 );

    run_pd( j1, j2, dv1, dv2, kp1, kv1, kp2, kv2,
            x1[1], x2[1], x1[2], x2[2], c1, c2 );

    if i > start_save then begin
        current.t := ( i - 1 ) * dt;
        current.p1 := x1[1];
        current.p2 := x2[1];
        current.v1 := x1[2];
        current.v2 := x2[2];
        current.c1 := c1;
        current.c2 := c2;
        addto_datastore( p, d, status );
        if not status then begin
            writeln('Failed attempt to save data. ');
            exit;
            end;
        end;

    SetTorque( c1, c2, sat1, sat2, status );

    predict_next_states( x1, a1, b1, c1, dt, xk1 );
    predict_next_states( x2, a2, b2, c2, dt, xk2 );

    GetJoint( p1, p2 );
    GetJointVelocities( v1, v2, status );

```

```

        if not status then begin
            writeln(' Error, sampling frequency to high. ' );
            exit;
        end;

        filter_states( p1, xk1, kal1, x1 );
        filter_states( p2, xk2, kal2, x2 );

    end;

    c1 := 0.0;
    c2 := 0.0;
    SetTorque( c1, c2, sat1, sat2, status );

    save_datastore( d );

{-----
}

    writeln(' Starting run 3. ');

    clear_datastore( d );
    set_filename( outputfile3, d );

    init_sine( j1freq, j1amp, j1nought3,
               j2freq, j2amp, j2nought3,
               dt, t );

    gotojoint( j1nought3, j2nought3, status );
    getjoint( p1, p2 );
    if not status then begin
        writeln( 'Failed to position the manipulator.' );
    end;

```

```

    exit;
end;

getjoint( p1, p2 );
init_kalman( p1, 0.0, xk1 );
init_kalman( p2, 0.0, xk2 );
filter_states( p1, xk1, kal1, x1 );
filter_states( p2, xk2, kal2, x2 );

for i := 1 to iterations do begin

    step_sine( t, j1, j2, dv1, dv2 );

    run_pd( j1, j2, dv1, dv2, kp1, kv1, kp2, kv2,
            x1[1], x2[1], x1[2], x2[2], c1, c2 );

    if i > start_save then begin
        current.t := ( i - 1 ) * dt;
        current.p1 := x1[1];
        current.p2 := x2[1];
        current.v1 := x1[2];
        current.v2 := x2[2];
        current.c1 := c1;
        current.c2 := c2;
        addto_datastore( p, d, status );
        if not status then begin
            writeln('Failed attempt to save data. ');
            exit;
        end;
    end;
end;

```

```

    SetTorque( c1, c2, sat1, sat2, status );

    predict_next_states( x1, a1, b1, c1, dt, xk1 );
    predict_next_states( x2, a2, b2, c2, dt, xk2 );

    GetJoint( p1, p2 );
    GetJointVelocities( v1, v2, status );
    if not status then begin
        writeln( ' Error, sampling frequency to high. ' );
        exit;
    end;

    filter_states( p1, xk1, kal1, x1 );
    filter_states( p2, xk2, kal2, x2 );

    end;

{ Let the exit handler clean up. }

end.

```

B.3.6 Sample Computer Source Code for Robot Mass Properties Estimation

```

% MASS.M - Matlab file to do least squares estimates for the
% mass matrix.

```

```

disp('Starting first data set.')
disp('Loading data file.')

```

```

load mass1.dat

```



```
T = 0.02; % Sampling interval.
```

```
%-----
```

```
%
```

```
% See my notes for the meaning of all these variables.
```

```
% This is a home grown multivariable least squares method.
```

```
%
```

```
disp('Computing z values.')
```

```
[ 1, n ] = size( mass1 );
```

```
z = zeros(1,4);
```

```
z(1,:) = mass1(1,2:5);
```

```
z(2:1,1:4) = mass1(2:1,2:5) - mass1(1:1-1,2:5);
```

```
z(2:1,1:2) = z(2:1,1:2) - T * mass1(1:1-1,4:5);
```

```
disp('Computing x matrix.')
```

```
x = zeros(3,3);
```

```
y = zeros(3,1);
```

```
x(1,1) = ( T*T + 4 ) * mass1(1:1-1,6)' * mass1(1:1-1,6);
```

```
x(1,2) = ( T*T + 4 ) * mass1(1:1-1,6)' * mass1(1:1-1,7);
```

```
x(3,2) = x(1,2);
```

```
x(3,3) = ( T*T + 4 ) * mass1(1:1-1,7)' * mass1(1:1-1,7);
```

```
x(2,1) = x(1,2);
```

```
x(2,2) = x(1,1) + x(3,3);
```

```
x(2,3) = x(1,2);
```

```
y(1) = T * mass1(1:l-1,6)' * z(2:l,1);
```

```
y(1) = y(1) + 2 * mass1(1:l-1,6)' * z(2:l,3);
```

```
y(2) = T * mass1(1:l-1,7)' * z(2:l,1);
```

```
y(2) = y(2) + T * mass1(1:l-1,6)' * z(2:l,2);
```

```
y(2) = y(2) + 2 * mass1(1:l-1,7)' * z(2:l,3);
```

```
y(2) = y(2) + 2 * mass1(1:l-1,6)' * z(2:l,4);
```

```
y(3) = T * mass1(1:l-1,7)' * z(2:l,2);
```

```
y(3) = y(3) + 2 * mass1(1:l-1,7)' * z(2:l,4);
```

```
a = inv(x) * y;
```

```
m1 = 2 * a(1) / T;
```

```
m2 = 2 * a(2) / T;
```

```
m3 = 2 * a(3) / T;
```

```
m = inv( [ m1 m2; m2 m3 ] )
```

```
save mass1
```

```
%-----
```

```
disp('Starting second data set.')
```

```
disp('Loading data file.')
```

```
clear
```

```
load mass2.dat
```

```
T = 0.02; % Sampling interval.
```

```
%-----
```

```

%
% See my notes for the meaning of all these variables.
% This is a home grown multivariable least squares method.
%

disp('Computing z values.')

[ l, n ] = size( mass2 );

z = zeros(1,4);
z(1,:) = mass2(1,2:5);

z(2:1,1:4) = mass2(2:1,2:5) - mass2(1:1-1,2:5);
z(2:1,1:2) = z(2:1,1:2) - T * mass2(1:1-1,4:5);

disp('Computing x matrix.')

x = zeros(3,3);
y = zeros(3,1);

x(1,1) = ( T*T + 4 ) * mass2(1:1-1,6)' * mass2(1:1-1,6);
x(1,2) = ( T*T + 4 ) * mass2(1:1-1,6)' * mass2(1:1-1,7);

x(3,2) = x(1,2);
x(3,3) = ( T*T + 4 ) * mass2(1:1-1,7)' * mass2(1:1-1,7);

x(2,1) = x(1,2);
x(2,2) = x(1,1) + x(3,3);
x(2,3) = x(1,2);

y(1) = T * mass2(1:1-1,6)' * z(2:1,1);

```

```
y(1) = y(1) + 2 * mass2(1:1-1,6)' * z(2:1,3);
```

```
y(2) = T * mass2(1:1-1,7)' * z(2:1,1);
```

```
y(2) = y(2) + T * mass2(1:1-1,6)' * z(2:1,2);
```

```
y(2) = y(2) + 2 * mass2(1:1-1,7)' * z(2:1,3);
```

```
y(2) = y(2) + 2 * mass2(1:1-1,6)' * z(2:1,4);
```

```
y(3) = T * mass2(1:1-1,7)' * z(2:1,2);
```

```
y(3) = y(3) + 2 * mass2(1:1-1,7)' * z(2:1,4);
```

```
a = inv(x) * y;
```

```
m1 = 2 * a(1) / T;
```

```
m2 = 2 * a(2) / T;
```

```
m3 = 2 * a(3) / T;
```

```
m = inv( [ m1 m2; m2 m3 ] )
```

```
save mass2
```

```
%-----
```

```
disp('Starting third data set.')
```

```
disp('Loading data file.')
```

```
clear
```

```
load mass3.dat
```

```
T = 0.02; % Sampling interval.
```

```
%-----
```

```
%
```

```
% See my notes for the meaning of all these variables.
```

```
% This is a home grown multivariable least squares method.
```

```
%
```

```
disp('Computing z values.')
```

```
[ 1, n ] = size( mass3 );
```

```
z = zeros(1,4);
```

```
z(1,:) = mass3(1,2:5);
```

```
z(2:1,1:4) = mass3(2:1,2:5) - mass3(1:1-1,2:5);
```

```
z(2:1,1:2) = z(2:1,1:2) - T * mass3(1:1-1,4:5);
```

```
disp('Computing x matrix.')
```

```
x = zeros(3,3);
```

```
y = zeros(3,1);
```

```
x(1,1) = ( T*T + 4 ) * mass3(1:1-1,6)' * mass3(1:1-1,6);
```

```
x(1,2) = ( T*T + 4 ) * mass3(1:1-1,6)' * mass3(1:1-1,7);
```

```
x(3,2) = x(1,2);
```

```
x(3,3) = ( T*T + 4 ) * mass3(1:1-1,7)' * mass3(1:1-1,7);
```

```
x(2,1) = x(1,2);
```

```
x(2,2) = x(1,1) + x(3,3);
```

```
x(2,3) = x(1,2);
```

```
y(1) = T * mass3(1:1-1,6)' * z(2:1,1);
```

```
y(1) = y(1) + 2 * mass3(1:1-1,6)' * z(2:1,3);
```

```

y(2) = T * mass3(1:1-1,7)' * z(2:1,1);
y(2) = y(2) + T * mass3(1:1-1,6)' * z(2:1,2);
y(2) = y(2) + 2 * mass3(1:1-1,7)' * z(2:1,3);
y(2) = y(2) + 2 * mass3(1:1-1,6)' * z(2:1,4);

```

```

y(3) = T * mass3(1:1-1,7)' * z(2:1,2);
y(3) = y(3) + 2 * mass3(1:1-1,7)' * z(2:1,4);

```

```

a = inv(x) * y;

```

```

m1 = 2 * a(1) / T;
m2 = 2 * a(2) / T;
m3 = 2 * a(3) / T;

```

```

m = inv( [ m1 m2; m2 m3 ] )

```

```

save mass3

```

END

2|3|1 2|92|

FIN