

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

RICE UNIVERSITY

Control of Serial and Parallel Robots: Analysis and Implementation

by

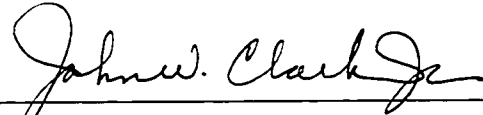
Ruvinda V. Gunawardana

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
Doctor of Philosophy

APPROVED, THESIS COMMITTEE:



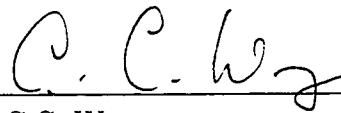
Dr. Fathi Ghorbel, Chairman,
Assistant Professor
Mechanical Engineering and Materials
Science



Dr. John Clark Jr.,
Professor
Electrical and Computer Engineering



Dr. Pol Spanos,
Lewis B. Ryon Professor of
Mechanical Engineering & Materials
Science



Dr. C.C. Wang,
Noah Harding Professor of
Mechanical Engineering & Materials
Science

Houston, Texas

October, 1998

UMI Number: 9928539

UMI Microform 9928539

Copyright 1999, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI

**300 North Zeeb Road
Ann Arbor, MI 48103**

Control of Serial and Parallel Robots: Analysis and Implementation

Ruvinda V. Gunawardana

Abstract

The research presented in this thesis is categorized into two areas. In the first part we address the issue of uniform boundedness of the elements of the equations of motion of serial robots, an important issue for the control of robots in this class. The second part is dedicated to the dynamic modeling and model based control of parallel robots. The field of serial robot control experienced tremendous growth over the past few decades resulting in a rigorous body of control results. An important assumption that is frequently made in establishing stability properties of these control laws is that the terms associated with the equations of motion of serial robots such as the inertia matrix, the Coriolis/centrifugal terms, and the Hessian of potential energy are uniformly bounded. This assumption however, is not valid for all serial robots. Since the stability conclusions of many control laws become local for robots that violate this assumption, it's important to be able to determine whether the terms in question are indeed uniformly bounded for a given robot. In the first part of this research we examine this issue and characterize the class of serial robots for which each of these terms are uniformly bounded. We also derive explicit uniform bounds for these terms which become important in control synthesis since the uniform bounds appear in the expressions of many control laws. The second part of this research is dedicated to parallel robots. Unlike in the case of serial robots, in parallel robots the independent generalized coordinates corresponding to the actuated joints do not uniquely determine the configuration of the robot. Therefore, an important issue that must be resolved in order to derive the dynamics of parallel robots is the existence of a transformation from the independent coordinates to a set of dependent coordinates that completely determine the robot configuration. The existence of such a transformation will enable the extension of most results in serial robots to parallel robots. In this research we characterize a region with specified boundaries where such a transformation exists and derive a numerical scheme for implementing the transformation in real time. Another contribution of this research is the design and construction of the Rice Planar Delta Robot which will serve as a test bed for results on parallel robots. This robot was used to experimentally verify the above result in a trajectory tracking experiment and a fast pick and place experiment.

To my wife Asini

and my parents

Jinadasa and Lilian Gunawardana
for their constant support and love.

Acknowledgments

I would first like to thank my advisor, Dr. Fathi Ghorbel, for his support during the course of my research. I would like to thank Dr. J. W. Clark, Jr., Dr. P. D. Spanos, and Dr. C. C. Wang for serving on my thesis committee and for their valuable suggestions. I would also like to thank Mr. Richard Massey and Dr. Joel Cyprus of the Electrical Engineering Department for their assistance during this project.

I would also like to thank my colleagues Athanasios Athanasiades, Scott Hejny, Kevin Magee, Jon Olansen, Xueqing Sun, and Muhammed Were for their support during the course of this research. Their suggestions and friendship I will never forget.

I would like to thank my parents, sisters, and brothers for their encouragement, guidance, and support throughout my academic career. Finally, I would like to thank my wife, Asini, for her love and constant support which made all of this possible.

Table of Contents

Abstract	ii
Acknowledgments	iv
List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 The Uniform Boundedness Property of Serial Robots	3
1.2 Model Based Control of Parallel Robots	9
2 The Uniform Boundedness of the Coriolis/Centrifugal Terms	19
2.1 Kinematics Review	19
2.2 A Uniform Bound for the $C(\mathbf{q}, \dot{\mathbf{q}})$ Matrix	27
2.3 Characterization of the Class of Robots for Which the $C(\dot{\mathbf{q}}, \mathbf{q})$ Matrix is Uniformly Bounded	31
2.4 Computing the Uniform Bound γ : Illustrative Example	34
2.5 Summary	37
3 The Uniform Boundedness of the Hessian of Potential Energy	40
3.1 The Hessian of Potential Energy	41
3.2 Characterization of the Class of Robots with Uniformly Bounded Hessian (Class \mathcal{BGJ})	43
3.2.1 Preliminary Results	44
3.2.2 Properties of the Matrices B and $U_i, i = 1, \dots, n$	48

3.2.3	Class \mathcal{BGJ} Robots	57
3.2.4	Necessary and Sufficient Conditions For the Hessian to be Uniformly Bounded	58
3.2.5	Examples	62
3.3	A Uniform Bound for Class \mathcal{BGJ} Robots	66
3.3.1	An Explicit Expression for the Uniform Bound β Satisfying (1.2)	66
3.3.2	Algorithm for Computing the Uniform Bound for the Hessian	67
3.3.3	Example: PUMA560	68
3.4	A Bound for Class $\overline{\mathcal{BGJ}}$ Robots	71
3.4.1	An Explicit Expression for the Bound c_4 satisfying (1.4) . . .	72
3.4.2	Example: The Bound c_4 for the \mathcal{RP} Robot of Section 3.2.5.1 .	73
3.5	Summary	76
4	Implementing PD control with Simple Gravity Compensation in Parallel Robots for Set Point Tracking Applications	77
4.1	PD Control of Closed-Chain Mechanisms	77
4.1.1	The Reduced Model for Parallel Robots	78
4.1.2	PD Plus Simple Gravity Compensation	81
4.2	The Rice Planar Delta Robot (R.P.D.R.)	82
4.2.1	Equations of Motion of the R.P.D.R.	86
4.3	Implementation of the PD Control Law	89
4.3.1	Evaluation of the Parameters in the Equations of Motion . . .	91
4.3.2	Experimental Results and Simulations: Point to Point Motion	92
4.3.3	Experimental Results and Simulations: Repetitive Point to Point Motion	94
4.4	Summary	95

5	Model Based Control of Parallel Robots for Trajectory Tracking Applications	98
5.1	The Domain of the Parameterization $\mathbf{q}' = \sigma(\mathbf{q})$	98
5.2	Extending the domain to an arbitrary trajectory	109
5.3	The Real Time Computation of $\mathbf{q}' = \sigma(\mathbf{q})$	111
5.4	Conditions for Convergence	112
5.5	Procedure for Implementing Control in Parallel Robots	114
5.6	Trajectory Tracking Experiment	115
5.7	Implementation of the Procedure Defined in Section 5.5	115
5.8	Experimental Results	118
5.9	Summary	120
6	Conclusions and Future Recommendations	125
6.1	Control of Serial Robots	125
6.2	Control of Parallel Robots	126
6.3	Summary	128
	Bibliography	130
A	The Design of the Rice Planar Delta Robot	140
A.1	Introduction	140
A.2	The Robot and the Base	142
A.3	Circuit Box and Power Supply	148
A.4	The Computer, dSPACE Board and Software	149

List of Figures

1.1	A Parallel Robot and a Serial Robot	2
1.2	A revolute-prismatic (2 d.o.f.) Serial Robot	5
1.3	Free System, Constraints, and Constrained System	14
2.1	The Center of Mass Location+Modified DH Parameters	20
2.2	A 2 d.o.f \mathcal{RP} of Class \mathcal{BD}	35
2.3	A 2 d.o.f \mathcal{RR} Robot	36
2.4	The Uniform Bound γ and $\ C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\ $ as a Function of q_2 for the \mathcal{RR} Robot of Figure 2.3 with $\ \dot{\mathbf{q}}\ = 1$	38
3.1	Partitioning of \mathbf{v}_c in Coordinate Frame j	56
3.2	The \mathcal{RP} Robot of Example 3.2.5.1	63
3.3	The \mathcal{RP} Robot of Example 3.2.5.2	65
3.4	The \mathcal{RP} Robot of Example 3.2.5.3	67
3.5	The PUMA560 Robot	69
3.6	Uniform Bound β (dashed line) and $\left\ \frac{\partial \mathbf{g}}{\partial \mathbf{q}}\right\ $ (solid curve) for the PUMA560	72
3.7	Uniform Bound β and $\left\ \frac{\partial \mathbf{g}}{\partial \mathbf{q}}\right\ $ for the \mathcal{RP} Robot	75
4.1	Link and Joint Configuration of the R.P.D.R.	83

4.2	Rice Planar Delta Robot (R.P.D.R.)	84
4.3	The “free system”	86
4.4	Singular Configurations of the R.P.D.R. and the Region \mathcal{D}	90
4.5	Point-To-Point Trajectory	92
4.6	Simulation (Dashed) and Experimental (Solid) Results	96
4.7	Pick and Place Routine: Simulation (Dashed) and Experimental (Solid) Results	97
5.1	Two different configurations of the planar delta corresponding to the same positions of the actuated joints (joints 1 and 2)	99
5.2	The trajectory in \mathcal{D} where control is to be implemented	110
5.3	The Trajectory to be Followed by the RPDR	115
5.4	The Joint Velocities and the Corresponding Magnitude of the Velocity Vector for the Selected Motion.	117
5.5	The Response of the R.P.D.R. When The Initiation Frequency was Set at 1 KHz. ($T_i = 1$ millisecond).	121
5.6	The Response of the R.P.D.R. When the Initiation Frequency was Set at 100 Hz. ($T_i = 10$ millisecond).	122
5.7	The Response of the R.P.D.R. When the Initiation Frequency was Set at 10 Hz. ($T_i = 100$ millisecond).	123
A.1	Joint and Link Configuration of the Rice Planar Delta Robot	143
A.2	The Rice Planar Delta Robot-Full View	144
A.3	The Actuator Unit	145
A.4	The Transmission	146
A.5	The Circuit Box	148

A.6 The Interior View of the Circuit Box	149
A.7 The Computer	150

List of Tables

2.1	Notation for Kinematics Parameters	21
3.1	Kinematic Link Parameters of \mathcal{RP} Robot of Figure 3.2	64
3.2	Kinematic Link Parameters of \mathcal{RP} Robot of Figure 3.2	64
3.3	Kinematic Link Parameters of the PUMA560	70
4.1	Kinematic Link Parameters of the Rice Planar Delta Robot	91

Chapter 1

Introduction

In most robots that are popular today, the links are connected sequentially starting from the base to the end effector or the gripper. Each link is connected to the previous link in the link chain by a revolute joint (permits relative rotation) or a prismatic joint (permits relative translation) and all of the joints are actuated. We refer to this type of robot as a “serial robot” to distinguish them from another class called “parallel robots”. In a parallel robot the links are connected in series and parallel combinations and only some of the joints are actuated. Figure 1.1 shows an example of a parallel robot and a serial robot. Parallel robots offer advantages over traditional serial robots such as faster acceleration and greater rigidity at the end effector.

Control of serial robots is an area of research that has received much attention over the past two decades resulting in a comprehensive body of control results. While initial results were developed assuming ideal conditions such as perfectly rigid links, ideal actuators, and no friction, as the research in this area became more mature control strategies that account for these effects were developed. In establishing the stability of many of these control laws a commonly made assumption is that the terms arising in the equations of motion of serial robots are uniformly bounded. This assumption however, is not valid for all serial robots. Since the uniform boundedness is necessary in order to establish global stability, for those robots that don't satisfy uniform boundedness the stability conclusions of many control laws become local. In the first part of this research we will re-examine the validity of the uniform boundedness assumption and its consequences in control law synthesis.

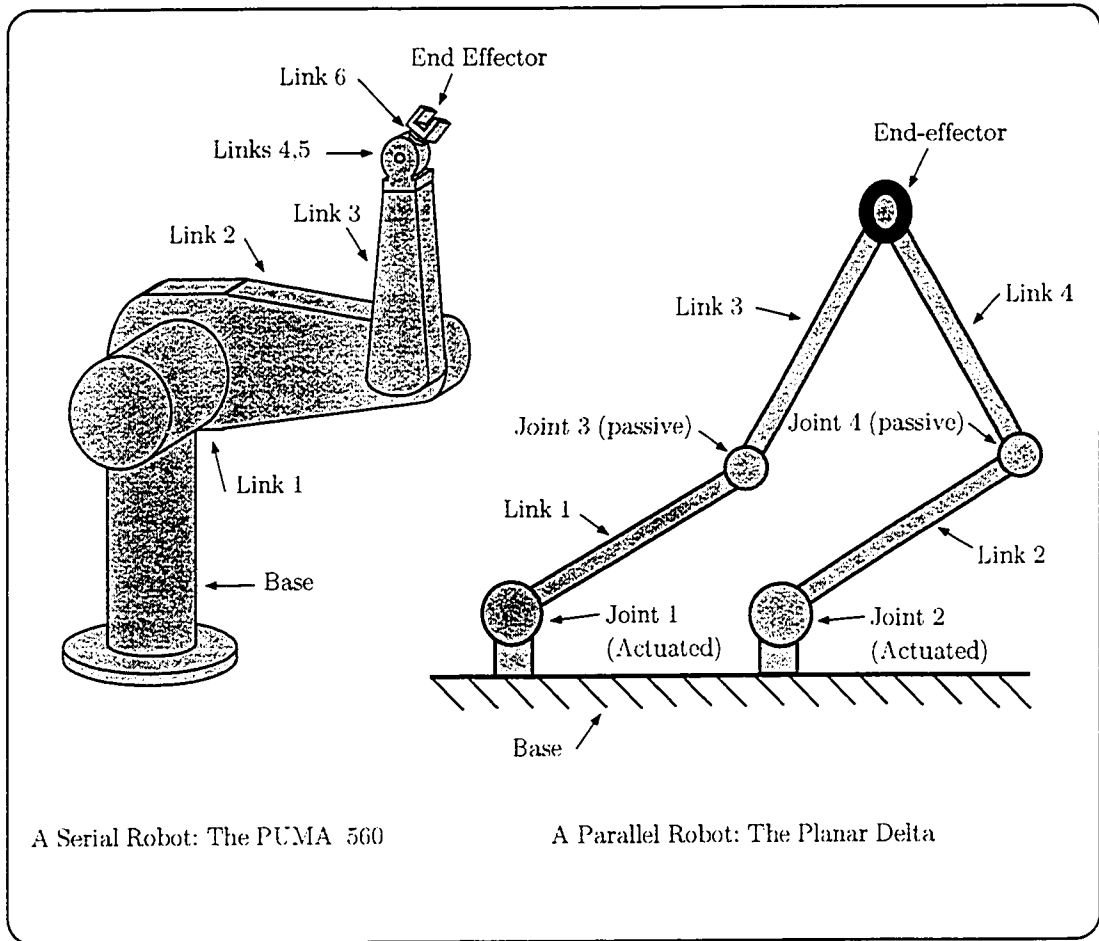


Figure 1.1: A Parallel Robot and a Serial Robot

In the second part of this research we will concentrate on parallel robots. Since parallel robots are a more recent development, control of parallel robots has received relatively less attention. Furthermore, due to the complexities involved with kinematic and dynamic formulation, designing controllers for parallel robots is more challenging. Due to these reasons, control of parallel robots is still an open research problem. An interesting question is whether the wealth of control results that already exists for serial robots can be extended to parallel robots. Since the structure

and properties of the equations of motion of serial robots have been exploited in control, in order to extend these results to parallel robots the equations of motion of parallel robots must be derived such that they retain the same structure as the equations of motion of serial robots. In the second part of this research we will examine some important issues related to this problem.

1.1 The Uniform Boundedness Property of Serial Robots

The equations of motion of a serial robot assuming perfectly rigid links, rigid transmissions, and zero friction have the following structure [66]:

$$D(\mathbf{q})\ddot{\mathbf{q}} + C(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \quad (1.1)$$

where the n -vectors \mathbf{q} , $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ represent the joint variables, velocities and accelerations respectively, $D(\mathbf{q})$ is the $n \times n$ inertia matrix, $C(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}}$ represents the Coriolis/centrifugal terms, $\mathbf{g}(\mathbf{q})$ is the gravity vector, and \mathbf{u} is the vector of generalized input forces/torques. Even though (1.1) represents a highly coupled nonlinear system of equations, it possesses some very important properties. These properties enabled researchers to achieve considerable success in designing control strategies. The following are some of the properties that are commonly exploited in control:

- 1.1. The equations of motion can be written in the form of (1.1) where there is an independent input corresponding to each degree of freedom.
- 1.2. The matrix $\dot{D} - 2C$ is skew symmetric.
- 1.3. The inertia matrix $D(\mathbf{q})$ is symmetric and positive definite.
- 1.4. We can define a vector θ representing all of the unknown mass, length and inertia parameters such that the equations of motion (1.1) is linear in θ . That

is, we can write

$$D(\mathbf{q})\ddot{\mathbf{q}} + C(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\theta$$

where $Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is a matrix that is dependent only on the joint variables and their derivatives.

The control strategies that were developed over the past two decades include PD and PID based control, passivity based control, adaptive control, and robust control [51]. In establishing the stability properties of these control laws, in addition to properties 1.1-1.4 it's also commonly assumed that the matrices $D(\mathbf{q})$, $C(\dot{\mathbf{q}}, \mathbf{q})$, and $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ (the Hessian of potential energy, which will be referred to as the Hessian in this thesis) are uniformly bounded. We refer to a matrix $M(\mathbf{q})$ as being uniformly bounded when there exists a constant c independent of \mathbf{q} , such that¹ $\|M(\mathbf{q})\| \leq c$ for all $\mathbf{q} \in \mathbb{R}^n$. That is, in the stability analysis of many control laws it's assumed that constants σ_1 , σ_2 , γ , and β exist such that²

$$\|D(\mathbf{q})\|_{\min} > \sigma_1, \quad \|D(\mathbf{q})\|_{\max} < \sigma_2, \quad \left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq \beta, \quad \forall \mathbf{q} \in \mathbb{R}^n, \quad (1.2)$$

and

$$\|C(\dot{\mathbf{q}}, \mathbf{q})\| \leq \gamma \|\dot{\mathbf{q}}\|, \quad \forall \mathbf{q} \in \mathbb{R}^n. \quad (1.3)$$

Note that since the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is a function of both $\dot{\mathbf{q}}$ and \mathbf{q} , it can be uniformly bounded with respect to either of these variables. For control, the uniform boundedness of interest is that with respect to \mathbf{q} .

¹In this thesis, we use the following standard notation and terminology: \mathbb{R} denotes the set of real numbers, and \mathbb{R}^n denotes the usual n -dimensional vector space over \mathbb{R} endowed with the Euclidean norm $\|\mathbf{x}\| = \{\sum_{i=1}^n x_i^2\}^{\frac{1}{2}}$. $\mathbb{R}^{n \times m}$ denotes the set of all $n \times m$ matrices with real elements. Unless otherwise specified, for $M \in \mathbb{R}^{n \times n}$, $\|M\|$ is the induced-2 matrix norm of M corresponding to the Euclidean vector norm on \mathbb{R}^n .

²The max-bound of a PSD matrix M is a mapping $\|\cdot\| : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^+$ defined by $\|M\|_{\max} = \max_{\mathbf{x}^T \mathbf{x} = 1, \mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T M \mathbf{x}$. The min-bound of a PSD matrix M is a mapping $\|\cdot\| : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^+$ defined by $\|M\|_{\min} = \min_{\mathbf{x}^T \mathbf{x} = 1, \mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T M \mathbf{x}$.

It's known that the inertia matrix, the Hessian, and the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix are all uniformly bounded for serial robots with joints that are all revolute [51]. They are not, however, uniformly bounded for any general serial robot. For example, consider the revolute-prismatic (\mathcal{RP}) robot of Figure 1.2 for which the terms in question are

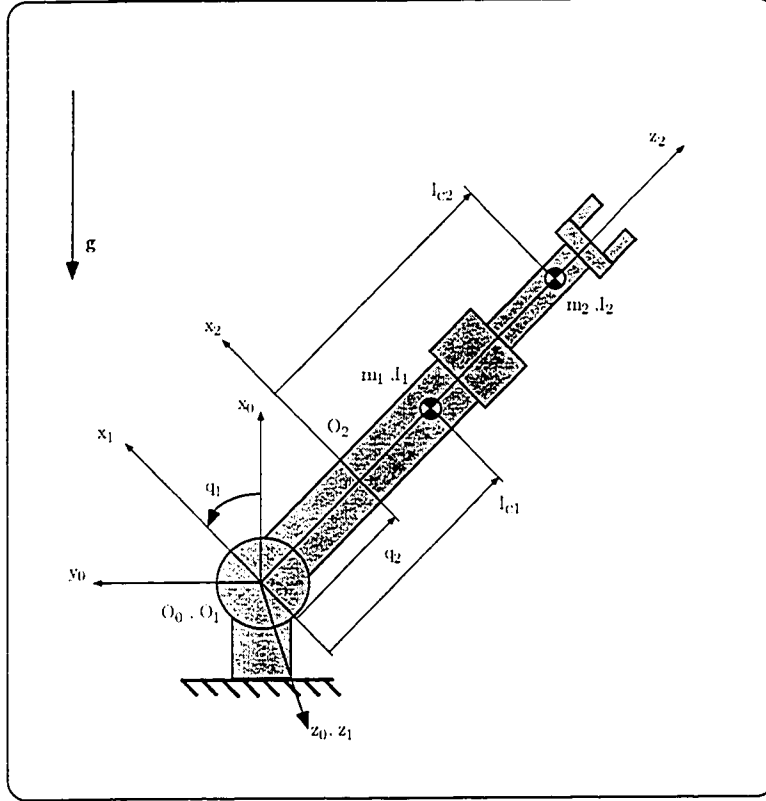


Figure 1.2: A revolute-prismatic (2 d.o.f.) Serial Robot

given by,

$$D(\mathbf{q}) = \begin{bmatrix} d_{1,1} & 0 \\ 0 & m_2 \end{bmatrix}, \quad C(\dot{\mathbf{q}}, \mathbf{q}) = \begin{bmatrix} \dot{q}_2 m_2 (l_{c2} + q_2) & \dot{q}_1 m_2 (l_{c2} + q_2) \\ -\dot{q}_1 m_2 (l_{c2} + q_2) & 0 \end{bmatrix}$$

$$\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} -m_1 l_{c1} g \sin(q_1) - m_2 (q_2 + l_{c2}) g \sin(q_1) & m_2 g \cos q_1 \\ m_2 g \cos q_1 & 0 \end{bmatrix},$$

where $d_{1,1} = m_1 l_{c1}^2 + m_2(q_2 + l_{c2})^2 + I_1 + I_2$. It can be seen that $d_{1,1}$ can be made arbitrarily large by selecting q_2 sufficiently large. Therefore, a constant σ_2 (independent of \mathbf{q}) satisfying (1.2) does not exist for this robot. Thus the inertia matrix is not uniformly bounded. It can be seen that the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix and the Hessian are also not uniformly bounded for this robot for the same reason.

At this point a reader might question as to why the uniform boundedness is important since all of the terms in question are bounded within the workspace of the robot. That is, for any given robot it's always possible to find constant bounds, c_1, c_2, c_3 , and c_4 such that

$$\|D(\mathbf{q})\|_{\min} > c_1, \quad \|D(\mathbf{q})\|_{\max} < c_2, \quad \|C(\dot{\mathbf{q}}, \mathbf{q})\| \leq c_3 \|\dot{\mathbf{q}}\|, \quad \left\| \frac{\partial g(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq c_4, \quad \forall \mathbf{q} \in W, \quad (1.4)$$

where W is the workspace of the robot defined as follows:

$$W = \left\{ \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_i \\ \vdots \\ q_n \end{bmatrix} \in \mathbb{R}^n : \begin{cases} q_i \in \mathbb{R} & \text{if joint } i \text{ is revolute} \\ |q_i| \leq \bar{q}_i & \text{if joint } i \text{ is prismatic} \end{cases} \right\}.$$

Note that \bar{q}_i represents the physical extension limits of prismatic joints. The importance of uniform boundedness in (1.2)-(1.3) versus the boundedness in (1.4) above lie in the approach that is taken in establishing the stability of many control laws. We will consider, for example, the adaptive PD controller of Tomei [70] to illustrate this point.

Let's first consider the background out of which the work of [70] arose. It was well established before this work that asymptotic stability and zero steady state error for the set point tracking problem can be achieved with PD control, provided the gravity vector can be compensated exactly [66]. In implementation, it was difficult to

compute the gravity vector exactly due to uncertainties in measurements. In the work of [70], the linearity in the parameters of the robot equations of motion (Property 1.4) is exploited to adaptively improve the estimate of the gravity vector. The control law is given by

$$\mathbf{u} = -K_P(\mathbf{q} - \mathbf{q}^d) - K_D\dot{\mathbf{q}} + E(\mathbf{q})\mathbf{p},$$

where \mathbf{q}^d is the desired position, the matrices K_P and K_D are the PD gains, and $E(\mathbf{q})\mathbf{p} = \mathbf{g}(\mathbf{q})$ is the estimate of the gravity vector. The vector \mathbf{p} represents the link parameters and the matrix $E(\mathbf{q})$ is the regressor. The matrix K_D must be selected positive definite and the matrix K_P must be selected such that $\left[\frac{\partial g(\mathbf{q})}{\partial \mathbf{q}} + K_P\right]$ is positive definite. This can be achieved by selecting K_P to be diagonal with each entry k_{pi} satisfying

$$k_{pi} > \beta, \quad (1.5)$$

where $\left\|\frac{\partial g(\mathbf{q})}{\partial \mathbf{q}}\right\| \leq \beta$. The parameter adaptation law is given by

$$\dot{\mathbf{p}} = -\omega E^T(\mathbf{q}) \left[\delta \dot{\mathbf{q}} + \frac{2(\mathbf{q} - \mathbf{q}^d)}{1 + (\mathbf{q} - \mathbf{q}^d)^T(\mathbf{q} - \mathbf{q}^d)} \right],$$

where ω is a positive constant, and δ must be selected such that³

$$\delta > \max \left\{ \frac{2\sigma_2}{\sqrt{\sigma_1 \lambda_m(K_P)}}, \frac{1}{\lambda_m(K_D)} \left[\frac{\lambda_M(K_D)^2}{2\lambda_m(K_P)} + 4\sigma_2 + \frac{\gamma}{\sqrt{2}} \right] \right\}. \quad (1.6)$$

The notation $\lambda_m(A)$ and $\lambda_M(A)$ is used to denote respectively, the smallest and largest eigenvalues of a matrix A . The stability properties are established using Lyapunov stability analysis. The global positive definiteness of the Lyapunov function V and the global negative semidefiniteness of \dot{V} is established provided (1.5) and (1.6) are satisfied for all $\mathbf{q} \in \mathbb{R}^n$. It can be seen that this can only be achieved if the inertia

³We have ignored coulomb friction even though it's included in [70] since it's not important in our discussion.

matrix, the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix, and the Hessian are all uniformly bounded since the uniform bounds σ_1 , σ_2 , β , and γ of these terms are used in (1.5) and (1.6). If these terms are not uniformly bounded and if the bounds c_1, c_2, c_3 , and c_4 of (1.4) are used instead of the uniform bounds σ_1 , σ_2 , β , and γ of (1.2) and (1.3), only the local positive definiteness of the Lyapunov function V and the local negative semidefiniteness of \dot{V} can be established. Hence, when the uniform boundedness assumption is not satisfied, the stability conclusions becomes local. In this case, even though (1.4) is valid throughout the robot workspace W , the stability conclusion will not, in general, be valid throughout W . Furthermore, it's very difficult to characterize the valid domain of a local stability result. Hence, a global stability result is always more useful than a local stability result.

Therefore, it can be seen that in the implementation of many control laws it's necessary to know the class of robots for which the terms in question are uniformly bounded. In [31], the class of robots for which the inertia matrix is uniformly bounded was characterized. This class was named class \mathcal{BD} (which stands for \mathcal{B} ounded inertia matrix \mathcal{D}). In addition to characterizing class \mathcal{BD} , explicit expressions for uniform bounds σ_1 and σ_2 satisfying (1.2) were also presented in [31]. The explicit expressions of the uniform bounds are important in controller synthesis since the uniform bounds appear in the control law expression. In Chapter 2 we extend this work and characterize the class of robots for which the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is uniformly bounded and derive an expression for the uniform bound γ satisfying (1.3). The uniform boundedness of the Hessian is addressed in Chapter 3 where we characterize the class of robots for which the Hessian is uniformly bounded and derive an expression for the uniform bound β satisfying (1.2). In addition we will also derive a bound c_4 satisfying (1.4) for robots for which a uniform bound β satisfying (1.2) doesn't exist.

This work will be important in the implementation of many control laws that exploit the uniform boundedness property. These include the PD plus simple gravity compensation controller of [68], the PD based controller of [44], the learning gravity compensators of [15], the PI²D regulator of [59], the robust control law of [63], the robust adaptive controllers of [8] and [18], the controllers for flexible joint robots reported in [30], [28], [29], [27], [1], [71], [43], [57], and [62], the adaptive control laws of [67], [70], and [65], the control laws based on the energy Lyapunov function approach studied in [78], and the passivity based controller-observer of [9].

In summary, this work will clarify and extend our current understanding of an important property of serial robots, namely, the uniform boundedness property which is essential to establish global stability of many control laws. This work will enable one to determine simply by inspecting the joint configuration of a given robot whether the uniform boundedness is satisfied and thereby determine whether a large number of control laws will retain globality. The explicit expressions for the uniform bounds γ and β will be extremely important in controller synthesis since the explicit values of the uniform bounds are necessary to compute the control law expression.

1.2 Model Based Control of Parallel Robots

During the past few years, results on control of parallel robots have begun to appear in the literature. Many of the methods used in serial robot control have been proposed for parallel robots also. These include PID control [2], adaptive control [74], robust control [12], optimal control [48], learning control [46], hybrid force/position control [64], and artificial intelligence based methods [5], [20]. Unlike in the case of serial robots, where the stability properties of most model based control laws have been rigorously proved, for parallel robots the stability properties of many of these control laws are not well established.

A precursor to the development of rigorous stability analyses for serial robots was the formulation of equations of motion of serial robots (1.1) and the establishment of properties 1.1-1.4. A major hurdle in establishing stability properties for parallel robots is the lack of a dynamics model similar to (1.1) that is valid for all parallel robots. The problem of deriving the equations of motion of parallel robots received considerable attention over the past decade. Much of this work was directed at deriving the equations of motion of a particular robot or a restricted class of robots with a specific structure [50], [39], [61], [21], [32], [49], [41], [19] and [17]. The results that are of interest for control are those that are more general.

In 1985 Luh and Zeng proposed a procedure for computing the dynamics of parallel robots for simulation and real time control [53]. Their objective was to derive an efficient algorithm to compute the input force/torques corresponding to a particular motion of the robot. That is, the algorithm should be able to compute the input joint forces/torques given the position, velocity, and acceleration of the robot. Most of the research on the dynamics of parallel robots concentrate on this problem, which is frequently referred to as the “inverse dynamics” problem. In [53] parallel robots with one loop only are considered and the authors claim that the extension of the method to general parallel robots is straight forward. The proposed method involves virtually cutting open the closed loop at a joint that is not actuated and then deriving the equations of motion of the resulting open loop robot. The Newton-Euler recursive formulation is proposed for deriving the equations of motion of the open loop robot. The actual equations of motion of the parallel robot are then computed, in terms of the joint variables and Lagrange multipliers, by using d’Alembert’s principal. An expression for the Lagrange multipliers in terms of the joint variables are also presented. The final equations of motion are in terms of both active and passive joint variables. While the procedure is useful for the inverse dynamics problem, it’s not

useful for control law design since we need information on the structure of the final equations of motion for control law design.

Around the same time Kleinfinger and Khalil proposed a similar procedure for computing the dynamics of parallel robots [45]. They also virtually cut open the closed loop and analyzed the resulting open loop robot first. This method became popular and essentially all of the results on the dynamics of parallel robots involves virtually cutting open a joint in the closed loop and then analyzing the open loop robot first. Kleinfinger and Khalil proposed a more efficient notation for describing the kinematics of the open loop robot and proposed the Newton-Euler recursive scheme to compute the dynamics of the open loop robot. In their formulation, the Lagrange multipliers are eliminated and the final equations of motion of the parallel robot are given in terms of the dynamics of the open loop robot and the partial derivatives of the constraints corresponding to the virtually cut joints. Their equations are also in terms of both the active and passive joint variables. Note that in a typical parallel robot the number of actuated joints equal the number of degrees of freedom of the robot and hence, the actuated joint variables form an independent set of coordinates. The joint variables corresponding to both the actuated and passive joints form a dependent coordinate set.

In [56], the dynamics of parallel robots that are over actuated are also taken in to account. The result is similar to that of [53] and is valid for parallel robots with multiple loops. Their method does not involve the Lagrange multipliers. In this work it is assumed that explicit expressions for the passive joint variables in terms of the active joint variables exist. This assumption however, is not valid globally for all parallel robots. For the case when the parallel robot is over actuated, the extra actuators are used to optimize some desired criterion such as the joint torques.

The issue of computing the equations of motion of parallel robots in terms of the actuated joint variables was addressed in several results. In the work of Murray and Lovell [55], the system that is obtained by virtually cutting open joints is referred to as the “reduced system”⁴. That is, the reduced system together with the constraints introduced by the loop closure represents the true parallel robot.

The dynamics of the reduced system can be obtained in terms of a coordinate set that is of greater dimension than the number of degrees of freedom of the parallel robot using methods that are already known. An explicit expression for the equations of motion of the parallel robot is then derived in terms of the dynamics of the reduced system and the Jacobians of the functions relating the different coordinate systems. In order to derive the equations of motion in terms of the independent coordinate set corresponding to the actuated joints, a coordinate transformation from the independent set to the dependent coordinate set must exist. The equations of motion derived in [55] assumes the existence of such a coordinate transformation. They discuss this issue and state that while it’s possible to derive explicit expressions for such a transformation for some parallel robots, for more complicated systems the transformation has to be computed through iterative techniques. The need for deriving the equations of motion in terms of the independent generalized coordinates corresponding to the actuated joint variables arise since they are the variables that are measured in control. In [55] this point is considered further and the situation when the measured variables are not the same as the actuated joint variables is also accounted for. This extra generalization is not important in most applications since the measured joint variables are almost always the actuated joint variables. A very similar result to that of [55], which also derives the dynamics of parallel robots in

⁴The term “Reduced system” is used in [55] to describe the open loop system obtained by virtually cutting open joints of a parallel robot. In our research, we will use the term “free system” to describe this open loop system.

terms of independent coordinates, is presented in [58] where an explicit expression, in terms of the derivatives of the constraint equations, for the jacobian matrix relating the independent and dependent coordinate systems is derived.

Several other related results on the dynamics of parallel robots are more suitable for simulations and fast computation for real time control. Among these, [52] uses the concept of dynamic equilibrium to solve the inverse dynamics problems for parallel robots. This work is equivalent to the result of [53]. Another result that is closely related to the results mentioned above is presented in [42]. This work has led to a numerical algorithm for the dynamic modeling and simulation of parallel robots. In [75] the equations of motion are formulated in cartesian space and then transformed into joint space using transformation Jacobian matrices. The authors propose using a recursive algorithm they developed in a previous paper [76] to solve the kinematics problem which can then be used to derive these Jacobian matrices. This work has also resulted in the development of a computer program that can compute the dynamics of parallel robots.

The specific problem of deriving a general expression for the equations of motion of parallel robots in a manner that is conducive to control law design was addressed in [24]. The final equations of motion derived in [24] have the same structure as the equations of motion for serial robots (1.1) and in subsequent work it was shown that they possess some of the important properties, such as the skew symmetry property (Property 1.2) of (1.1) [22]. Therefore, this work allows for the straight forward extension of the control results in the serial robot domain to parallel robots.

In the derivation of [24], the parallel robot is considered to be a system which consists of a free system Σ' to which constraints \mathcal{C} are applied as shown in Figure 1.3 for an illustrative planar mechanism. The free system Σ' is an n' degree-of-freedom (d.o.f) holonomic system obtained by virtually cutting joints in the parallel robot

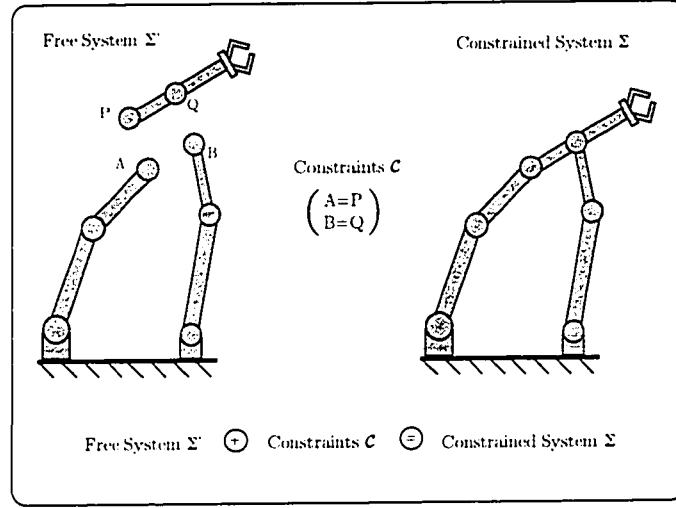


Figure 1.3: Free System, Constraints, and Constrained System

until we obtain a combination of serial robots or rigid bodies. Hence, the free system can be described by the following differential equation

$$\Sigma' : D'(\mathbf{q}') \ddot{\mathbf{q}}' + C'(\mathbf{q}', \dot{\mathbf{q}}') \dot{\mathbf{q}}' + \mathbf{g}'(\mathbf{q}') = 0 \quad (1.7)$$

where $\mathbf{q}' \in \mathbb{R}^{n'}$ is the vector of the generalized coordinates of the free system Σ' , $D'(\mathbf{q}') \in \mathbb{R}^{n' \times n'}$ is the inertia matrix, $C'(\mathbf{q}', \dot{\mathbf{q}}') \dot{\mathbf{q}}' \in \mathbb{R}^{n'}$ represent the centrifugal and Coriolis terms, and $\mathbf{g}'(\mathbf{q}') \in \mathbb{R}^{n'}$ is the gravity vector. The final equations of motion of the constrained system (the parallel robot) are given in terms of independent generalized coordinates \mathbf{q} by,

$$D(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u} \quad (1.8)$$

where the vector \mathbf{u} is the vector of generalized input forces/torques corresponding to the actuated joints and

$$\dot{\mathbf{q}}' = \rho(\mathbf{q}') \dot{\mathbf{q}},$$

$$\mathbf{q}' = \sigma(\mathbf{q}), \quad (1.9)$$

and

$$\begin{aligned}
 D(\mathbf{q}') &= \rho(\mathbf{q}')^T D'(\mathbf{q}') \rho(\mathbf{q}') \\
 C(\mathbf{q}', \dot{\mathbf{q}}') &= \rho(\mathbf{q}')^T C'(\mathbf{q}', \dot{\mathbf{q}}') \rho(\mathbf{q}') + \rho(\mathbf{q}')^T D'(\mathbf{q}') \dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}') \\
 \mathbf{g}(\mathbf{q}') &= \rho(\mathbf{q}')^T \mathbf{g}'(\mathbf{q}').
 \end{aligned}$$

Explicit expressions for the matrices $\rho(\mathbf{q}')$ and $\dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}')$ were also presented. The existence of the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ of (1.9) is established in the immediate neighborhood of a non singular configuration of the robot by the Implicit Function Theorem. Here, a singular configuration is defined as a point in the robot workspace for which the Jacobian of an augmented set of constraints becomes singular. It can be seen that the final equations of motion (1.8) have the identical structure as the equations of motion of serial robots in (1.1).

Before this result can be used to design control laws however, there are two important issues that must be resolved. Note that the existence of the parameterization (1.9) is necessary for the equations of motion (1.8) to be valid. Since the Implicit Function Theorem establishes only the existence of a neighborhood where (1.9) is valid and it does not characterize the neighborhood, we only know that the equations of motion are valid in the immediate vicinity of a valid, non singular configuration of the robot. In order to implement control we must be able to define a useful region of the robot workspace where the equations of motion (1.8) are valid. Therefore, the first issue that needs to be resolved is the characterization of a useful region where (1.9) is valid and thereby control can be successfully implemented. The second issue that must be addressed is the computation of (1.9) in real time for the implementation of control laws which compensate for the terms in the equations of motion. In the implementation of these control laws, the control system must perform several tasks during each sampling period T , which can be less than 1 milli second. These

include reading the sensory data, computing $\mathbf{q}' = \sigma(\mathbf{q})$, computing the control law expression, and sending the control to the actuators. Therefore, the time available for computing $\mathbf{q}' = \sigma(\mathbf{q})$ is in the microsecond range. Even though the existence of (1.9) is established by the Implicit Function Theorem, it does not provide a method for computing it. Therefore, in order to implement control laws that compensate for the dynamics (such as computed torque) there is a need for an efficient numerical scheme with guaranteed convergence to compute $\mathbf{q}' = \sigma(\mathbf{q})$ in real time. In this research we address these issues.

Another important contribution of this research is the design and construction of the Rice Planar Delta Robot (R.P.D.R.). An important aspect of research that is often neglected is the experimental verification of theoretical results. In the Robotics group at Rice University, we have paid much attention to this aspect. In order to experimentally verify the results of this research as well as future results on parallel robots, there was a need to design and construct a parallel robot. Since this robot was to be designed for academic research, many special features including ease of modeling, flexibility in terms of control law implementation, and the ability to vary model parameters were desired. We designed the R.P.D.R. with special attention to these issues. It was successfully used to verify the results of this research and will be used in the future to verify other results on parallel robots. The design process of the R.P.D.R. is described in Appendix A.

One of the most common applications of parallel robots is in pick-and-place operations where the robot is made to move back and forth between two points in the workspace. Here it's not important exactly what trajectory the robot followed in order to get to the desired end point. For this type of motion, commonly referred to as set point tracking, it's possible to use the PD plus simple gravity compensation

control law given by,

$$\mathbf{u} = K_P(\mathbf{q}^d - \mathbf{q}) - K_V\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}^d), \quad (1.10)$$

originally proposed in [68] for serial robots and later extended in [22] and [23] for parallel robots with guaranteed asymptotic stability. The main advantage of this control law is that the compensation term, $\mathbf{g}(\mathbf{q}^d)$, can be computed off-line. Hence, the issue of computing $\mathbf{q}' = \sigma(\mathbf{q})$ in real time does not arise. In Chapter 4 we implement the PD plus simple gravity compensation control law in a pick-and-place application. We use the Rice Planar Delta Robot (R.P.D.R.) for this experiment. We derive the equations of motion of the R.P.D.R. based on the formulation of [24] and perform simulations of the pick-and-place motion. We present experimental results along with simulations which show satisfactory agreement.

The two issues associated with the dynamics formulation of [24] mentioned above are addressed in Chapter 5. We begin by characterizing an explicit set, surrounding a given valid configuration of the robot, where the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ is valid. This is useful in determining the domain of attraction for a given control law. We extended this result and characterize a region of the robot workspace where the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ is valid along any trajectory contained within it. This result is important since control design and analysis is meaningful only when the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ exists. This contribution is useful in control synthesis also for trajectory planning. Finally we address the second issue mentioned above and propose a numerical algorithm for computing $\mathbf{q}' = \sigma(\mathbf{q})$ in real time and derive sufficient conditions for convergence. Based on this result we propose a procedure for implementing control in parallel robots. With this result we were able to implement trajectory tracking in the R.P.D.R. The trajectory selected makes the end-effector of the R.P.D.R. follow a circle and the well known inverse dynamics control law [66] was used. The experimental results are also presented and they confirm the theoretical

predictions. Finally in Chapter 6 we summarize the contributions of this thesis and propose extensions to this research.

Chapter 2

The Uniform Boundedness of the Coriolis/Centrifugal Terms

In this chapter we study the uniform boundedness of the Coriolis/centrifugal terms in (1.1). We will first characterize the class of robots for which the Coriolis/centrifugal terms are uniformly bounded and then propose a method for computing the uniform bound γ satisfying (1.3). This chapter is organized as follows: in Section 2.1 we set up notation and derive an expression for the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix in terms of link parameters. In section 2.2 we prove that the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is uniformly bounded for class \mathcal{BD} robots. This is followed by Section 2.3 where we prove that the necessary and sufficient condition for the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix to be uniformly bounded in a given robot is that it belong to class \mathcal{BD} . In section 2.4 we illustrate the procedure for computing the uniform bound γ satisfying (1.3) with an example. Finally, in Section 2.5 we summarize the findings of this chapter.

2.1 Kinematics Review

In this thesis we use the modified Denavit-Hartenberg (DH) convention of [14] to describe the kinematics of robots (see Figure 2.1). We summarize in Table 2.1, the notation used to determine the coordinate transformations. The transformation matrix ${}^i{}_{i-1}T$ from frame $(i-1)$ to frame i is given by,

$${}^i{}_{i-1}T = \begin{bmatrix} R_{i-1}^i & \mathbf{d}_{i-1}^i \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad R_{i-1}^i = R_x(\alpha_{i-1})R_z(\theta_i) = \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 \\ S_{\theta_i}C_{\alpha_{i-1}} & C_{\theta_i}C_{\alpha_{i-1}} & -S_{\alpha_{i-1}} \\ S_{\theta_i}S_{\alpha_{i-1}} & C_{\theta_i}S_{\alpha_{i-1}} & C_{\alpha_{i-1}} \end{bmatrix},$$

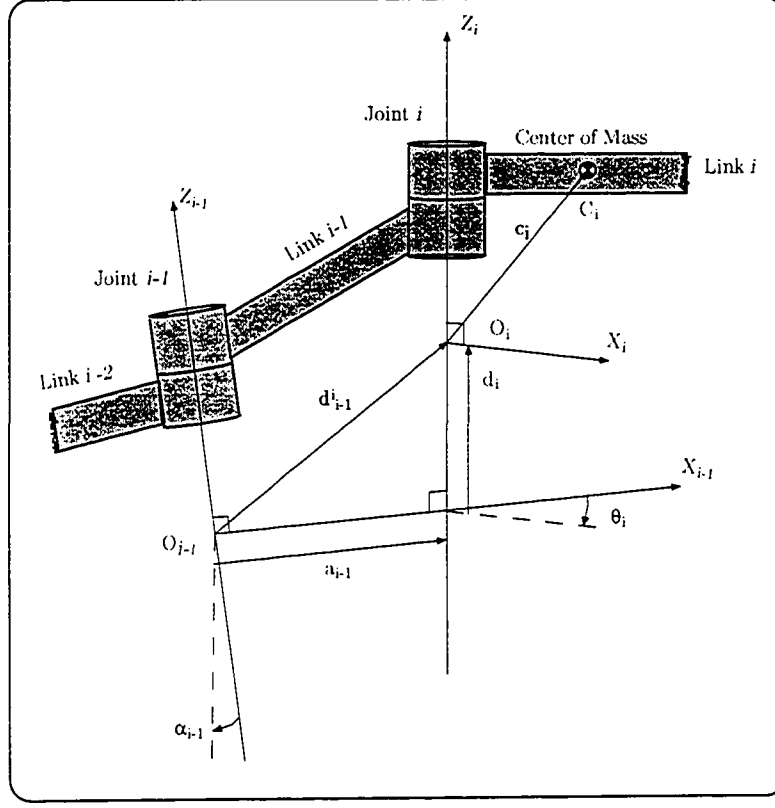


Figure 2.1: The Center of Mass Location+Modified DH Parameters

$$\mathbf{d}_{i-1}^i = \begin{bmatrix} a_{i-1} \\ -d_i S_{\alpha_{i-1}} \\ d_i C_{\alpha_{i-1}} \end{bmatrix},$$

where R_{i-1}^i is the rotation matrix from frame $(i-1)$ to frame i , and \mathbf{d}_{i-1}^i is the position vector of the origin of frame i in frame $(i-1)$. Note that $C_{angle} \triangleq \cos(angle)$, and $S_{angle} \triangleq \sin(angle)$.

In the results of this chapter, the center of mass locations also plays an important role. Let \mathbf{c}_i denote the position vector of the center of mass of link i expressed in coordinate frame i as shown in Figure 2.1. Since frame i is attached rigidly to link i , \mathbf{c}_i is constant. Let \mathbf{h}_i denote the position vector of the center of mass of link i in the

Symbol	Description
n	Number of joints in the robot
q_i	i^{th} joint variable
\mathbf{q}	n-dimensional vector of joint variables
a_i	Length parameter of link i
α_i	Twist parameter of link i
d_i	Offset parameter of link $(i - 1)$
θ_i	Angle parameter of link $(i - 1)$
\mathbf{z}_i	Unit vector along Z axis of frame i in base frame
$R_x(\alpha)$	Rotation matrix representing a rotation of α degrees about the X axis
$R_z(\theta)$	Rotation matrix representing a rotation of θ degrees about the Z axis

Table 2.1: Notation for Kinematics Parameters

base frame (frame 0). From geometry, we see that

$$\mathbf{h}_i = \sum_{j=1}^i \overrightarrow{O_{j-1}O_j} + \overrightarrow{O_iC_i},$$

where $\overrightarrow{O_{j-1}O_j}$ denotes the vector from O_{j-1} to O_j expressed in the base frame, and $\overrightarrow{O_iC_i}$ is the vector from O_i to the center of mass of link i expressed in the base frame. In terms of the quantities defined above, $\overrightarrow{O_{j-1}O_j} = R_0^{j-1} \mathbf{d}_{j-1}^j$ and $\overrightarrow{O_iC_i} = R_0^i \mathbf{c}_i$ (where $R_0^i = \prod_{j=1}^i R_{j-1}^j$ is the rotation matrix from the base frame to the i^{th} frame). Therefore,

$$\mathbf{h}_i = \sum_{j=1}^i R_0^{j-1} \mathbf{d}_{j-1}^j + R_0^i \mathbf{c}_i. \quad (2.1)$$

Note that when $j = 1$, $R_0^{j-1} = R_0^0 = I$.

Our next objective is to derive an expression for the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix in terms of link parameters. In order to retain the skew symmetry property, we will use the formulation in terms of the Christoffel symbols presented in [66]. The $(i, j)^{th}$ element of the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is defined as follows:

$$c_{i,j} = \sum_{k=1}^n c_{k,j,i} \dot{q}_k$$

$$= \sum_{k=1}^n \frac{1}{2} \left[\frac{\partial d_{i,j}}{\partial q_k} + \frac{\partial d_{i,k}}{\partial q_j} - \frac{\partial d_{k,j}}{\partial q_i} \right] \dot{q}_k, \quad (2.2)$$

where $c_{k,j,i}$, $i = 1, \dots, n$, $j = 1, \dots, n$, $k = 1, \dots, n$, are Christoffel symbols, and $d_{i,j}$ is the $(i, j)^{th}$ element of the inertia matrix, $D(\mathbf{q})$. Following the derivation in [66], the inertia matrix is given by

$$D(\mathbf{q}) = \sum_{l=1}^n \left(m_l J_{vl}^T(\mathbf{q}) J_{vl}(\mathbf{q}) + J_{\omega l}(\mathbf{q})^T R_0^l(\mathbf{q}) I_l R_0^l(\mathbf{q})^T J_{\omega l}(\mathbf{q}) \right),$$

where m_l and I_l are respectively, the mass and inertia about the center of mass of link l . Note that the inertia I_l is measured with respect to a coordinate frame rigidly attached to link l and hence, it's constant. The matrix $J_{vl}(\mathbf{q})$ is the Jacobian with respect to $\dot{\mathbf{q}}$ of the velocity of the center of mass of link l . Similarly $J_{\omega l}(\mathbf{q})$ is the Jacobian with respect to $\dot{\mathbf{q}}$ of the angular velocity of link l .

In order to simplify (2.2), we seek an expression for the $(i, j)^{th}$ element, $d_{i,j}$ of $D(\mathbf{q})$. Let $\mathbf{v}_{l,i}(\mathbf{q})$ and $\omega_{l,i}(\mathbf{q})$ be the i^{th} column of $J_{vl}(\mathbf{q})$ and $J_{\omega l}(\mathbf{q})$ respectively. From the expression above for the inertia matrix, we see that

$$d_{i,j} = \sum_{l=1}^n \left(m_l \mathbf{v}_{l,i}(\mathbf{q})^T \mathbf{v}_{l,j}(\mathbf{q}) + \omega_{l,i}(\mathbf{q})^T R_0^l(\mathbf{q}) I_l R_0^l(\mathbf{q})^T \omega_{l,j}(\mathbf{q}) \right). \quad (2.3)$$

Our Next objective is to derive expressions for the terms $\omega_{l,i}(\mathbf{q})$ and $\mathbf{v}_{l,i}(\mathbf{q})$ in (2.3) above. Since $J_{\omega l}(\mathbf{q}) = \frac{\partial \omega_l}{\partial \dot{\mathbf{q}}}$, where $\omega_l(\mathbf{q})$ is the angular velocity of link l expressed in the base frame,

$$\omega_{l,i}(\mathbf{q}) = \frac{\partial \omega_l(\mathbf{q})}{\partial \dot{q}_i}. \quad (2.4)$$

It was shown in [66] that

$$\omega_l(\mathbf{q}) = \sum_{i=1}^l \rho_i \dot{q}_i \mathbf{z}_i,$$

where

$$\rho_i = \begin{cases} 1 & : \text{if joint } i \text{ is revolute,} \\ 0 & : \text{if joint } i \text{ is prismatic,} \end{cases} \quad (2.5)$$

and \mathbf{z}_i is the axis of rotation of joint i based on our notation. By substituting into (2.4) we have the following:

$$\begin{aligned}\omega_{l,i}(\mathbf{q}) &= \frac{\partial}{\partial \dot{q}_i} \left(\sum_{i=1}^l \rho_i \dot{q}_i \mathbf{z}_i \right) \\ &= \frac{\partial}{\partial \dot{q}_i} \left(\sum_{i=1}^l \rho_i \dot{q}_i R_0^i \mathbf{z}_0 \right),\end{aligned}$$

where $\mathbf{z}_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. Consequently, by substituting for ρ_i from (2.5) and simplifying the expression we obtain the following:

$$\omega_{l,i}(\mathbf{q}) = \begin{cases} R_0^i \mathbf{z}_0 & : \text{ if joint } i \text{ is revolute and } i \leq l, \\ \mathbf{0} & : \text{ otherwise.} \end{cases} \quad (2.6)$$

Similarly, since $J_{vl}(\mathbf{q}) = \frac{\partial \mathbf{v}_l}{\partial \dot{\mathbf{q}}}$, where \mathbf{v}_l is velocity of the center of mass of link l expressed in the base frame,

$$\mathbf{v}_{l,i}(\mathbf{q}) = \frac{\partial \mathbf{v}_l(\mathbf{q})}{\partial \dot{q}_i}. \quad (2.7)$$

Note that $\mathbf{v}_l(\mathbf{q})$ is the time derivative of the position vector \mathbf{h}_l of the center of mass of link l . Therefore,

$$\mathbf{v}_l(\mathbf{q}) = \frac{d\mathbf{h}_l}{dt} = \sum_{i=1}^n \frac{\partial \mathbf{h}_l}{\partial q_i} \dot{q}_i,$$

and consequently,

$$\frac{\partial \mathbf{v}_l(\mathbf{q})}{\partial \dot{q}_i} = \frac{\partial \mathbf{h}_l}{\partial q_i}.$$

By substituting the above expression into (2.7) and using (2.1) we have the following:

$$\begin{aligned}\mathbf{v}_{l,i}(\mathbf{q}) &= \frac{\partial \mathbf{h}_l}{\partial q_i} \\ &= \frac{\partial}{\partial q_i} \left(\sum_{j=1}^l R_0^{j-1} \mathbf{d}_{j-1}^j + R_0^l \mathbf{c}_l \right).\end{aligned} \quad (2.8)$$

In order to simplify the expression for $\mathbf{v}_{l,i}(\mathbf{q})$, we have to take the derivative of \mathbf{d}_{i-1}^i . Let's explore the term \mathbf{d}_{i-1}^i next. Since θ_i is the joint variable when joint i is revolute

and d_i is the joint variable when joint i is prismatic, we can write

$$\mathbf{d}_{i-1}^i = \begin{cases} \begin{bmatrix} a_{i-1} \\ -d_i S_{\alpha_{i-1}} \\ d_i C_{\alpha_{i-1}} \end{bmatrix} & : \text{ if joint } i \text{ is revolute,} \\ \begin{bmatrix} a_{i-1} \\ -q_i S_{\alpha_{i-1}} \\ q_i C_{\alpha_{i-1}} \end{bmatrix} & : \text{ if joint } i \text{ is prismatic.} \end{cases}$$

It can be seen that \mathbf{d}_{i-1}^i is independent of joint variables and hence, is constant when joint i is revolute. For the case when joint i is prismatic, we will partition \mathbf{d}_{i-1}^i and separate the part that is dependent on prismatic joint variables from the part that is constant. By examining the transformation matrix ${}^{i-1}_i T$, it can be seen that \mathbf{d}_{i-1}^i is composed of three transformations. The following partitioning of \mathbf{d}_{i-1}^i is based on these transformations:

$$\mathbf{d}_{i-1}^i = a_{i-1} \mathbf{x}_0 + R_x(\alpha_{i-1}) \mathbf{z}_0 q_i,$$

where $\mathbf{x}_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. Next we define \mathbf{d}'_i to be a constant vector as follows:

$$\mathbf{d}'_i = \begin{cases} \begin{bmatrix} a_{i-1} \\ -d_i S_{\alpha_{i-1}} \\ d_i C_{\alpha_{i-1}} \end{bmatrix} & : \text{ if joint } i \text{ is revolute,} \\ a_{i-1} \mathbf{x}_0 & : \text{ if joint } i \text{ is prismatic.} \end{cases} \quad (2.9)$$

Now we can express \mathbf{d}_{i-1}^i in terms of \mathbf{d}'_i as follows:

$$\mathbf{d}_{i-1}^i = \mathbf{d}'_i + R_x(\alpha_{i-1}) \mathbf{z}_0 q_i \delta_i, \quad (2.10)$$

where,

$$\delta_i = \begin{cases} 0 & : \text{ if joint } i \text{ is revolute,} \\ 1 & : \text{ if joint } i \text{ is prismatic.} \end{cases} \quad (2.11)$$

By substituting from (2.10) into (2.7) we obtain

$$\begin{aligned} \mathbf{v}_{l,i}(\mathbf{q}) &= \frac{\partial}{\partial q_i} \left(\sum_{j=1}^l [R_0^{j-1} \mathbf{d}'_j + R_0^{j-1} R_x(\alpha_{j-1}) \mathbf{z}_0 q_j \delta_j] + R_0^l \mathbf{c}_l \right) \\ &= \frac{\partial}{\partial q_i} \left(\sum_{j=1}^l [R_0^{j-1} \mathbf{d}'_j + R_0^j \mathbf{z}_0 q_j \delta_j] + R_0^l \mathbf{c}_l \right), \end{aligned} \quad (2.12)$$

since $R_z(\theta_j) \mathbf{z}_0 = \mathbf{z}_0$ and therefore,

$$R_0^{j-1} R_x(\alpha_{j-1}) \mathbf{z}_0 = R_0^{j-1} R_x(\alpha_{j-1}) R_z(\theta_j) \mathbf{z}_0 = R_0^{j-1} R_{j-1}^j \mathbf{z}_0 = R_0^j \mathbf{z}_0.$$

In order to simplify (2.12) we will need to take derivatives of vectors of the form $R_0^i \mathbf{c}$, where \mathbf{c} is a constant vector, with respect to joint variables. Since such terms are independent of prismatic joint variables, their derivatives with respect to prismatic joint variables will vanish. For the derivatives with respect to revolute joint variables, we have the following result:

Lemma 2.1 For any revolute joint j and any constant vector \mathbf{c} :

$$\frac{\partial [R_0^i \mathbf{c}]}{\partial q_j} = \begin{cases} \mathbf{z}_j \times [R_0^i \mathbf{c}] & : \text{ if } j \leq i, \\ \mathbf{0} & : \text{ if } j > i, \end{cases} \quad (2.13)$$

where $\mathbf{z}_j = R_0^j \mathbf{z}_0$ is the unit vector in the direction of Z_j , the Z axis of coordinate frame j expressed in the base coordinate frame. \diamond

Proof of Lemma 2.1: Before presenting the proof of Lemma 2.1, we will present some well known properties of rotation matrices, skew symmetric matrices and the vector cross product that will be used in the proof. The transformation matrix, R_{i-1}^i is given by $R_{i-1}^i = R_x(\alpha_{i-1}) R_z(\theta_i)$, where

$$R_x(\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{\alpha_{i-1}} & -S_{\alpha_{i-1}} \\ 0 & S_{\alpha_{i-1}} & C_{\alpha_{i-1}} \end{bmatrix}, \text{ and } R_z(\theta_i) = \begin{bmatrix} C_{\theta_i} & -S_{\theta_i} & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

are basic rotation matrices about the x axis and the z axis [14]. Since $R_z(\theta_i) \in \mathcal{SO}(3)$, it follows that for a revolute joint i ,

$$\frac{\partial R_z(\theta_i)}{\partial \theta_i} = S(k)R_z(\theta_i); S(k) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathcal{SS}(3),$$

where $\mathcal{SS}(3)$ is the set of all 3×3 skew symmetric matrices [66]. For simplicity, we drop the index in $S(k)$ and we simply refer to it in this thesis by S . Therefore,

$$\frac{\partial R_{i-1}^i}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} R_x(\alpha_{i-1})R_z(\theta_i) = R_x(\alpha_{i-1})\frac{\partial R_z(\theta_i)}{\partial \theta_i} = R_x(\alpha_{i-1})SR_z(\theta_i). \quad (2.14)$$

Two important properties of rotation matrices and skew symmetric matrices are as follows [66]: For any given vector \mathbf{p} ,

$$S\mathbf{p} = S(\mathbf{k})\mathbf{p} = \mathbf{z}_0 \times \mathbf{p}, \quad (2.15)$$

where $\mathbf{z}_0 \triangleq [0 \ 0 \ 1]^T$. The second property is, for any two vectors \mathbf{a} and \mathbf{b} and any rotation matrix R ,

$$R(\mathbf{a} \times \mathbf{b}) = R\mathbf{a} \times R\mathbf{b}. \quad (2.16)$$

Now we are ready to prove Lemma 2.1. When $j > i$, the result follows immediately since the term to be differentiated is independent of q_j . When $j \leq i$, $R_0^i \mathbf{c} = R_0^{j-1} R_{j-1}^j R_j^i \mathbf{c}$ and since joint j is revolute, R_{j-1}^j is dependent on q_j , while all the other matrices and vectors are independent of q_j . Therefore,

$$\frac{\partial [R_0^i \mathbf{c}]}{\partial q_j} = R_0^{j-1} \frac{\partial [R_{j-1}^j]}{\partial q_j} R_j^i \mathbf{c} = R_0^{j-1} R_x(\alpha_{j-1}) S R_z(\theta_j) R_j^i \mathbf{c},$$

by using (2.14). Now by applying (2.15) we obtain the following:

$$\frac{\partial [R_0^i \mathbf{c}]}{\partial q_j} = R_0^{j-1} R_x(\alpha_{j-1}) (\mathbf{z}_0 \times R_z(\theta_j) R_j^i \mathbf{c}).$$

We use (2.16) next and we have

$$\begin{aligned} \frac{\partial [R_0^i \mathbf{c}]}{\partial q_j} &= R_0^{j-1} R_x(\alpha_{j-1}) \mathbf{z}_0 \times R_0^{j-1} R_x(\alpha_{j-1}) R_z(\theta_j) R_j^i \mathbf{c} \\ &= R_0^{j-1} R_x(\alpha_{j-1}) \mathbf{z}_0 \times R_0^i \mathbf{c}. \end{aligned} \quad (2.17)$$

It can be seen by direct computation that $R_x(\alpha_{j-1})\mathbf{z}_0 = R_x(\alpha_{j-1})R_z(\theta_j)\mathbf{z}_0 = R_{j-1}^j\mathbf{z}_0$. Therefore, $R_0^{j-1}R_x(\alpha_{j-1})\mathbf{z}_0 = R_0^{j-1}R_{j-1}^j\mathbf{z}_0 = R_0^j\mathbf{z}_0 = \mathbf{z}_j$. The result follows by substituting this into (2.17). \square

Now we are ready to simplify (2.12) by completing the differentiation. We will consider the two cases corresponding to joint i being prismatic and joint i being revolute separately. When joint i is prismatic, since the first and third terms are independent of prismatic joint variables, we have

$$\mathbf{v}_{l,i}(\mathbf{q}) = \begin{cases} R_0^i\mathbf{z}_0 & : \text{ if } i \leq l \\ \mathbf{0} & : \text{ if } i > l. \end{cases} \quad (2.18)$$

When joint i is revolute, by applying Lemma 2.1 to (2.12), we obtain

$$\mathbf{v}_{l,i}(\mathbf{q}) = \begin{cases} \mathbf{z}_i \times \left(\sum_{j=i+1}^l [R_0^{j-1}\mathbf{d}'_j + R_0^j\mathbf{z}_0q_j\delta_j] + R_0^l\mathbf{c}_l \right) & : \text{ if } i \leq l \\ \mathbf{0} & : \text{ if } i > l. \end{cases} \quad (2.19)$$

As we will see later, it's the terms involving prismatic joint variables that determine whether $\mathbf{v}_{l,i}(\mathbf{q})$ is uniformly bounded. For this reason it is convenient to partition the above expression for $\mathbf{v}_{l,i}(\mathbf{q})$ as follows:

$$\mathbf{v}_{l,i}(\mathbf{q}) = \mathbf{v}_{l,i}^b(\mathbf{q}) + \mathbf{v}_{l,i}^u(\mathbf{q}), \quad (2.20)$$

where

$$\mathbf{v}_{l,i}^b(\mathbf{q}) = \begin{cases} \mathbf{z}_i \times \left(\sum_{j=i+1}^l [R_0^{j-1}\mathbf{d}'_j] + R_0^l\mathbf{c}_l \right) & : \text{ if joint } i \text{ is } \mathcal{R} \text{ and } i \leq l \\ R_0^i\mathbf{z}_0 & : \text{ if joint } i \text{ is } \mathcal{P} \text{ and } i \leq l \\ \mathbf{0} & : \text{ otherwise,} \end{cases} \quad (2.21)$$

where \mathcal{R} and \mathcal{P} denotes revolute and prismatic respectively, and

$$\mathbf{v}_{l,i}^u(\mathbf{q}) = \begin{cases} \mathbf{z}_i \times \left(\sum_{j=i+1}^l R_0^j\mathbf{z}_0q_j\delta_j \right) & : \text{ if joint } i \text{ is revolute and } i \leq l \\ \mathbf{0} & : \text{ otherwise.} \end{cases} \quad (2.22)$$

2.2 A Uniform Bound for the $C(\mathbf{q}, \dot{\mathbf{q}})$ Matrix

In this section we will use the expressions we derived in the previous section to prove that the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is uniformly bounded for class \mathcal{BD} robots. Class \mathcal{BD} is defined as follows [31]:

Robots of Class \mathcal{BD} : Robots of this class have any of the following joint configurations:

- 2.1. All joints are prismatic ($\mathcal{PP} \cdots \mathcal{PP}$).
- 2.2. All joints are revolute ($\mathcal{RR} \cdots \mathcal{RR}$).
- 2.3. A series of prismatic joints followed by a series of revolute joints ($\mathcal{PP} \cdots \mathcal{PR} \cdots \mathcal{RR}$).
- 2.4. Configurations where the axis of translation of each prismatic joint j is parallel to all preceding revolute joints k .

Our first objective will be to prove that the Christoffel Symbols are uniformly bounded for class \mathcal{BD} robots. Consider the following expression for the Christoffel symbols [66]:

$$c_{i,j,k} = \frac{1}{2} \left[\frac{\partial d_{k,j}}{\partial q_i} + \frac{\partial d_{k,i}}{\partial q_j} - \frac{\partial d_{i,j}}{\partial q_k} \right]. \quad (2.23)$$

It can be seen that we need to prove that the derivatives of the elements of the inertia matrix are uniformly bounded in order to establish the uniform boundedness of the Christoffel symbols. By inspecting the expression for $d_{i,j}$ in (2.3), we see that $d_{i,j}$ is dependent on $\omega_{l,i}(\mathbf{q})$, $\mathbf{v}_{l,i}^b(\mathbf{q})$, $\mathbf{v}_{l,i}^u(\mathbf{q})$, R_0^l , and the constant inertia and mass parameters. We will consider the uniform boundedness of each of these terms next. We see from (2.6) that $\omega_{l,i}(\mathbf{q})$ is dependent only on revolute joint variables. Note further that revolute joint variables always appear as the arguments of trigonometric sine and cosine functions and that these trigonometric functions and their derivatives are always uniformly bounded by 1. Furthermore, due to the nature of the dependence of $\omega_{l,i}(\mathbf{q})$ on the trigonometric functions, they never become the denominator of any fraction (If a trigonometric sine or cosine term becomes the denominator of a fraction within $\omega_{l,i}(\mathbf{q})$, it will not be uniformly bounded as the trigonometric functions can

become zero). Hence $\omega_{l,i}(\mathbf{q})$ and its derivative is uniformly bounded. We see from (2.21) that $\mathbf{v}_{l,i}^b(\mathbf{q})$ and its derivative is also uniformly bounded for the same reason. Similarly, the terms R_0^l are also uniformly bounded. Since the inertia and mass terms are constant, the only source of non uniform boundedness is $\mathbf{v}_{l,i}^u(\mathbf{q})$. In the next lemma we prove that these terms vanish for class \mathcal{BD} robots.

Lemma 2.2 For Class \mathcal{BD} robots, $\mathbf{v}_{l,i}^u(\mathbf{q}) = 0$. \diamond

Proof of Lemma 2.2: From (2.22) it can be seen that $\mathbf{v}_{l,i}^u(\mathbf{q})$ is non zero only when joint i is revolute and $i \leq l$. We will prove next that $\mathbf{v}_{l,i}^u(\mathbf{q})$ is zero even in this case. By interchanging the vector product and the summation in (2.22) we have

$$\begin{aligned} \mathbf{v}_{l,i}^u(\mathbf{q}) &= \sum_{j=i+1}^l \mathbf{z}_i \times R_0^j \mathbf{z}_0 q_j \delta_j \\ &= \sum_{j=i+1}^l (\mathbf{z}_i \times \mathbf{z}_j) q_j \delta_j. \end{aligned}$$

In the summation $\sum_{j=i+1}^l (\mathbf{z}_i \times \mathbf{z}_j) q_j \delta_j$, for the terms corresponding to prismatic joints j , $\mathbf{z}_i \times \mathbf{z}_j = \mathbf{0}$ from the condition for belonging to class \mathcal{BD} , since joint i is a revolute joint preceding a prismatic joint j . Therefore, the terms corresponding to prismatic joints vanish. The terms corresponding to revolute joints j also vanish since $\delta_j = 0$ when joint j is revolute. This completes the proof of this lemma. \square

Now we are ready to present the following lemma on the uniform boundedness of the Christoffel symbols.

Lemma 2.3 For class \mathcal{BD} robots, all of the Christoffel symbols are uniformly bounded.

Proof of Lemma 2.3: Follows from the discussion above.

Our next objective is to derive a uniform bound γ satisfying (1.3) for class \mathcal{BD} robots. From (2.2) we see that

$$c_{i,j} = \sum_{k=1}^n c_{k,j,i} \dot{q}_k$$

$$\begin{aligned}
&= \begin{bmatrix} c_{1,j,i} & c_{2,j,i} & \cdots & c_{n,j,i} \end{bmatrix} \dot{\mathbf{q}} \\
&= \mathbf{p}_{j,i}^T \dot{\mathbf{q}},
\end{aligned}$$

where we define $\mathbf{p}_{j,i} = \begin{bmatrix} c_{1,j,i} & c_{2,j,i} & \cdots & c_{n,j,i} \end{bmatrix}^T$. By applying Schwarz' inequality [73],

$$|c_{i,j}| \leq \|\mathbf{p}_{j,i}\| \|\dot{\mathbf{q}}\|. \quad (2.24)$$

Since the Christoffel symbols are uniformly bounded for class \mathcal{BD} robots, for these robots there exists constants $\bar{c}_{k,j,i}$ such that

$$\|\mathbf{p}_{j,i}\| = \sqrt{\sum_{k=1}^n (c_{k,j,i})^2} \leq \sqrt{\sum_{k=1}^n (\bar{c}_{k,j,i})^2}, \forall \mathbf{q} \in \mathbb{R}^n. \quad (2.25)$$

Let

$$\bar{p}_{i,j} = \sqrt{\sum_{k=1}^n (\bar{c}_{k,j,i})^2}. \quad (2.26)$$

Therefore, from (2.24) we have the following:

$$|c_{i,j}| \leq \bar{p}_{i,j} \|\dot{\mathbf{q}}\|, \forall \mathbf{q} \in \mathbb{R}^n, \quad (2.27)$$

for class \mathcal{BD} robots. Thus we have derived uniform bounds for the elements of the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix. In order to derive a uniform bound γ satisfying (1.3), we need to translate the bounds on the elements to a bound on the matrix. The following lemma achieves this objective.

Lemma 2.4 Given matrices $A, B \in \mathbb{R}^{n \times n}$, where the $(i, j)^{th}$ elements of the matrices are denoted by $a_{i,j}$ and $b_{i,j}$ respectively, if $|a_{i,j}| \leq b_{i,j}$, for all $i = 1, \dots, n, j = 1, \dots, n$, then $\|A\| \leq \|B\|$ \diamond

Proof of Lemma 2.4: From the definition of the induced norm for matrices,

$$\|A\| = \sup_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$$

Let $\tilde{\mathbf{x}}$ be the unit vector satisfying the above equation. Hence, $\|\tilde{\mathbf{x}}\| = \sqrt{\sum_{i=1}^n (\tilde{x}_i)^2} = 1$ (\tilde{x}_i is the i^{th} component of $\tilde{\mathbf{x}}$) and $\|A\| = \|A\tilde{\mathbf{x}}\|$. Select $\mathbf{y} \in \mathbb{R}^n$ such that each component, $y_i = |\tilde{x}_i|$. Therefore, $\|\mathbf{y}\| = \sqrt{\sum_{i=1}^n (y_i)^2} = \sqrt{\sum_{i=1}^n (\tilde{x}_i)^2} = 1$. Consider $\|B\mathbf{y}\|^2 = \sum_{i=1}^n \left(\sum_{j=1}^n b_{i,j} y_j \right)^2$. Since $b_{i,j} y_j = b_{i,j} |\tilde{x}_j| \geq |a_{i,j}| |\tilde{x}_j|$,

$$\begin{aligned} \|B\mathbf{y}\|^2 &= \sum_{i=1}^n \left(\sum_{j=1}^n b_{i,j} y_j \right)^2 \geq \sum_{i=1}^n \left(\sum_{j=1}^n |a_{i,j}| |\tilde{x}_j| \right)^2 \\ &\geq \sum_{i=1}^n \left(\sum_{j=1}^n a_{i,j} \tilde{x}_j \right)^2 = \|A\tilde{\mathbf{x}}\|^2 = \|A\|^2. \end{aligned}$$

Therefore, $\|B\mathbf{y}\| \geq \|A\|$. Since $\|B\| = \sup_{\|\mathbf{x}\|=1} \|B\mathbf{x}\| \geq \|B\mathbf{y}\|$, $\|B\| \geq \|B\mathbf{y}\| \geq \|A\|$. This completes the proof of this lemma. \square

At this point we are ready to present the first major result of this chapter.

Theorem 2.1 For class \mathcal{BD} robots, the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is uniformly bounded and a constant γ satisfying (1.3) exists. This constant is given by,

$$\gamma = \|\overline{P}\|, \quad (2.28)$$

where the $(i, j)^{th}$ element of \overline{P} , $\overline{p}_{i,j}$ is defined in (2.26). \diamond

Proof of Theorem 2.1: Consider the two matrices $C(\dot{\mathbf{q}}, \mathbf{q})$ and $\overline{P}\|\dot{\mathbf{q}}\|$. From (2.27), we see that for class \mathcal{BD} robots, the $(i, j)^{th}$ elements satisfy the condition of Lemma 2.4. Therefore, from Lemma 2.4, $\|C(\dot{\mathbf{q}}, \mathbf{q})\| \leq \|[\overline{P}\|\dot{\mathbf{q}}\|]\| \leq \|\overline{P}\|\|\dot{\mathbf{q}}\|$, from the properties of the matrix norm since $\|\dot{\mathbf{q}}\|$ is a scalar. This completes the proof of this theorem. \square

2.3 Characterization of the Class of Robots for Which the $C(\dot{\mathbf{q}}, \mathbf{q})$ Matrix is Uniformly Bounded

In the previous section we proved that the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is uniformly bounded in \mathbf{q} for class \mathcal{BD} robots. Our objective in this section is to characterize the class of robots

for which the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is uniformly bounded. We will achieve this objective by proving that the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is uniformly bounded only for class \mathcal{BD} robots. We present the second major result of this chapter next.

Theorem 2.2 The $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is uniformly bounded for a given robot if and only if it belongs to class \mathcal{BD} . \diamond

Since we have proved in Theorem 2.1 that the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is uniformly bounded for class \mathcal{BD} robots, we have left to prove that for robots outside class \mathcal{BD} , the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is not uniformly bounded. Our approach will be to prove that for robots outside class \mathcal{BD} , one element of the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix can be made arbitrarily large. This will in turn imply that $\|C(\dot{\mathbf{q}}, \mathbf{q})\|$ can be made arbitrarily large. This will invalidate any uniform bound that is proposed for the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix.

Consider a robot that is outside class \mathcal{BD} . From the condition for belonging to class \mathcal{BD} , the robot has a revolute joint i and a prismatic joint p , such that $p > i$ and $\mathbf{z}_i \times \mathbf{z}_p \neq \mathbf{0}$. Consider the Christoffel symbol $c_{p,i,i}$, given by

$$\begin{aligned} c_{p,i,i} &= \frac{1}{2} \left(\frac{\partial d_{i,i}}{\partial q_p} + \frac{\partial d_{i,p}}{\partial q_i} - \frac{\partial d_{p,i}}{\partial q_i} \right) \\ &= \frac{1}{2} \frac{\partial d_{i,i}}{\partial q_p}, \end{aligned} \quad (2.29)$$

since the inertia matrix $D(\mathbf{q})$ is symmetric. From (2.3),

$$\frac{\partial d_{i,i}}{\partial q_p} = \sum_{l=1}^n \left(m_l \frac{\partial}{\partial q_p} ([\mathbf{v}_{l,i}(\mathbf{q})]^T [\mathbf{v}_{l,i}(\mathbf{q})]) + \frac{\partial}{\partial q_p} ([R_l^0(\mathbf{q})\omega_{l,i}(\mathbf{q})]^T I_l [R_l^0(\mathbf{q})\omega_{l,i}(\mathbf{q})]) \right).$$

From (2.6), we see that $\omega_{l,i}(\mathbf{q})$ is independent of prismatic joint variables. Therefore, since $R_l^0(\mathbf{q})$ is also independent of prismatic joint variables, $R_l^0(\mathbf{q})\omega_{l,i}(\mathbf{q})$ is independent of q_p and hence,

$$\begin{aligned} \frac{\partial d_{i,i}}{\partial q_p} &= \sum_{l=1}^n \left(m_l \frac{\partial}{\partial q_p} ([\mathbf{v}_{l,i}(\mathbf{q})]^T [\mathbf{v}_{l,i}(\mathbf{q})]) \right) \\ &= \sum_{l=1}^n \left(2m_l \left[\frac{\partial \mathbf{v}_{l,i}(\mathbf{q})}{\partial q_p} \right]^T [\mathbf{v}_{l,i}(\mathbf{q})] \right). \end{aligned}$$

Substituting the above expression into (2.29),

$$c_{p,i,i} = \sum_{l=1}^n \left(m_l \left[\frac{\partial \mathbf{v}_{l,i}(\mathbf{q})}{\partial q_p} \right]^T [\mathbf{v}_{l,i}(\mathbf{q})] \right).$$

It can be seen from (2.19), that $\mathbf{v}_{l,i}(\mathbf{q})$ is independent of q_p when $p > l$. Hence the terms corresponding to $l < p$ in the summation above vanish. Therefore,

$$c_{p,i,i} = \sum_{l=p}^n \left(m_l \left[\frac{\partial \mathbf{v}_{l,i}(\mathbf{q})}{\partial q_p} \right]^T [\mathbf{v}_{l,i}(\mathbf{q})] \right). \quad (2.30)$$

Let's concentrate on the term $\mathbf{v}_{l,i}(\mathbf{q})$ next. From (2.19),

$$\begin{aligned} \mathbf{v}_{l,i}(\mathbf{q}) &= \mathbf{z}_i \times \left(\sum_{j=i+1}^l [R_0^{j-1} \mathbf{d}'_j + R_0^j \mathbf{z}_0 q_j \delta_j] + R_0^l \mathbf{c}_l \right) \\ &= \mathbf{z}_i \times \left(\sum_{j=i+1}^l [R_0^{j-1} \mathbf{d}'_j] + R_0^l \mathbf{c}_l \right) + \sum_{j=i+1}^l (\mathbf{z}_i \times \mathbf{z}_j) \delta_j q_j. \end{aligned}$$

Now since $l > p$ for all of the terms appearing in the summation in (2.30), we can break up the second summation above to obtain the following:

$$\begin{aligned} \mathbf{v}_{l,i}(\mathbf{q}) &= \mathbf{z}_i \times \left(\sum_{j=i+1}^l [R_0^{j-1} \mathbf{d}'_j] + R_0^l \mathbf{c}_l \right) + \sum_{j=i+1}^{p-1} (\mathbf{z}_i \times \mathbf{z}_j) \delta_j q_j \\ &\quad + \sum_{j=p+1}^l (\mathbf{z}_i \times \mathbf{z}_j) \delta_j q_j + (\mathbf{z}_i \times \mathbf{z}_p) q_p \\ &= \mathbf{a}_l(\mathbf{q}) + (\mathbf{z}_i \times \mathbf{z}_p) q_p. \end{aligned}$$

Here $\mathbf{a}_l(\mathbf{q})$ is independent of q_p . Therefore, $\frac{\partial \mathbf{v}_{l,i}(\mathbf{q})}{\partial q_p} = \mathbf{z}_i \times \mathbf{z}_p$. By substituting into (2.30),

$$\begin{aligned} c_{p,i,i} &= \sum_{l=p}^n \left(m_l (\mathbf{z}_i \times \mathbf{z}_p)^T [\mathbf{a}_l(\mathbf{q}) + (\mathbf{z}_i \times \mathbf{z}_p) q_p] \right) \\ &= (\mathbf{z}_i \times \mathbf{z}_p)^T \sum_{l=p}^n \mathbf{a}_l(\mathbf{q}) m_l + (\|\mathbf{z}_i \times \mathbf{z}_p\|)^2 q_p \sum_{l=p}^n m_l \\ &= a_{p,i,i} + b_{p,i,i} q_p, \end{aligned}$$

where $a_{p,i,i}$ and $b_{p,i,i}$ are independent of q_p and $b_{p,i,i}$ is non zero (since $\mathbf{z}_i \times \mathbf{z}_p \neq \mathbf{0}$, and in order for $\sum_{l=p}^n m_l$ to be zero, all of m_p, m_{p+1}, \dots, m_n will have to be zero, which

is physically unrealistic). Now consider the element $c_{i,i}$. By keeping all other joints constant, hence $\dot{q}_i = 0$, for all $i \neq p$, we have $c_{i,i} = c_{p,i,i}\dot{q}_p = a_{p,i,i}\dot{q}_p + b_{p,i,i}\dot{q}_p q_p$, where $b_{p,i,i} \neq 0$. Therefore, $c_{i,i}$ can be made arbitrarily large by selecting q_p sufficiently large and $\dot{q}_p \neq 0$. Therefore, any uniform bound that is proposed for the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix can be invalidated. This completes the proof of this theorem. \square

Hence, the condition for non uniform boundedness of the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is the existence of a revolute joint i and a prismatic joint j such that $i < j$ and Z_i and Z_j are not parallel. Let's consider the example given in the introduction. Since the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is not uniformly bounded for that example, according to Theorem 2.2, there should be a revolute joint and a prismatic joint satisfying the above condition. We see that this is indeed the case since joint 1 is revolute and joint 2 is prismatic and Z_1 and Z_2 are not parallel in that example (See Figure 1.2). Now consider the same robot with the joint axes oriented such that Z_1 and Z_2 are parallel (Figure 2.2). Since the only revolute joint $i(= 1)$ and prismatic joint $j(= 2)$ such that $i < j$, satisfies the condition Z_i is parallel to Z_j , this robot belongs to class \mathcal{BD} . Hence, according to Theorem 2.2 the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix of this robot should be uniformly bounded. For this robot, the inertia matrix is given by,

$$D(\mathbf{q}) = \begin{bmatrix} m_1 l_{c1}^2 + m_2 a_1^2 + I_1 + I_2 & 0 \\ 0 & m_2 \end{bmatrix}.$$

It can be seen that the inertia matrix is constant and hence, all of the Christoffel symbols are zero. Therefore, the $C(\dot{\mathbf{q}}, \mathbf{q})$ matrix is zero and uniformly bounded. This is consistent with Theorem 2.2.

2.4 Computing the Uniform Bound γ : Illustrative Example

In this section we will illustrate the procedure for computing the uniform bound γ of (1.3) with an example. We see from (2.28) and (2.26) that we need to compute uniform bounds for the Christoffel symbols in order to compute γ . It was shown in Section 2.2

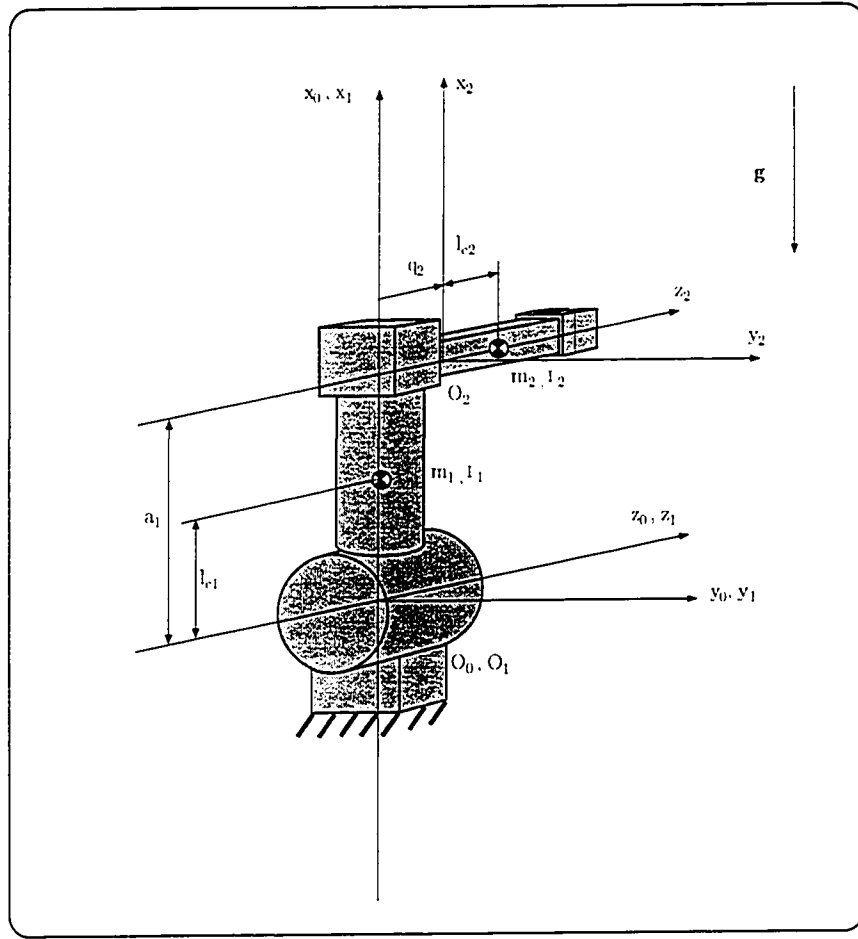


Figure 2.2: A 2 d.o.f \mathcal{RP} of Class \mathcal{BD}

that for class \mathcal{BD} robots, the Christoffel symbols are independent of prismatic joints and that the revolute joint variables only appear as arguments of trigonometric sine and cosine functions. Therefore, for class \mathcal{BD} robots it's possible to obtain uniform bounds for the Christoffel symbols by invoking the uniform bound 1 on trigonometric sine and cosine function. These can be substituted in (2.28) and (2.26) to obtain the uniform bound γ of (1.3). Consider the two d.o.f (\mathcal{RR}) robot of Figure 2.3. It can be seen that this robot belongs to class \mathcal{BD} since it has revolute joints only. Therefore, based on Theorem 2.1, the Coriolis/centrifugal terms are uniformly bounded for this

robot. The equations of motion of this robot was derived in [66] (see page 145 of

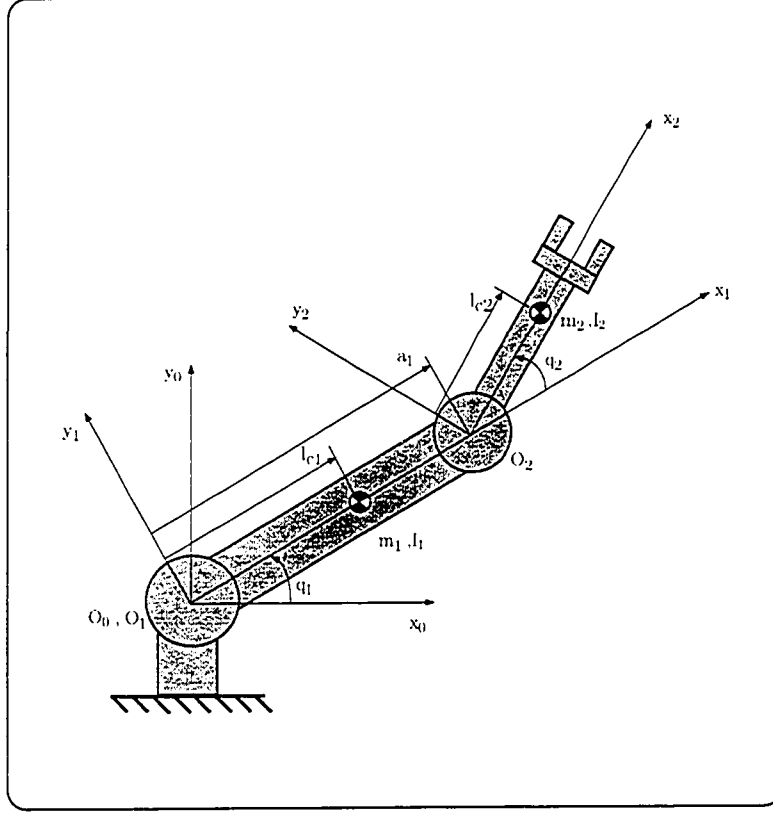


Figure 2.3: A 2 d.o.f \mathcal{RR} Robot

[66]). The Christoffel symbols, $c_{i,j,k}$ of this robot are as follows:

$$\begin{aligned}
 c_{1,1,1} &= 0 & c_{1,1,2} &= m_2 a_1 l_{c2} \sin q_2 & c_{1,2,1} &= -m_2 a_1 l_{c2} \sin q_2 \\
 c_{1,2,2} &= 0 & c_{2,1,1} &= -m_2 a_1 l_{c2} \sin q_2 & c_{2,1,2} &= 0 \\
 c_{2,2,1} &= -m_2 a_1 l_{c2} \sin q_2 & c_{2,2,2} &= 0.
 \end{aligned}$$

It can be seen by inspection that the following values for $\bar{c}_{i,j,k}$ satisfying (2.25):

$$\begin{aligned}
 \bar{c}_{1,1,1} &= 0 & \bar{c}_{1,1,2} &= m_2 a_1 l_{c2} & \bar{c}_{1,2,1} &= m_2 a_1 l_{c2} & \bar{c}_{1,2,2} &= 0 \\
 \bar{c}_{2,1,1} &= m_2 a_1 l_{c2} & \bar{c}_{2,1,2} &= 0 & \bar{c}_{2,2,1} &= m_2 a_1 l_{c2} & \bar{c}_{2,2,2} &= 0.
 \end{aligned}$$

Now by substituting in (2.26) we obtain the following:

$$\begin{aligned}\bar{p}_{1,1} &= \sqrt{\bar{c}_{1,1,1}^2 + \bar{c}_{2,1,1}^2} = m_2 a_1 l_{c2} \\ \bar{p}_{1,2} &= \sqrt{\bar{c}_{1,2,1}^2 + \bar{c}_{2,2,1}^2} = \sqrt{2} m_2 a_1 l_{c2} \\ \bar{p}_{2,1} &= \sqrt{\bar{c}_{1,1,2}^2 + \bar{c}_{2,1,2}^2} = m_2 a_1 l_{c2} \\ \bar{p}_{2,2} &= \sqrt{\bar{c}_{1,2,2}^2 + \bar{c}_{2,2,2}^2} = 0.\end{aligned}$$

Therefore, from (2.28)

$$\gamma = \|\bar{P}\| = \left\| \begin{bmatrix} m_2 a_1 l_{c2} & \sqrt{2} m_2 a_1 l_{c2} \\ m_2 a_1 l_{c2} & 0 \end{bmatrix} \right\| = 1.8478 m_2 a_1 l_{c2}.$$

In order to illustrate that the value of γ above satisfies (1.3), we will plot $\|C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\|$ and $\gamma\|\dot{\mathbf{q}}\|$ as a function of \mathbf{q} next. In order to make this plot meaningful, we must ensure that the point at which the left hand side of (1.3) becomes maximum is included in the plot. For this example the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is given by,

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -m_2 a_1 l_{c2} \dot{q}_2 \sin q_2 & -m_2 a_1 l_{c2} (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 a_1 l_{c2} \dot{q}_1 \sin q_2 & 0 \end{bmatrix}.$$

It can be seen that the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is independent of q_1 . Therefore, we will be plotting $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ and $\gamma\|\dot{\mathbf{q}}\|$ as a function of q_2 . We can select $\|\dot{\mathbf{q}}\| = 1$ without loss of generality. In order to ensure that the maximum point is included in the plot, we need to find $\dot{\mathbf{q}}$ that maximizes $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$. For this simple example, we were able to numerically determine this value to be $\dot{\mathbf{q}} = [0.5600 \ 0.8285]^T$. Figure 2.4 shows the plot of $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ and $\gamma\|\dot{\mathbf{q}}\|$ as a function of q_2 , with the optimum value for $\dot{\mathbf{q}}$. We used the values $a_1 = 0.7$, $m_2 = 0.25$, and $l_{c2} = 3$ for this plot. It can be seen from this plot that the uniform bound γ derived above satisfies (1.3).

2.5 Summary

The objective of this work was to characterize the class of robots for which the Coriolis/centrifugal terms are uniformly bounded and derive an explicit expression for

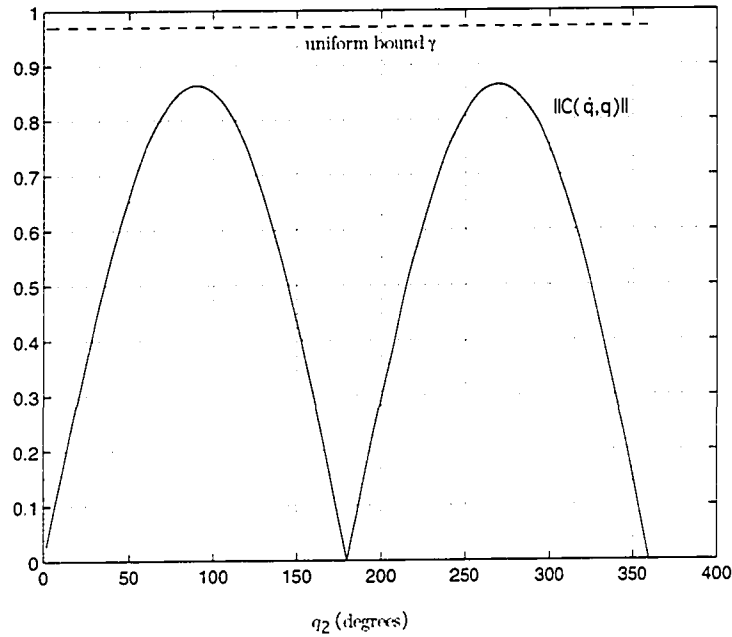


Figure 2.4: The Uniform Bound γ and $\|C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\|$ as a Function of q_2 for the \mathcal{RR} Robot of Figure 2.3 with $\|\dot{\mathbf{q}}\| = 1$

a uniform bound γ satisfying (1.3). The uniform boundedness of the Coriolis/centrifugal terms plays an important role in the stability analysis of several control laws. When the uniform boundedness property is not satisfied, the stability properties of these control laws become local. In this Chapter, we fully characterized the class of robots for which the Coriolis/centrifugal terms are uniformly bounded and found that this class is the same as the class of robots for which the inertia matrix is uniformly bounded. This class, named class \mathcal{BD} , was characterized in previous research. The characterization of the class of robots for which the Coriolis/centrifugal terms are uniformly bounded, will enable researchers to determine whether global stability conclusions of several control laws are valid for a given robot by inspection. This result can also be incorporated at the design stage since the condition for uniform boundedness of

the Coriolis/centrifugal terms is not hard to satisfy. In addition to characterizing the class of robots with uniformly bounded Coriolis/centrifugal terms, we derived an explicit expression for the uniform bound. The explicit value of the uniform bound is important in controller synthesis since the controller gains are a function of these uniform bounds. We illustrated the procedure for computing the uniform bound with an example.

Chapter 3

The Uniform Boundedness of the Hessian of Potential Energy

In this chapter we address the issue of the uniform boundedness of the Hessian. We begin by characterizing the class of robots for which the Hessian is uniformly bounded. This will allow one to determine by simple inspection of the joint configuration of a given robot whether the Hessian is uniformly bounded and thereby determine immediately whether control laws that use this property will retain globality. We refer to the class of robots with uniformly bounded Hessian as Class \mathcal{BGJ} robots¹. The second contribution of this chapter is the derivation of an easy to compute explicit expression for the uniform bound β in (1.2) in terms of link parameters for Class \mathcal{BGJ} robots.

Denoting the complement of Class \mathcal{BGJ} as Class $\overline{\mathcal{BGJ}}$, it follows that the latter consists of robots for which a uniform bound satisfying (1.2) does not exist. Thus the union of Class \mathcal{BGJ} and Class $\overline{\mathcal{BGJ}}$ consists of all robots. Even though a uniform bound satisfying (1.2) does not exist for robots of Class $\overline{\mathcal{BGJ}}$, we can derive a bound c_4 satisfying (1.4). If one chooses to apply the control laws that require the uniform bound β of (1.2) to a Class $\overline{\mathcal{BGJ}}$ robot, even though the stability properties will not be valid globally they will still be valid locally. Therefore, if a robot at hand is a Class $\overline{\mathcal{BGJ}}$ robot, an explicit expression for the bound c_4 satisfying (1.4) can still be used to implement control. Hence, for the third contribution of this chapter we derive

¹ \mathcal{BGJ} is an acronym for *Bounded Gravity Jacobian* which is equivalent to the Hessian of potential energy.

an expression for the bound c_4 of (1.4) for Class $\overline{\mathcal{BGJ}}$ robots. This bound is given in terms of the constant link parameters and upper bounds of prismatic joint variables.

In this chapter we use the same notation that was introduced in Chapter 2 to describe the kinematics of the robot. The remainder of this chapter is organized as follows: In Section 3.1 we derive an expression for the Hessian. Section 3.2 is devoted to the characterization of Class \mathcal{BGJ} robots. In Section 3.3 we derive an explicit expression for the uniform bound β that satisfies (1.2) for Class \mathcal{BGJ} robots. In Section 3.4 we derive an explicit expression for the bound c_4 that satisfies (1.4) for robots of Class $\overline{\mathcal{BGJ}}$. Finally in Section 3.5 we summarize the findings of this chapter.

3.1 The Hessian of Potential Energy

In this section we derive an expression for the Hessian. We begin by deriving an expression for the potential energy. Let \mathbf{v}_c be a unit vector in the direction of the gravity field but opposite in sense (pointing vertically upwards), expressed in the base frame. Note that since the direction of the gravity field is fixed with respect to the base frame, the direction of the unit vector \mathbf{v}_c is fixed and since its magnitude is also fixed, \mathbf{v}_c is constant. Let m_i denote the mass of link i . The total gravitational potential energy $V(\mathbf{q})$ of a robot described by (1.1) is given by the following expression:

$$V(\mathbf{q}) = \sum_{i=1}^n m_i g (\mathbf{v}_c \cdot \mathbf{h}_i), \quad (3.1)$$

where g is the gravitational acceleration constant $\approx 9.81 \text{ m/sec}^2$. By substituting from (2.1),

$$\begin{aligned} V(\mathbf{q}) &= \sum_{i=1}^n m_i g \left(\mathbf{v}_c \cdot \left[\sum_{j=1}^i R_0^{j-1} \mathbf{d}_{j-1}^j + R_0^i \mathbf{c}_i \right] \right) \\ &= g \mathbf{v}_c \cdot \sum_{i=1}^n m_i \sum_{j=1}^i \left(R_0^{j-1} \mathbf{d}_{j-1}^j \right) + \sum_{i=1}^n m_i g \mathbf{v}_c \cdot \left(R_0^i \mathbf{c}_i \right). \end{aligned} \quad (3.2)$$

We can obtain a more compact expression for $V(\mathbf{q})$ by changing the order of taking summation in the first term. We begin by expanding a part of this term.

$$\begin{aligned}
& \sum_{i=1}^n m_i \sum_{j=1}^i \left(R_0^{j-1} \mathbf{d}_{j-1}^j \right) = \\
& \quad m_1 \mathbf{d}_0^1 \\
& \quad + m_2 \mathbf{d}_0^1 \quad + m_2 R_0^1 \mathbf{d}_1^2 \\
& \quad \vdots \quad \quad \quad \ddots \\
& \quad + m_{n-1} \mathbf{d}_0^1 \quad + m_{n-1} R_0^1 \mathbf{d}_1^2 \quad \cdots \quad + m_{n-1} R_0^{n-2} \mathbf{d}_{n-2}^{n-1} \\
& \quad + m_n \mathbf{d}_0^1 \quad + m_n R_0^1 \mathbf{d}_1^2 \quad \cdots \quad + m_n R_0^{n-2} \mathbf{d}_{n-2}^{n-1} \quad + m_n R_0^{n-1} \mathbf{d}_{n-1}^n. \\
& = \sum_{i=1}^n \left(R_0^{i-1} \mathbf{d}_{i-1}^i \right) \sum_{j=i}^n m_j \\
& = \sum_{i=1}^n \left(R_0^{i-1} \mathbf{d}_{i-1}^i \right) \bar{m}_i,
\end{aligned} \tag{3.3}$$

where

$$\bar{m}_i \triangleq \sum_{j=i}^n m_j. \tag{3.4}$$

Note that (3.3) is obtained by taking summation over each column first. Now by substituting the above in (3.2), we get

$$\begin{aligned}
V(\mathbf{q}) &= g \mathbf{v}_c \cdot \sum_{i=1}^n \left(R_0^{i-1} \mathbf{d}_{i-1}^i \right) \bar{m}_i + \sum_{i=1}^n m_i g \mathbf{v}_c \cdot \left(R_0^i \mathbf{c}_i \right) \\
&= \sum_{i=1}^n \bar{m}_i g \mathbf{v}_c \cdot \left(R_0^{i-1} \mathbf{d}_{i-1}^i \right) + \sum_{i=1}^n m_i g \mathbf{v}_c \cdot \left(R_0^i \mathbf{c}_i \right).
\end{aligned} \tag{3.5}$$

By substituting from (2.10) into (3.5) we get the following:

$$V(\mathbf{q}) = \sum_{i=1}^n \bar{m}_i g \mathbf{v}_c \cdot \left(R_0^{i-1} \mathbf{d}_i' \right) + \sum_{i=1}^n \bar{m}_i g \mathbf{v}_c \cdot \left(R_0^{i-1} R_x(\alpha_{i-1}) \mathbf{z}_0 q_i \delta_i \right) + \sum_{i=1}^n m_i g \mathbf{v}_c \cdot \left(R_0^i \mathbf{c}_i \right).$$

Now since $R_z(\theta_i) \mathbf{z}_0 = \mathbf{z}_0$, we have $R_x(\alpha_{i-1}) \mathbf{z}_0 = R_x(\alpha_{i-1}) R_z(\theta_i) \mathbf{z}_0 = R_{i-1}^i \mathbf{z}_0$. Therefore, in the second term of the above equation, $R_0^{i-1} R_x(\alpha_{i-1}) \mathbf{z}_0 = R_0^{i-1} R_{i-1}^i \mathbf{z}_0 = R_0^i \mathbf{z}_0$. By substituting we get

$$V(\mathbf{q}) = \sum_{i=1}^n \bar{m}_i g \mathbf{v}_c \cdot \left(R_0^{i-1} \mathbf{d}_i' \right) + \sum_{i=1}^n \bar{m}_i g \mathbf{v}_c \cdot \left(R_0^i \mathbf{z}_0 q_i \delta_i \right) + \sum_{i=1}^n m_i g \mathbf{v}_c \cdot \left(R_0^i \mathbf{c}_i \right). \tag{3.6}$$

Now we are ready to derive an expression for $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$. Since the gravity vector \mathbf{g} is the gradient of the potential energy, we can write $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial^2 V(\mathbf{q})}{\partial \mathbf{q}^2}$. In the expression for $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$, the revolute joint variables are the arguments of trigonometric sine and cosine functions and hence are uniformly bounded. Therefore, terms that are independent of prismatic joint variables are uniformly bounded. For this reason, we will derive the expression for $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ such that the terms that are dependent on prismatic joint variables are isolated from those terms that are independent of prismatic joint variables. Note that the first and the last terms in the expression for potential energy in (3.6) are independent of prismatic joint variables. Hence, we define the matrix B to be independent of prismatic joint variables as follows:

$$\begin{aligned} B &= \frac{\partial^2}{\partial \mathbf{q}^2} \left(\sum_{i=1}^n \bar{m}_i g \mathbf{v}_c \cdot R_0^{i-1} \mathbf{d}'_i + \sum_{i=1}^n m_i g \mathbf{v}_c \cdot R_0^i \mathbf{c}_i \right) \\ &= \frac{\partial^2}{\partial \mathbf{q}^2} \sum_{i=1}^n \left(\bar{m}_i g \mathbf{v}_c \cdot R_0^{i-1} \mathbf{d}'_i + m_i g \mathbf{v}_c \cdot R_0^i \mathbf{c}_i \right) \end{aligned} \quad (3.7)$$

We define next matrices U_i corresponding to each joint $i = 1, \dots, n$, as follows:

$$U_i = \frac{\partial^2}{\partial \mathbf{q}^2} \delta_i q_i \bar{m}_i g \mathbf{v}_c \cdot R_0^i \mathbf{z}_0. \quad (3.8)$$

Here, each matrix U_i will in general depend on the prismatic joint variable q_i if joint i is prismatic. Note that if joint i is revolute, $\delta_i = 0$ according to (2.11) and consequently, $U_i = 0$. Based on the definitions above, we have the following expression for $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$:

$$\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} = B + \sum_{i=1}^n U_i. \quad (3.9)$$

3.2 Characterization of the Class of Robots with Uniformly Bounded Hessian (Class \mathcal{BGJ})

In this section we use the expression derived in the previous section for the Hessian to characterize Class \mathcal{BGJ} . We begin by presenting, in Section 3.2.1, some preliminary results that are necessary to establish the results to follow. Then in section 3.2.2 we

examine some properties of the matrices B and U_i , $i = 1, \dots, n$, appearing in (3.9), followed by the definition of Class \mathcal{BGJ} in Section 3.2.3. In Section 3.2.4 we prove that for a given robot, the Hessian is uniformly bounded if and only if it belongs to Class \mathcal{BGJ} .

3.2.1 Preliminary Results

In order to obtain expressions for the elements of B and U_i , $i = 1, \dots, n$, appearing in (3.9), we will need to take derivatives of vectors of the form $R_0^i \mathbf{c}$, where \mathbf{c} is a constant vector, with respect to joint variables. Since such terms are independent of prismatic joint variables, their derivatives with respect to prismatic joint variables will vanish. For the derivatives with respect to revolute joint variables, we use Lemma 2.1 and the following result:

Lemma 3.1 For any two revolute joints j and k such that $j \leq k$ and any constant vector \mathbf{c} , the following is true:

$$\frac{\partial^2 [R_0^i \mathbf{c}]}{\partial q_j \partial q_k} = \begin{cases} \mathbf{z}_j \times (\mathbf{z}_k \times [R_0^i \mathbf{c}]) & : \text{ if } k \leq i, \\ \mathbf{0} & : \text{ if } k > i. \end{cases} \quad (3.10)$$

◇

Proof of Lemma 3.1: When $k > i$, the result is immediate since $R_0^i (= \prod_{l=1}^i R_{l-1}^l)$ and \mathbf{c} are both independent of q_k . When $k \leq i$, since $j \leq k \leq i$, we can use Lemma 2.1 to obtain $\frac{\partial^2 [R_0^i \mathbf{c}]}{\partial q_j \partial q_k} = \frac{\partial^2 [R_0^i \mathbf{c}]}{\partial q_k \partial q_j} = \frac{\partial}{\partial q_k} \left(\frac{\partial [R_0^i \mathbf{c}]}{\partial q_j} \right) = \frac{\partial}{\partial q_k} (\mathbf{z}_j \times [R_0^i \mathbf{c}]) = \mathbf{z}_j \times \frac{\partial [R_0^i \mathbf{c}]}{\partial q_k}$ (since $\mathbf{z}_j = R_0^j \mathbf{z}_0$ is independent of q_k , as $k \geq j$). The result follows by applying Lemma 2.1 again. □

In the proofs to follow we will be using the set $W_j(\mathbf{q}^*)$ which is a subspace of the robot joint space. The use of the set $W_j(\mathbf{q}^*)$ allows us to present these proofs in a compact manner. We begin by defining $W_j(\mathbf{q}^*)$ as follows:

Definition 3.1 The set $W_j(\mathbf{q}^*)$ is defined as the set of joint vectors \mathbf{q} such that all of the components of \mathbf{q} are identical to the corresponding components of \mathbf{q}^* except for the j^{th} component, q_j which is arbitrary, namely,

$$W_j(\mathbf{q}^*) \triangleq \{\mathbf{q} \in \mathbb{R}^n : q_i = q_i^*, i = 1, \dots, j-1, j+1, \dots, n, \} \subset \mathbb{R}^n. \quad (3.11)$$

We can interpret the set $W_j(\mathbf{q}^*)$ as follows: Suppose we start the robot at point \mathbf{q}^* and then only move joint j while keeping all of the other joints fixed, then $W_j(\mathbf{q}^*)$ is the set of all possible vectors \mathbf{q} (of joint angles) corresponding to this motion. Now we are ready to present the following lemma:

Lemma 3.2 For any given revolute joint $j (< n)$ in a robot, the following is true: Suppose we pick a point $P^*(\mathbf{q}^*)$ in the robot workspace such that $\mathbf{z}_j \times \mathbf{z}_k \neq 0$ for some joint $k > j$, then the following two conditions are equivalent:

$$3.1. \quad \mathbf{v}_c \cdot \mathbf{z}_k = \text{constant for all points in } W_j(\mathbf{q}^*).$$

$$3.2. \quad \mathbf{z}_j \times \mathbf{v}_c = \mathbf{0} \text{ for all points in } W_j(\mathbf{q}^*).$$

◇

Proof of Lemma 3.2: We are given that $\mathbf{z}_j \times \mathbf{z}_k \neq 0$ at point \mathbf{q}^* . Now since we don't move any of the joints $l, \forall l > j$ in $W_j(\mathbf{q}^*)$, we can imagine that the robot has one rigid link from joint j to joint k . Now since we start at a point where the \mathbf{z}_j and \mathbf{z}_k axes are non parallel, and the only allowable motion is a pure rotation about the \mathbf{z}_j axis, the \mathbf{z}_j and \mathbf{z}_k axes will remain non parallel throughout $W_j(\mathbf{q}^*)$. Hence,

$$\mathbf{z}_j \times \mathbf{z}_k \neq 0, \forall \mathbf{q} \in W_j(\mathbf{q}^*). \quad (3.12)$$

We have to prove that $\mathbf{v}_c \cdot \mathbf{z}_k = \text{constant} \forall \mathbf{q} \in W_j(\mathbf{q}^*)$ if and only if $\mathbf{z}_j \times \mathbf{v}_c = \mathbf{0} \forall \mathbf{q} \in W_j(\mathbf{q}^*)$.

Proof of the “only if” part: Suppose $\mathbf{v}_c \cdot \mathbf{z}_k = \text{constant}$, $\forall \mathbf{q} \in W_j(\mathbf{q}^*)$. Therefore,

$$\frac{\partial}{\partial q_j} (\mathbf{v}_c \cdot \mathbf{z}_k) = 0.$$

Now since $\frac{\partial}{\partial q_j} (\mathbf{v}_c \cdot \mathbf{z}_k) = \mathbf{v}_c \cdot \frac{\partial(R_0^k \mathbf{z}_0)}{\partial q_j} = \mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_k)$ (by using Lemma 2.1),

$$\mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_k) = 0. \quad (3.13)$$

Since $\mathbf{v}_c \cdot \mathbf{z}_k = \text{constant}$, $\frac{\partial^2}{\partial q_j^2} (\mathbf{v}_c \cdot \mathbf{z}_k) = 0$. Now by applying Lemma 3.1, we get the following:

$$\begin{aligned} \frac{\partial^2}{\partial q_j^2} (\mathbf{v}_c \cdot \mathbf{z}_k) &= \mathbf{v}_c \cdot \frac{\partial^2 (R_0^k \mathbf{z}_0)}{\partial q_j^2} \\ &= \mathbf{v}_c \cdot [\mathbf{z}_j \times (\mathbf{z}_j \times R_0^k \mathbf{z}_0)] \\ &= (\mathbf{v}_c \times \mathbf{z}_j) \cdot (\mathbf{z}_j \times \mathbf{z}_k), \end{aligned}$$

by using the properties of the scalar triple product. Hence we have,

$$(\mathbf{v}_c \times \mathbf{z}_j) \cdot (\mathbf{z}_j \times \mathbf{z}_k) = 0. \quad (3.14)$$

A property of the vector product is as follows [47]. For any given four vectors, \mathbf{a} , \mathbf{b} , \mathbf{c} and \mathbf{d} ,

$$(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = [(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{d}] \mathbf{c} - [(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}] \mathbf{d}.$$

Now setting $\mathbf{a} = \mathbf{v}_c$, $\mathbf{b} = \mathbf{z}_j$, $\mathbf{c} = \mathbf{z}_j$ and $\mathbf{d} = \mathbf{z}_k$ we get,

$$\begin{aligned} &(\mathbf{v}_c \times \mathbf{z}_j) \times (\mathbf{z}_j \times \mathbf{z}_k) \\ &= [(\mathbf{v}_c \times \mathbf{z}_j) \cdot \mathbf{z}_k] \mathbf{z}_j - [(\mathbf{v}_c \times \mathbf{z}_j) \cdot \mathbf{z}_j] \mathbf{z}_k \\ &= 0, \end{aligned} \quad (3.15)$$

since $(\mathbf{v}_c \times \mathbf{z}_j) \cdot \mathbf{z}_j = \mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_j) = 0$ and from (3.13), $(\mathbf{v}_c \times \mathbf{z}_j) \cdot \mathbf{z}_k = \mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_k) = 0$. Comparing (3.14) and (3.15) we see that $\mathbf{a}' \cdot \mathbf{b}' = 0$ and $\mathbf{a}' \times \mathbf{b}' = 0$ simultaneously, where $\mathbf{a}' = \mathbf{v}_c \times \mathbf{z}_j$ and $\mathbf{b}' = \mathbf{z}_j \times \mathbf{z}_k$. Since for any two vectors \mathbf{a}' and \mathbf{b}' in the three dimensional Euclidean vector space $\mathbf{a}' \cdot \mathbf{b}' = \|\mathbf{a}'\| \|\mathbf{b}'\| \cos \theta$ and $\mathbf{a}' \times \mathbf{b}' = \|\mathbf{a}'\| \|\mathbf{b}'\| \sin \theta$ (where θ is the angle between the two vectors), $\mathbf{a}' \cdot \mathbf{b}' = 0$ and $\mathbf{a}' \times \mathbf{b}' = 0$ implies $\|\mathbf{a}'\| = 0$ or $\|\mathbf{b}'\| = 0$. Since $\mathbf{b}' = \mathbf{z}_j \times \mathbf{z}_k \neq 0$ from (3.12), we get $\mathbf{v}_c \times \mathbf{z}_j = 0$. This completes the proof of the “only if” part.

Proof of the “if” part: Suppose $\mathbf{v}_c \times \mathbf{z}_j = 0, \forall \mathbf{q} \in W_j(\mathbf{q}^*)$. By applying Lemma 2.1 to the scalar product $(\mathbf{v}_c \cdot \mathbf{z}_k)$ and then using the properties of the scalar triple product, we get

$$\begin{aligned} \frac{\partial}{\partial q_j} (\mathbf{v}_c \cdot \mathbf{z}_k) &= \mathbf{v}_c \cdot \frac{\partial (R_0^{k-1} \mathbf{z}_0)}{\partial q_j} \\ &= \mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_k) \\ &= (\mathbf{v}_c \times \mathbf{z}_j) \cdot \mathbf{z}_k \\ &= 0. \end{aligned}$$

Note that the other variables in $(\mathbf{v}_c \cdot \mathbf{z}_k)$ are independent of q_j and therefore, $\frac{d}{dq_j} (\mathbf{v}_c \cdot \mathbf{z}_k) = \frac{\partial}{\partial q_j} (\mathbf{v}_c \cdot \mathbf{z}_k)$. Now since q_j is the only variable in $W_j(\mathbf{q}^*)$, and since $\frac{d}{dq_j} (\mathbf{v}_c \cdot \mathbf{z}_k) = \frac{\partial}{\partial q_j} (\mathbf{v}_c \cdot \mathbf{z}_k) = 0$, $\mathbf{v}_c \cdot \mathbf{z}_k$ remains constant throughout $W_j(\mathbf{q}^*)$. This completes the proof of this lemma. \square

This result can be given a physical interpretation as follows: The scalar product $\mathbf{v}_c \cdot \mathbf{z}_k$ represents the component along the direction of gravity of the vector \mathbf{z}_k . Suppose we start at a point where the two vectors \mathbf{z}_j and \mathbf{z}_k are not parallel. This lemma states that the component along the direction of gravity of \mathbf{z}_k will remain constant as joint j is moved if and only if \mathbf{z}_j is always parallel to \mathbf{v}_c . Using this result we prove Lemma 3.3, which will be used during the characterization of Class \mathcal{BGJ} .

Lemma 3.3 For any revolute joint j and any other joint k (revolute or prismatic) such that $j \leq k$, if \mathbf{z}_k is always parallel to \mathbf{v}_c , then \mathbf{z}_j is also always parallel to \mathbf{v}_c . \diamond

Proof of Lemma 3.3: When $j = k$, the result is obvious. When $j < k$, we prove by contradiction. Suppose there exists a point \mathbf{q}^* where \mathbf{z}_j is not parallel to \mathbf{v}_c . Since \mathbf{z}_k is parallel to \mathbf{v}_c at \mathbf{q}^* , \mathbf{z}_j and \mathbf{z}_k are not parallel and hence, $\mathbf{z}_j \times \mathbf{z}_k \neq 0$ at \mathbf{q}^* . Now since there are points in $W_j(\mathbf{q}^*)$ where \mathbf{z}_j is not parallel to \mathbf{v}_c , from Lemma 3.2, $\mathbf{v}_c \cdot \mathbf{z}_k$ cannot be constant throughout $W_j(\mathbf{q}^*)$. This is a contradiction since \mathbf{v}_c and \mathbf{z}_k are parallel and $\mathbf{v}_c \cdot \mathbf{z}_k = 1$ is constant throughout the robot workspace. This completes the proof of Lemma 3.3. \square

3.2.2 Properties of the Matrices B and $U_i, i = 1, \dots, n$

We will be exploiting the symmetry and boundedness properties of the matrices B and $U_i, i = 1, \dots, n$, when deriving bounds for the Hessian. In this section we prove these properties.

Lemma 3.4 The matrices B, U_i for $i = 1, \dots, n$, are symmetric and hence, $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is symmetric. \diamond

Proof of Lemma 3.4: Consider $b_{j,k}$, a general element of B . From (3.7),

$$\begin{aligned} b_{j,k} &= \frac{\partial^2}{\partial q_j \partial q_k} \left[\sum_{i=1}^n \left(\bar{m}_i g \mathbf{v}_c \cdot R_0^{i-1} \mathbf{d}'_i + m_i g \mathbf{v}_c \cdot R_0^i \mathbf{c}_i \right) \right] \\ &= \frac{\partial^2}{\partial q_k \partial q_j} \left[\sum_{i=1}^n \left(\bar{m}_i g \mathbf{v}_c \cdot R_0^{i-1} \mathbf{d}'_i + m_i g \mathbf{v}_c \cdot R_0^i \mathbf{c}_i \right) \right] \\ &= b_{k,j}. \end{aligned}$$

Hence, B is symmetric. It can be seen from (3.8) that the matrices U_i are also symmetric for the same reason. It can be seen from (3.9) that $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is symmetric since it is the sum of symmetric matrices. This completes the proof of Lemma 3.4. \square

Lemma 3.5 The matrix B is uniformly bounded, that is,

$$\|B\| \leq b \quad (3.16)$$

where the uniform bound b is given by

$$b = \sqrt{\sum_{j=1}^n (\mu_j)^2 \left(\sum_{k=j+1}^n (\bar{b}_k)^2 + (\bar{b}_j)^2 \right)}. \quad (3.17)$$

Here,

$$\mu_j = \begin{cases} 0 & : \text{if joint } j \text{ is prismatic,} \\ |\sin(\gamma_j)| & : \text{if } \mathbf{z}_j \text{ is fixed in the base frame,} \\ 1 & : \text{otherwise,} \end{cases} \quad (3.18)$$

where γ_j is the angle between \mathbf{z}_j and \mathbf{v}_c , and \bar{b}_k is defined as follows:

$$\bar{b}_k = \begin{cases} \sum_{i=k+1}^n \bar{m}_i g \bar{d}_{k,i} + \sum_{i=k}^n m_i g \bar{c}_{k,i} & : \text{if joint } k \text{ is revolute,} \\ 0 & : \text{if joint } k \text{ is prismatic,} \end{cases} \quad (3.19)$$

and

$$\bar{d}_{k,i} = \begin{cases} \|\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i)\| & : \text{if for all revolute joints } l \text{ such that } k < l \leq i, \mathbf{z}_l \parallel \mathbf{z}_k, \\ \|\mathbf{d}'_i\| & : \text{otherwise,} \end{cases} \quad (3.20)$$

$$\bar{c}_{k,i} = \begin{cases} \|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\| & : \text{if for all revolute joints } l \text{ such that } k < l \leq i, \mathbf{z}_l \parallel \mathbf{z}_k, \\ \|\mathbf{c}_i\| & : \text{otherwise.} \end{cases} \quad (3.21)$$

◇

Proof of Lemma 3.5: It is well known that,

$$\|B\| \leq \sqrt{\text{Trace}(B^T B)}. \quad (3.22)$$

It can be easily verified by direct computation that the Trace of $B^T B$ is the sum of the squares of all of the terms of B . That is, $\text{Trace}(B^T B) = \sum_{j=1}^n \sum_{k=1}^n (b_{j,k})^2$. Next

we partition this double summation into three blocks, corresponding to the elements below the diagonal, the elements above the diagonal, and the elements on the diagonal as follows:

$$\begin{aligned} \text{Trace}(B^T B) &= \sum_{j=1}^n \sum_{k=1}^n (b_{j,k})^2 \\ &= \sum_{j=2}^n \sum_{k=1}^{j-1} (b_{j,k})^2 + \sum_{j=1}^{n-1} \sum_{k=j+1}^n (b_{j,k})^2 + \sum_{j=1}^n (b_{j,j})^2. \end{aligned}$$

Now since B is symmetric, the first two terms in the above equation are equal. Therefore,

$$\begin{aligned} \text{Trace}(B^T B) &= 2 \sum_{j=1}^{n-1} \sum_{k=j+1}^n (b_{j,k})^2 + \sum_{j=1}^n (b_{j,j})^2 \\ &= \sum_{j=1}^n \left(2 \sum_{k=j+1}^n (b_{j,k})^2 + (b_{j,j})^2 \right). \end{aligned} \quad (3.23)$$

Now consider the j, k^{th} element of B . From (3.7),

$$b_{j,k} = \frac{\partial^2}{\partial q_j \partial q_k} \left[\sum_{i=1}^n \left(\bar{m}_i g \mathbf{v}_c \cdot R_0^{i-1} \mathbf{d}'_i + m_i g \mathbf{v}_c \cdot R_0^i \mathbf{c}_i \right) \right].$$

It can be seen immediately that $b_{j,k} = 0$ if either of joints j or k is prismatic since the term to be differentiated is independent of prismatic joint variables. When both joints j and k are revolute, we can use Lemma 3.1 to obtain the following expression:

$$\begin{aligned} b_{j,k} &= \sum_{i=k+1}^n \bar{m}_i g \mathbf{v}_c \cdot \left[\mathbf{z}_j \times \left(\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i) \right) \right] + \sum_{i=k}^n m_i g \mathbf{v}_c \cdot \left[\mathbf{z}_j \times \left(\mathbf{z}_k \times (R_0^i \mathbf{c}_i) \right) \right] \\ &= \sum_{i=k+1}^n \bar{m}_i g (\mathbf{v}_c \times \mathbf{z}_j) \cdot \left(\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i) \right) + \sum_{i=k}^n m_i g (\mathbf{v}_c \times \mathbf{z}_j) \cdot \left(\mathbf{z}_k \times (R_0^i \mathbf{c}_i) \right) \\ &= (\mathbf{v}_c \times \mathbf{z}_j) \cdot \left[\sum_{i=k+1}^n \bar{m}_i g \left(\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i) \right) + \sum_{i=k}^n m_i g \left(\mathbf{z}_k \times (R_0^i \mathbf{c}_i) \right) \right]. \end{aligned}$$

Note the change in the limits of the summations due to the derivatives being zero outside these limits. Now we use Shwarz's inequality [73] and the triangle inequality to obtain,

$$|b_{j,k}| \leq \|\mathbf{v}_c \times \mathbf{z}_j\| \left[\sum_{i=k+1}^n \bar{m}_i g \|\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i)\| + \sum_{i=k}^n m_i g \|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\| \right]. \quad (3.24)$$

From the properties of the vector product, we see that $\|\mathbf{v}_c \times \mathbf{z}_j\| = \|\mathbf{v}_c\| \|\mathbf{z}_j\| |\sin(\gamma_j)|$, where γ_j is the angle between \mathbf{v}_c and \mathbf{z}_j . Now since \mathbf{v}_c and \mathbf{z}_j are both unit vectors, $\|\mathbf{v}_c \times \mathbf{z}_j\| = |\sin(\gamma_j)|$. When joint j is fixed in the base frame, $\sin(\gamma_j)$ is constant and it can be used as an upper bound for $\|\mathbf{v}_c \times \mathbf{z}_j\|$. When joint j is not fixed in the base frame, since $|\sin(\gamma_j)| \leq 1$, $\|\mathbf{v}_c \times \mathbf{z}_j\|$ is bounded from above by 1. Hence, we see that

$$\|\mathbf{v}_c \times \mathbf{z}_j\| \leq \mu_j \quad (3.25)$$

for the case when joint j is revolute, where μ_j is defined in (3.18). Now since $b_{j,k} = 0$ when joint j is prismatic, by defining μ_j as in (3.18), we can combine the two cases corresponding to joint j being revolute and joint j being prismatic. Thus (3.25) is satisfied irrespective of whether joint j is revolute or prismatic.

Our next objective is to obtain a constant bound for the second term in the right hand side of (3.24), $\sum_{i=k+1}^n \bar{m}_i g \|\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i)\| + \sum_{i=k}^n m_i g \|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\|$. We achieve this by deriving bounds for the terms in the summations. That is, we seek constant bounds $\bar{d}_{k,i}$ and $\bar{c}_{k,i}$ such that,

$$\|\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i)\| \leq \bar{d}_{k,i}, \quad (3.26)$$

and

$$\|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\| \leq \bar{c}_{k,i}. \quad (3.27)$$

Since \mathbf{z}_k is a unit vector, it can be seen immediately that $\|\mathbf{z}_k \times (R_0^{i-1} \mathbf{d}'_i)\| \leq \|\mathbf{d}'_i\|$ and $\|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\| \leq \|\mathbf{c}_i\|$. Now since both \mathbf{d}'_i and \mathbf{c}_i are constant vectors, we can set $\bar{d}_{k,i} = \|\mathbf{d}'_i\|$ and $\bar{c}_{k,i} = \|\mathbf{c}_i\|$ which will satisfy (3.26) and (3.27). While these will be satisfactory most of the time, there is one special case in which this will result in a conservative bound for B . This case occurs when for all revolute joints l such that $k < l \leq i$, $\mathbf{z}_k \parallel \mathbf{z}_l$. In this case, as we will prove next, $\|\mathbf{z}_k \times (R_0^i \mathbf{d}'_i)\|$ and $\|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\|$

are constant and we can select these as the bounds $\bar{d}_{k,i}$ and $\bar{c}_{k,i}$. Therefore, we define $\bar{d}_{k,i}$ and $\bar{c}_{k,i}$ as in (3.20) and (3.21).

We have left to prove that $\bar{d}_{k,i} = \|\mathbf{z}_k \times (R_0^{i-1}\mathbf{d}'_i)\|$ and $\bar{c}_{k,i} = \|\mathbf{z}_k \times (R_0^i\mathbf{c}_i)\|$ are constant in the first case of (3.20) and (3.21) (when for all revolute joint l such that $k < l \leq i$, $\mathbf{z}_l \parallel \mathbf{z}_k$). Based on the requirements for this case, the axes of all revolute joints l such that $k < l \leq i$ are parallel to the axis of joint k . Therefore, we can express $R_0^{i-1}\mathbf{d}'_i$ as follows:

$$R_0^{i-1}\mathbf{d}'_i = R_0^k R_z(q^*) R_c \mathbf{d}'_i, \quad (3.28)$$

where $R_z(q^*)$ is the basic rotation matrix about the z axis and R_c is a constant rotation matrix (we can regard $R_z(q^*)$ as the rotation matrix from coordinate frame j to the frame corresponding to the last revolute joint m such that $k < m \leq i$ and R_c as the rotation matrix from frame m to frame i). Note that q^* is variable. It can be seen that

$$\mathbf{z}_k = R_0^k \mathbf{z}_0 = R_0^k R_z(q^*) \mathbf{z}_0 \quad (3.29)$$

Now by combining (3.28) and (3.29) we have

$$\begin{aligned} \mathbf{z}_k \times (R_0^i \mathbf{d}'_i) &= R_0^k R_z(q^*) \mathbf{z}_0 \times R_0^k R_z(q^*) R_c \mathbf{d}'_i \\ &= R_0^k R_z(q^*) [\mathbf{z}_0 \times R_c \mathbf{d}'_i], \end{aligned}$$

by using the property described in (2.16). Now since $\|R\mathbf{x}\| = \|\mathbf{x}\|$ for any rotation matrix R and any vector \mathbf{x} , $\|\mathbf{z}_k \times (R_0^i \mathbf{d}'_i)\| = \|R_0^k R_z(q^*) [\mathbf{z}_0 \times R_c \mathbf{d}'_i]\| = \|\mathbf{z}_0 \times R_c \mathbf{d}'_i\|$ is constant. We can see from the same argument that $\|\mathbf{z}_k \times (R_0^i \mathbf{c}_i)\|$ is also constant in this case.

Now we are ready to use the bounds defined above to derive an upper bound for the Hessian. We substitute from (3.25), (3.26) and (3.27) into (3.24) to obtain the

following:

$$\begin{aligned}
 |b_{j,k}| &\leq \mu_j \left[\sum_{i=k+1}^n \bar{m}_i g \bar{d}_{k,i} + \sum_{i=k}^n m_i g \bar{c}_{k,i} \right] \\
 &\leq \mu_j \bar{b}_k,
 \end{aligned} \tag{3.30}$$

where \bar{b}_k is defined in (3.19). Note that we have combined the two cases corresponding to joint k being revolute and joint k being prismatic by defining \bar{b}_k in this manner. Now by substituting from (3.30) into (3.23)

$$\begin{aligned}
 \text{Trace}(B^T B) &= \sum_{j=1}^n \left(2 \sum_{k=j+1}^n (b_{j,k})^2 + (b_{j,j})^2 \right) \\
 &\leq \sum_{j=1}^n \left(2 \sum_{k=j+1}^n (\mu_j \bar{b}_k)^2 + (\mu_j \bar{b}_j)^2 \right) \\
 &\leq \sum_{j=1}^n (\mu_j)^2 \left(2 \sum_{k=j+1}^n (\bar{b}_k)^2 + (\bar{b}_j)^2 \right).
 \end{aligned}$$

The result follows by substituting in (3.22). This completes the proof of Lemma 3.5

□

One criticism that might be aimed at the uniform bound b derived above, is the use of the Trace of $B^T B$ as an upper bound for $\|B\|$ in (3.22). Since the Trace is the sum of all of the eigenvalues of $\|B^T B\|$ and $\|B\|$ is the maximum eigenvalue of $\|B\|$, for most matrices (3.22) will be highly conservative. However, in this case due to the special structure of the matrix B , we found that the maximum eigenvalue dominates the Trace. In all of the examples we looked at, the maximum eigenvalue was greater than ninety five percent of the Trace. Even though the use of (3.22) is easily justified, at one point we searched for a less conservative bound. Since the determinant of B can be zero, most of the inequalities presented in the literature cannot be used and we derived a uniform bound (say b') for $\|B\|$ in terms of upper bounds for the row vectors of B . We proved that the bound b' is always less than or equal to the bound b resulting from (3.22). However, when we applied the less conservative bound b' to

examples we discovered that the difference between the two bounds was usually less than one percent. For this reason, since the second bound b' is much more complicated we decided to abandon it.

In the next lemma, we derive bounds for the elements of the matrices $U_i, i = 1, \dots, n$. Note that even though we consider elements that are on or above the diagonal only, there is no loss of generality since the matrices $U_i, i = 1, \dots, n$, are symmetric.

Lemma 3.6 When joint i is prismatic, the elements on or above the diagonal of the matrix U_i satisfy the following inequalities:

- A. If $k > i$, $u_{j,k}^i = 0$.
- B. If joint k is prismatic and $k \neq i$, $u_{j,k}^i = 0$.
- C. If joint j is prismatic, $u_{j,k}^i = 0$.
- D. If joint j is revolute and $k = i$, $|u_{j,k}^i| \leq \bar{m}_i g \mu_j \mu_{j,i}$.
- E. If both joints j and k are revolute, and $j \leq k < i$, $|u_{j,k}^i| \leq \bar{m}_i g \mu_j \mu_{k,i} \bar{q}_i$.

Here, $u_{j,k}^i$ is the $(j, k)^{th}$ element of U_i and $j \leq k$ since we only considered elements on or above the diagonal. Out of the terms appearing in D and E, μ_j is defined in (3.18), \bar{q}_i is an upper bound for $|q_i|$, that is $|q_i| \leq \bar{q}_i$, and

$$\mu_{k,i} = \begin{cases} |\sin(\gamma_i - \gamma_k)| & : \text{ if for all revolute joints } l \text{ such that } k < l \leq i, \mathbf{z}_l \parallel \mathbf{z}_k, \\ 1 & : \text{ otherwise,} \end{cases} \quad (3.31)$$

where, as defined before, γ_l is the angle between \mathbf{z}_l and \mathbf{v}_c for all l such that $1 \leq l \leq n$. ◇

Proof of Lemma 3.6: In the proof we will consider each of the cases of Lemma 3.6 separately. From (3.8) we have the following expression for the j, k^{th} element of U_i :

$$u_{j,k}^i = \frac{\partial^2}{\partial q_j \partial q_k} \delta_i q_i \bar{m}_i g \mathbf{v}_c \cdot R_0^i \mathbf{z}_0 = \bar{m}_i g \mathbf{v}_c \cdot \frac{\partial^2}{\partial q_j \partial q_k} [R_0^i \mathbf{z}_0 q_i], \quad (3.32)$$

since joint i is prismatic in this lemma and hence, $\delta_i = 1$. It should be noted that the term to be differentiated, $R_0^i \mathbf{z}_0 q_i$ depends only on revolute joint variables l such that $l < i$ and the prismatic joint variable i . It is independent of all other joint variables.

Proof of A: In this case, $u_{j,k}^i = 0$, since the term to be differentiated is independent of the joint variable q_k .

Proof of B: Obvious since the term to be differentiated is independent of the joint variable q_k .

Proof of C: If $j \neq i$, $u_{j,k}^i = 0$, since the term to be differentiated is independent of the joint variable q_j . If $j = i$, since $k \geq j$, $k \geq i$. When $k = i$, $u_{j,k}^i = 0$ since $\frac{\partial^2 [R_0^i \mathbf{z}_0 q_i]}{\partial q_j \partial q_k} = \frac{\partial^2 [R_0^i \mathbf{z}_0 q_i]}{\partial q_i^2} = 0$. When $k > i$, $u_{j,k}^i = 0$ as we have Case A.

Proof of D: In this case we can apply Lemma 2.1 to (3.32) to obtain the following:

$$\begin{aligned} u_{j,k}^i &= \bar{m}_i g \mathbf{v}_c \cdot \frac{\partial^2}{\partial q_j \partial q_i} [R_0^i \mathbf{z}_0 q_i] = \bar{m}_i g \mathbf{v}_c \cdot \frac{\partial}{\partial q_j} [R_0^i \mathbf{z}_0] \\ &= \bar{m}_i g \mathbf{v}_c \cdot [\mathbf{z}_j \times R_0^i \mathbf{z}_0] = \bar{m}_i g \mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_i). \end{aligned} \quad (3.33)$$

Our next objective is to derive an expression for the triple scalar product in the above equation in terms of the angles between \mathbf{v}_c , \mathbf{z}_j and \mathbf{z}_i . We begin by partitioning \mathbf{v}_c into two vectors, based on coordinate frame j . The first component, \mathbf{v}_{cxy} is along the $(x - y)$ plane of coordinate frame j and the second component \mathbf{v}_{cz} is parallel to \mathbf{z}_j (See Figure 3.1). Therefore, $\mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_i) = \mathbf{v}_{cxy} \cdot (\mathbf{z}_j \times \mathbf{z}_i) + \mathbf{v}_{cz} \cdot (\mathbf{z}_j \times \mathbf{z}_i)$. Now since $(\mathbf{z}_j \times \mathbf{z}_i)$ is perpendicular to \mathbf{z}_j and since \mathbf{v}_{cz} is parallel to \mathbf{z}_j , \mathbf{v}_{cz} is perpendicular to $(\mathbf{z}_j \times \mathbf{z}_i)$. Hence $\mathbf{v}_{cz} \cdot (\mathbf{z}_j \times \mathbf{z}_i) = 0$. Therefore, $\mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_i) = \mathbf{v}_{cxy} \cdot (\mathbf{z}_j \times \mathbf{z}_i)$. By using Schwarz' inequality we get $|\mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_i)| = \|\mathbf{v}_{cxy}\| \|\mathbf{z}_j \times \mathbf{z}_i\|$. It can be seen

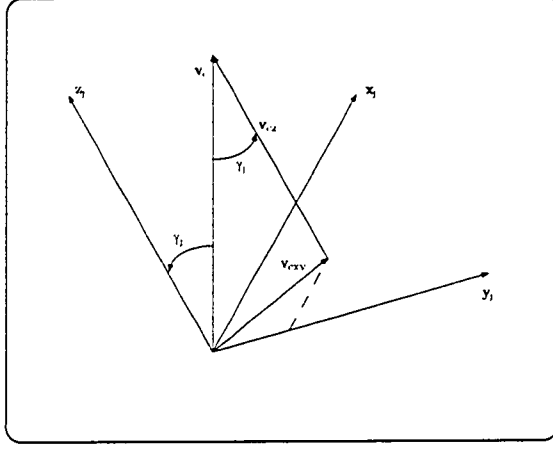


Figure 3.1: Partitioning of \mathbf{v}_c in Coordinate Frame j

from Figure 3.1, that $\|\mathbf{v}_{cxy}\| = |\sin(\gamma_j)|$ since $\|\mathbf{v}_c\| = 1$, due to geometry. Now since $\|\mathbf{z}_j \times \mathbf{z}_i\| = \|\mathbf{z}_j\| \|\mathbf{z}_i\| |\sin(\gamma_j - \gamma_i)| = |\sin(\gamma_j - \gamma_i)|$, by substituting in (3.33),

$$|u_{j,k}^i| \leq \overline{m}_i g |\sin(\gamma_j)| |\sin(\gamma_j - \gamma_i)|. \quad (3.34)$$

Now since $|\sin(\gamma_j)| \leq 1$ and $|\sin(\gamma_j - \gamma_i)| \leq 1$, we define μ_j and $\mu_{j,i}$ to be constant upper bounds for $|\sin(\gamma_j)|$ and $|\sin(\gamma_j - \gamma_i)|$ by defining them as in (3.18) and (3.31). Note that when, for all revolute joints l such that $k < l \leq i$, $\mathbf{z}_l \parallel \mathbf{z}_k$, we can use the same argument that was used in proving, $\overline{d}_{k,i}$ is constant in (3.20), to prove that $\|\mathbf{z}_j \times \mathbf{z}_i\|$ is constant. Note further that the first case of (3.18) is irrelevant here since joint j is revolute. The result follows immediately for this case by substituting in (3.34).

Proof of E: In this case we can apply Lemma 3.1 to (3.32) to obtain the following:

$$\begin{aligned} u_{j,k}^i &= \overline{m}_i g \mathbf{v}_c \cdot \frac{\partial^2}{\partial q_j \partial q_k} [R_0^i \mathbf{z}_0 q_i] \\ &= \overline{m}_i g \mathbf{v}_c \cdot [\mathbf{z}_j \times (\mathbf{z}_k \times R_0^i \mathbf{z}_0)] q_i \\ &= \overline{m}_i g (\mathbf{v}_c \times \mathbf{z}_j) \cdot (\mathbf{z}_k \times \mathbf{z}_i) q_i. \end{aligned} \quad (3.35)$$

Now we can use Shwarz' inequality to obtain the following:

$$|u_{j,k}^i| \leq \bar{m}_i g \|\mathbf{v}_c \times \mathbf{z}_j\| \|\mathbf{z}_k \times \mathbf{z}_i\| |q_i|.$$

The result follows immediately since $\|\mathbf{v}_c \times \mathbf{z}_j\| = \|\mathbf{v}_c\| \|\mathbf{z}_j\| |\sin \gamma_j| \leq \mu_j$, $\|\mathbf{z}_k \times \mathbf{z}_i\| \leq \mu_{k,i}$ and $|q_i| \leq \bar{q}_i$. This completes the proof of Lemma 3.6. \square

3.2.3 Class \mathcal{BGJ} Robots

At this point we present the class of robots for which the gravity Jacobian is uniformly bounded. In Section 3.2.4, we will prove, in Theorem 3.1, that the necessary and sufficient condition for a robot to have a uniformly bounded Hessian is that it belong to Class \mathcal{BGJ} . The classification of the class of robots with bounded Hessian is the first major contribution of this chapter.

Definition 3.2 Class \mathcal{BGJ} consists of robots with the following joint configurations:

- 3.1. There does not exist a revolute joint j and a prismatic joint i such that $j < i$.
- 3.2. If there exists a revolute joint j and a prismatic joint i such that $j < i$, then at least one of the following conditions must be satisfied.
 - (a) The axis of rotation of joint j and the axis of translation of joint i are always parallel.
 - (b) The axis of rotation of joint j is always parallel to \mathbf{v}_c .

\diamond

The following interpretation of Class \mathcal{BGJ} robots is more practical in order to determine whether a given robot belongs to Class \mathcal{BGJ} .

Robots of Class \mathcal{BGJ} : Robots of this class have any of the following joint configurations:

- 3.1. All joints are prismatic ($\mathcal{PP}, \dots, \mathcal{PP}$).
- 3.2. All joints are revolute ($\mathcal{RR}, \dots, \mathcal{RR}$).
- 3.3. A series of prismatic joints followed by a series of revolute joints ($\mathcal{PP}, \dots, \mathcal{PR}, \dots, \mathcal{RR}$).
- 3.4. Configurations such that for all revolute joints j preceding a prismatic joint i , one of the following is true:
 - (a) The axis of rotation of joint j and the axis of translation of joint i are always parallel.
 - (b) The axis of rotation of joint j is always parallel to \mathbf{v}_c .

An important point to note is that the class of robots for which the inertia matrix $\mathbf{D}(\mathbf{q})$ is bounded, Class \mathcal{BD} [31], is a subclass of Class \mathcal{BGJ} .

3.2.4 Necessary and Sufficient Conditions For the Hessian to be Uniformly Bounded

Since $\|B\|$ is uniformly bounded, we see from (3.9) that the gravity Jacobian is uniformly bounded for robots for which $U_i = 0$, $i = 1, \dots, n$. We see from the next lemma that this is the case for robots of Class \mathcal{BGJ} .

Lemma 3.7 For robots of Class \mathcal{BGJ} , $U_i = 0$, for $i = 1, \dots, n$. \diamond

Proof of Lemma 3.7: We prove $U_i = 0$ by proving that each of its elements are zero. If joint i is revolute, by the definition of δ_i , $U_i = 0$. Therefore, for the rest of this proof we will only consider the case corresponding to joint i being prismatic. In Lemma 3.6, we considered five cases, which covered all possibilities, and derived

bounds for each of these cases. In this proof, we will show that $u_{j,k}^i = 0$ for each of these cases for Class \mathcal{BGJ} robots. Since $u_{j,k}^i = 0$ for cases A, B, and C, we will only consider the remaining two cases, D and E.

Case D: When joint j is revolute and $k = i$, from (3.33), $u_{j,k}^i = \overline{m}_i g[\mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_i)]$. Since joint j is a revolute joint preceding a prismatic joint i , from the condition for belonging to Class \mathcal{BGJ} , either \mathbf{z}_j and \mathbf{z}_i are parallel or \mathbf{z}_j and \mathbf{v}_c are parallel. When \mathbf{z}_j and \mathbf{z}_i are parallel, $u_{j,k}^i = 0$ since $(\mathbf{z}_j \times \mathbf{z}_i) = \mathbf{0}$. When \mathbf{z}_j and \mathbf{v}_c are parallel, $u_{j,k}^i = \overline{m}_i g[\mathbf{v}_c \cdot (\mathbf{z}_j \times \mathbf{z}_i)] = \overline{m}_i g[(\mathbf{v}_c \times \mathbf{z}_j) \cdot \mathbf{z}_i] = 0$ since $(\mathbf{v}_c \times \mathbf{z}_j) = \mathbf{0}$. This completes the proof for this case.

Case E: When both joints j and k are revolute and $j \leq k < i$, from (3.35), $u_{j,k}^i = \overline{m}_i g[(\mathbf{v}_c \times \mathbf{z}_j) \cdot (\mathbf{z}_k \times \mathbf{z}_i)] q_i$. Since joint k is a revolute joint preceding a prismatic joint i , from the conditions for belonging to Class \mathcal{BGJ} , either \mathbf{z}_k and \mathbf{z}_i are always parallel or \mathbf{z}_k is always parallel to \mathbf{v}_c . When \mathbf{z}_k and \mathbf{z}_i are parallel, $u_{j,k}^i = 0$ since $(\mathbf{z}_k \times \mathbf{z}_i) = \mathbf{0}$. When \mathbf{z}_k is always parallel to \mathbf{v}_c , from Lemma 3.3, \mathbf{z}_j is also always parallel to \mathbf{v}_c . Hence, $(\mathbf{v}_c \times \mathbf{z}_j) = \mathbf{0}$ and consequently $u_{j,k}^i = 0$. This completes the proof for this case.

Note that even though we only considered elements that are on or above the diagonal in this proof, there is no loss of generality since U_i is symmetric. This completes the proof of this lemma.

□

We are ready now to present our next major result.

Theorem 3.1 For serial link robots described by (1.1), $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is uniformly bounded if and only if the robot belongs to class \mathcal{BGJ} . ◇

Proof of Theorem 3.1: We have to prove that for a given robot, $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is uniformly bounded if and only if the robot belongs to Class \mathcal{BGJ} . Clearly, the “if” part follows immediately from Lemma 3.7, Lemma 3.5 and (3.9).

Proof of the “only if” part: We have to show that if $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is uniformly bounded then the robot belongs to Class \mathcal{BGJ} or equivalently, we have to show that if the robot does not belong to Class \mathcal{BGJ} , then $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is not uniformly bounded.

Suppose the robot does not belong to Class \mathcal{BGJ} . Therefore, there exists a revolute joint j and a prismatic joint i such that $\mathbf{z}_j \times \mathbf{v}_c \neq 0$ and $\mathbf{z}_j \times \mathbf{z}_i \neq 0$ at some point (say $\bar{\mathbf{q}}$) in the workspace. Consider the set $W_j(\bar{\mathbf{q}})$. Note that since $q_1 = \bar{q}_1, q_2 = \bar{q}_2, \dots, q_{j-1} = \bar{q}_{j-1}$ are constant, \mathbf{z}_j remains constant throughout $W_j(\bar{\mathbf{q}})$. Therefore, \mathbf{z}_j will remain non parallel to \mathbf{v}_c . Furthermore, since $q_{j+1} = \bar{q}_{j+1}, q_{j+2} = \bar{q}_{j+2}, \dots, q_n = \bar{q}_n$ remain constant, R_j^i is also constant in $W_j(\bar{\mathbf{q}})$. Therefore, \mathbf{z}_j and \mathbf{z}_i remain non parallel also. Hence, throughout $W_j(\bar{\mathbf{q}})$ the following two conditions are satisfied:

$$\mathbf{z}_j \times \mathbf{v}_c \neq \mathbf{0}, \quad (3.36)$$

$$\mathbf{z}_j \times \mathbf{z}_i \neq \mathbf{0}. \quad (3.37)$$

Now consider the j^{th} diagonal element $u_{j,j}^i$, of U_i . By applying Lemma 3.1 to (3.8) we get the following:

$$\begin{aligned} u_{j,j}^i &= \frac{\partial^2}{\partial q_j^2} \bar{m}_i g \delta_i q_i \mathbf{v}_c \cdot (R_0^i \mathbf{z}_0) \\ &= \bar{m}_i g \delta_i q_i \mathbf{v}_c \cdot [\mathbf{z}_j \times (\mathbf{z}_j \times \mathbf{z}_i)], \end{aligned}$$

Now by expanding the triple vector product we get,

$$\begin{aligned} &\mathbf{v}_c \cdot [\mathbf{z}_j \times (\mathbf{z}_j \times \mathbf{z}_i)] \\ &= \mathbf{v}_c \cdot [(\mathbf{z}_j \cdot \mathbf{z}_i) \mathbf{z}_j - (\mathbf{z}_j \cdot \mathbf{z}_j) \mathbf{z}_i] \\ &= (\mathbf{z}_j \cdot \mathbf{z}_i) (\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_j). \end{aligned}$$

Substituting the above in the expression for $u_{j,j}^i$ we get,

$$u_{j,j}^i = \bar{m}_i g \delta_i q_i [(\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_i)]. \quad (3.38)$$

Claim: There are points in $W_j(\bar{\mathbf{q}})$ such that $(\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_i) \neq 0$.

Proof of Claim: We Prove this by contradiction. Suppose there are no points in $W_j(\bar{\mathbf{q}})$ such that $(\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_i) \neq 0$. Then,

$$\mathbf{v}_c \cdot \mathbf{z}_i = (\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j), \quad (3.39)$$

throughout $W_j(\bar{\mathbf{q}})$. Now since \mathbf{z}_j is constant throughout $W_j(\bar{\mathbf{q}})$, $\mathbf{v}_c \cdot \mathbf{z}_j$ is constant throughout $W_j(\bar{\mathbf{q}})$. Since R_j^i is constant, $\mathbf{z}_j \cdot \mathbf{z}_i$ is also constant throughout $W_j(\bar{\mathbf{q}})$. Therefore, since the right hand side of (3.39) is constant, $\mathbf{v}_c \cdot \mathbf{z}_i$ is constant in $W_j(\bar{\mathbf{q}})$. Since we have (3.37) satisfied in $W_j(\bar{\mathbf{q}})$, from Lemma 3.2, we get that $\mathbf{v}_c \times \mathbf{z}_j = 0$ throughout $W_j(\bar{\mathbf{q}})$ which contradicts (3.36). This completes the proof of this claim.

Now consider the j^{th} diagonal element of $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$, denoted by $p_{j,j}$. Using (3.9),

$$\begin{aligned} p_{j,j} &= \sum_{l=1}^n u_{j,j}^l + b_{j,j} \\ &= u_{j,j}^i + \sum_{l=1}^{i-1} u_{j,j}^l + \sum_{l=i+1}^n u_{j,j}^l + b_{j,j} \\ &= \bar{m}_i g [(\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_i)] \delta_i q_i + \sum_{l=1}^{i-1} u_{j,j}^l + \sum_{l=i+1}^n u_{j,j}^l + b_{j,j} \\ &= \beta_j q_i + \alpha_j, \end{aligned}$$

where $\alpha_j = b_{j,j} + \sum_{l=1}^{i-1} u_{j,j}^l + \sum_{l=i+1}^n u_{j,j}^l$ ($b_{j,j}$ is the j^{th} diagonal element of B) and $\beta_j = \bar{m}_i g [(\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_i)]$ (since $\delta_i=1$). We proved above, that there are points within the robot workspace such that $(\mathbf{z}_j \cdot \mathbf{z}_i)(\mathbf{v}_c \cdot \mathbf{z}_j) - (\mathbf{v}_c \cdot \mathbf{z}_i) \neq 0$. Now since $\bar{m}_i g$ is non zero ($\bar{m}_i g$ can be zero only if all of the links beyond and including link i have zero mass, which is not physically realistic) there are points within the workspace where $\beta_j \neq 0$. Now since both α_j and β_j are independent of q_i , by keeping

all other joints fixed except for joint i , we can keep both α_j and β_j constant. Now by selecting q_i to be sufficiently large, we can make $p_{j,j}$ arbitrarily large. Therefore any uniform bound that is proposed (a uniform bound has to be valid throughout \mathfrak{R}^n) for $\|\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}\|$ can be invalidated by selecting q_i sufficiently large. Therefore, $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ is not uniformly bounded for robots outside Class \mathcal{BGJ} . This completes the proof of this theorem. \square

3.2.5 Examples

In this section we look at three examples to demonstrate Theorem 3.1. For all three examples we choose robots consisting of a revolute joint followed by a prismatic joint. Based on Definition 3.2, we see that such robots will belong to Class \mathcal{BGJ} only if at least one of the two conditions, 3.2a and 3.2b are satisfied. For the first example we will select a robot that violates both these conditions and derive an explicit expression for the Hessian. We will see that the Hessian is not uniformly bounded for this robot as predicted by Theorem 3.1. For the second example we will change the relative orientations of the joints of this same robot so that it satisfies Condition 3.2a. As we will see, the Hessian becomes uniformly bounded for this case as expected from Theorem 3.1. In the third example we consider the same robot that we considered in the first example and orient it such that Condition 3.2b is satisfied. We will see that this also causes the Hessian to become uniformly bounded as predicted by Theorem 3.1.

3.2.5.1 An \mathcal{RP} Robot of Class $\overline{\mathcal{BGJ}}$

Consider the revolute-prismatic (\mathcal{RP}) robot of Figure 3.2. This robot does not belong to Class \mathcal{BGJ} since there exists a revolute joint $j(= 1)$ and a prismatic joint $i(= 2)$ such that $j < i$, the axes of the two joints are non parallel, and the axis of joint j is not always parallel to \mathbf{v}_c . The kinematic link parameters are given in Table 3.1 which

is organized as follows: Columns three to six contain the standard DH parameters while column seven contains the masses of each of the links. Finally, the last three columns contain the x , y and z components of the position vector of the center of mass c_i . The potential energy of this robot is given by

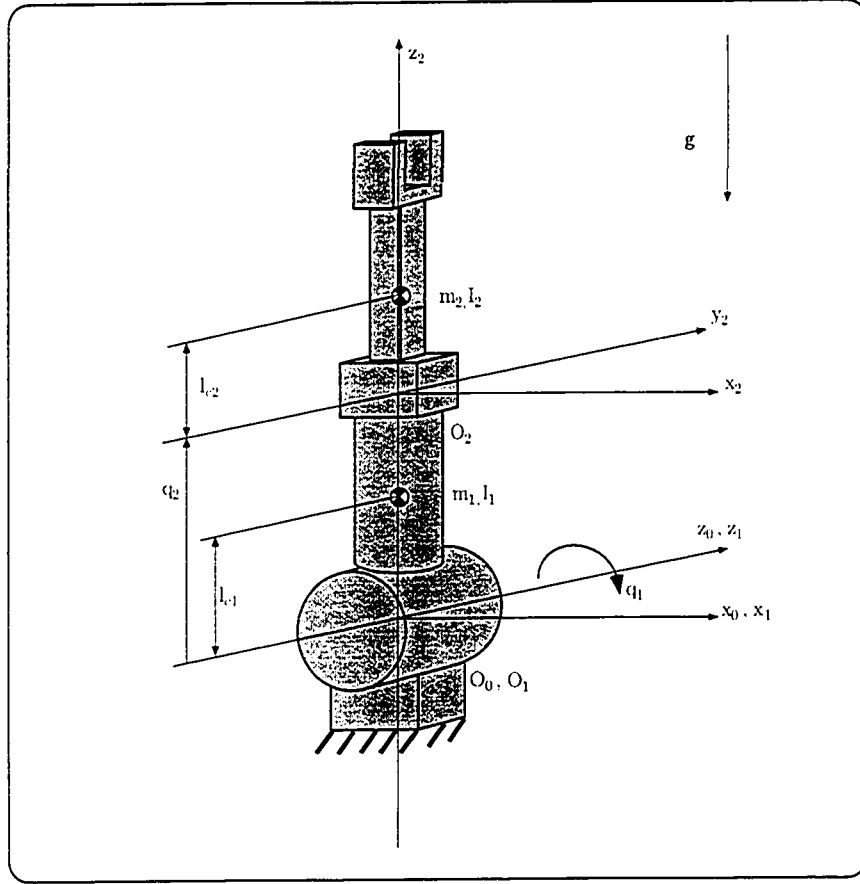


Figure 3.2: The \mathcal{RP} Robot of Example 3.2.5.1

$$V = m_1 g c_{y1} \cos q_1 + m_2 g (q_2 + c_{z2}) \cos q_1.$$

It follows that the Hessian is given by,

$$\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} -m_1 g c_{1y} \cos(q_1) - m_2 g (q_2 + c_{2z}) \cos(q_1) & -m_2 g \sin(q_1) \\ -m_2 g \sin(q_1) & 0 \end{bmatrix}. \quad (3.40)$$

Link i	Joint Type	α_{i-1}	θ_i	a_{i-1}	d_i	m_i	c_{ix}	c_{iy}	c_{iz}
1	revolute	0	q_1	0	0	m_1	0	c_{1y}	0
2	prismatic	90	0	0	q_2	m_2	0	0	c_{2z}

Table 3.1: Kinematic Link Parameters of \mathcal{RP} Robot of Figure 3.2

It can be seen that the element $[-m_1gc_{1y}\cos(q_1) - m_2g(q_2 + c_{2z})\cos(q_1)]$ can be made arbitrarily large by selecting q_2 sufficiently large. Therefore, $\|\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}\|$ can be made arbitrarily large when $\mathbf{q} \in \mathbb{R}^n$. Hence, the Hessian is not uniformly bounded for this robot.

3.2.5.2 An \mathcal{RP} Robot with Parallel Joint Axes

For this example we consider the same robot that we considered in Section 3.2.5.1 and change the orientation of the axis of Joint 1 with respect to the axis of Joint 2 (See Figure 3.3). The Kinematic parameters for this robot are given in Table 3.2. Note that the only revolute joint j ($= 1$) and prismatic joint k ($= 2$) such that $j < k$ satisfies Condition 3.2a (since the axis of rotation of joint j and the axis of translation of joint k are parallel). Therefore, this robot belongs to Class \mathcal{BGJ} . Next we derive an explicit expression for the Jacobian of the gravity vector for this robot. The potential energy is given by

Link i	Joint Type	α_{i-1}	θ_i	a_{i-1}	d_i	m_i	c_{ix}	c_{iy}	c_{iz}
1	revolute	0	q_1	0	0	m_1	0	c_{1y}	0
2	prismatic	0	0	a_1	q_2	m_2	0	0	c_{2z}

Table 3.2: Kinematic Link Parameters of \mathcal{RP} Robot of Figure 3.2

$$V = m_1gc_{1y}\cos q_1 + m_2ga_1\cos q_1.$$

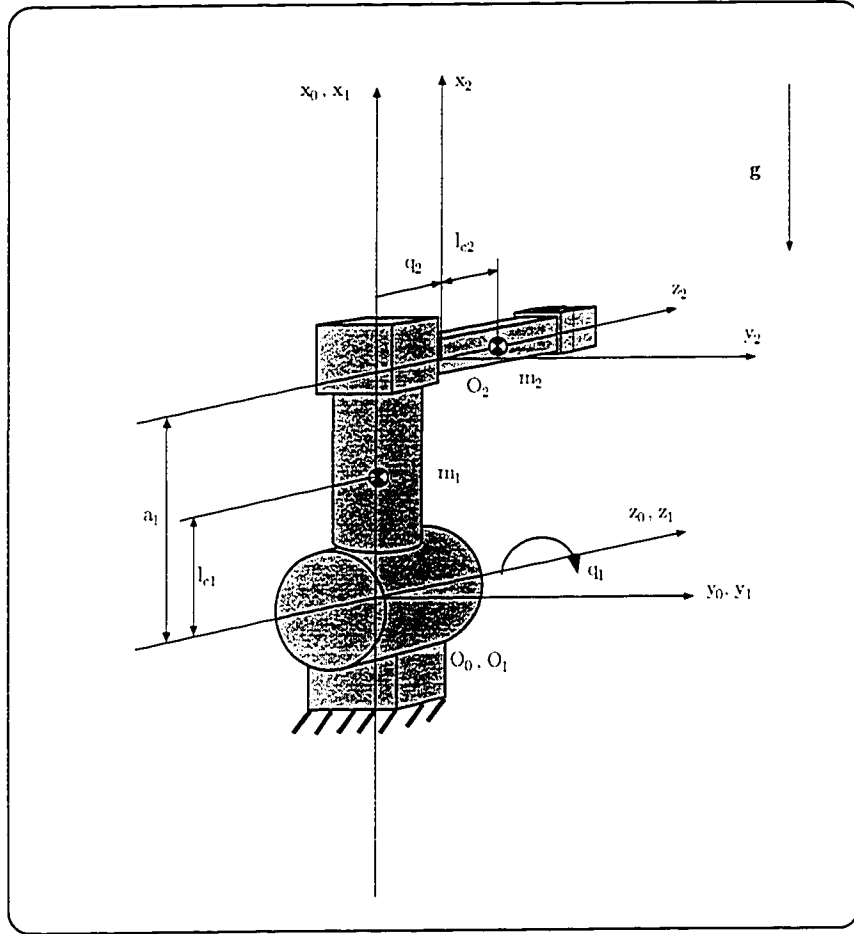


Figure 3.3: The \mathcal{RP} Robot of Example 3.2.5.2

Hence, the Jacobian of the gravity vector is given by,

$$\frac{\partial \mathbf{g}}{\partial \mathbf{q}} = \begin{bmatrix} -(m_1 g c_{1y} + m_2 g a_1) \cos q_1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Therefore, $\|\frac{\partial \mathbf{g}}{\partial \mathbf{q}}\| = |(m_1 g c_{1y} + m_2 g a_1) \cos q_1|$ is uniformly bounded for this robot as predicted by Theorem 3.1.

3.2.5.3 An \mathcal{RP} Robot with the Revolute Joint Axis Vertical

For this example we take the same robot as in Figure 3.2 and orient it such that the axis of rotation of the first joint is always vertical (see Figure 3.4). Since the only revolute joint j ($= 1$) and prismatic joint k ($= 2$) such that $j < k$ satisfies one of the required conditions, namely, the axis of rotation of joint j is always vertical, this robot also belongs to class \mathcal{BJG} . We consider next the Jacobian of the gravity vector for this robot. It can be seen that the center of mass locations of both joints remain on the same horizontal plane as the joints are moved. Hence, for this robot, the potential energy is constant. Therefore, the Hessian becomes zero. Hence, as expected, $\frac{\partial \mathbf{g}}{\partial \mathbf{q}}$ is uniformly bounded for this robot. We see from this example that in addition to the relative orientation of the joint axes, the orientation of the robot in the gravitational field also plays an important role in the uniform boundedness of the Jacobian of the gravity vector.

3.3 A Uniform Bound for Class \mathcal{BJG} Robots

In this section we present a uniform bound for the Jacobian of the gravity vector for Class \mathcal{BJG} robots and propose an algorithm to compute the uniform bound. This procedure is illustrated with an example.

3.3.1 An Explicit Expression for the Uniform Bound β Satisfying (1.2)

Theorem 3.2 For Class \mathcal{BJG} robots, the uniform bound $\beta = b$ defined in (3.17) satisfies (1.2). \diamond

Proof of Theorem 3.2: From Lemma 3.7 and (3.9), we see that for Class \mathcal{BJG} robots $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} = B$. The result follows immediately from Lemma 3.5. \square

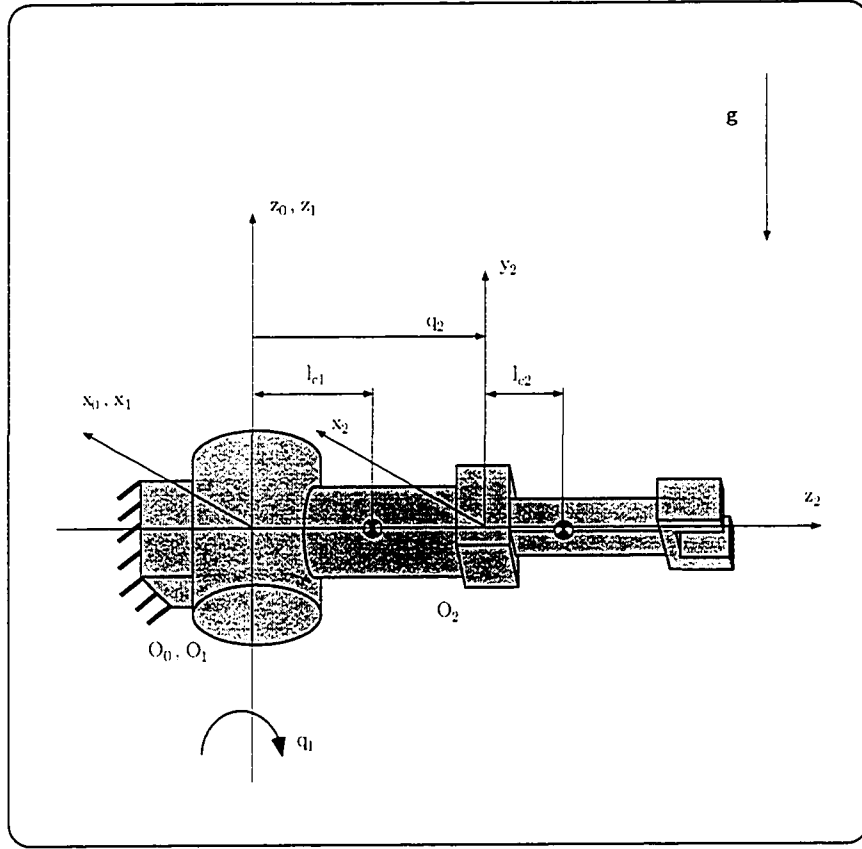


Figure 3.4: The \mathcal{RP} Robot of Example 3.2.5.3

3.3.2 Algorithm for Computing the Uniform Bound for the Hessian

We propose the following steps for computing the uniform bound β satisfying (1.2) for a robot once it is established that it is an element of Class \mathcal{BGJ} :

- 3.1. Assign the coordinate frames $0, \dots, n$, according to the modified DH convention of [14].
 - (a) The vector \mathbf{d}'_i is the position vector of O_i (the origin of frame i) in coordinate frame $(i - 1)$ with all prismatic joints in zero position.

- (b) The vector \mathbf{c}_i is the position vector of the center of mass of link i in coordinate frame i .

- 3.2. Determine the values of μ_j , $j = 1, \dots, n$, using (3.18).
- 3.3. Obtain the uniform bound of (3.17) in terms of \bar{b}_k , $k = 1, \dots, n$, by substituting for each μ_j .
- 3.4. Calculate the constants $\bar{d}_{k,i}$, $i = k+1, \dots, n$, and $\bar{c}_{k,i}$, $i = k, \dots, n$, corresponding to each \bar{b}_k appearing in the expression for the uniform bound using (3.20) and (3.21) respectively.
- 3.5. Calculate each \bar{b}_k and substitute in the expression for the uniform bound, equation (3.17).

3.3.3 Example: PUMA560

In this section we compute the uniform bound β for the six d.o.f PUMA560 robot (see Figure 3.5). The purpose of this example is to illustrate the computational procedure proposed in the previous section and to provide a comparison of the uniform bound with the exact value of $\left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\|$ for a typical industrial robot. In the literature, there are several estimates of the dynamic parameters for the PUMA560. For this example we use the estimates given in [4]. The kinematic link parameters for the PUMA560 are given in Table 3.3 which has the same format as Table 3.1. The parameters a_i , d_i , c_{ix} , c_{iy} , and c_{iz} are given in *meters* while the parameters α_i and θ_i are given in *degrees*. The masses m_i are expressed in *kilograms*. It can be seen immediately that the PUMA belongs to Class \mathcal{BGJ} since it does not contain any prismatic joints. Therefore we can use Theorem 3.2 to obtain a uniform bound for the Hessian of the PUMA560 robot. Next we will follow the procedure outlined in Section 3.3.2 to compute the uniform bound satisfying (1.2) for the PUMA560.

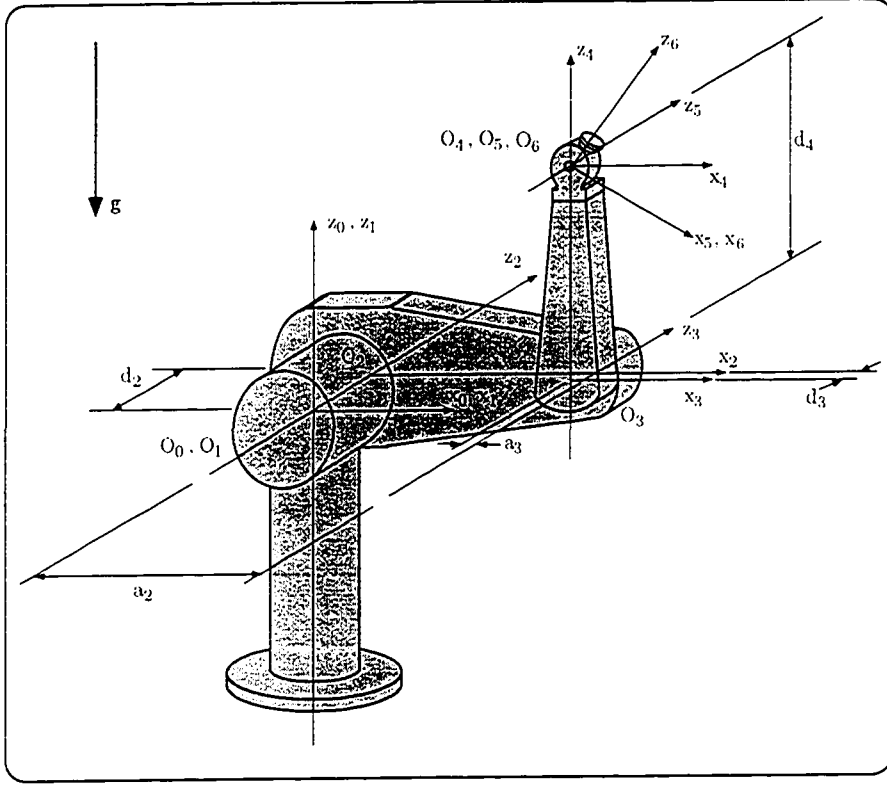


Figure 3.5: The PUMA560 Robot

3.1. Since the axis of the first joint is always parallel to \mathbf{v}_c , $\mu_1 = 0$. Since none of the other joint axes are fixed in the base coordinate frame, $\mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = 1$.

3.2. By substituting the values above into (3.17), we get the following expression:

$$\begin{aligned}
 b^2 &= (\bar{b}_2)^2 + 2 \left[(\bar{b}_3)^2 + (\bar{b}_4)^2 + (\bar{b}_5)^2 + (\bar{b}_6)^2 \right] + \\
 &\quad (\bar{b}_3)^2 + 2 \left[(\bar{b}_4)^2 + (\bar{b}_5)^2 + (\bar{b}_6)^2 \right] + \\
 &\quad (\bar{b}_4)^2 + 2 \left[(\bar{b}_5)^2 + (\bar{b}_6)^2 \right] + (\bar{b}_5)^2 + 2 \left[(\bar{b}_6)^2 \right] + (\bar{b}_6)^2 \\
 &= (\bar{b}_2)^2 + 3(\bar{b}_3)^2 + 5(\bar{b}_4)^2 + 7(\bar{b}_5)^2 + 9(\bar{b}_6)^2.
 \end{aligned} \tag{3.41}$$

3.3. Next we calculate the $\bar{d}_{k,i}$ corresponding to $k = 2, \dots, 6$, from (3.20) as follows:

Link i	Joint Type	α_{i-1}	θ_i	a_{i-1}	d_i	m_i	c_{ix}	c_{iy}	c_{iz}
1	revolute	0	q_1	0	0	-	-	-	-
2	revolute	-90	q_2	0	0.244	17.4	0.068	0.006	-0.016
3	revolute	0	q_3	0.432	-0.093	4.8	0	-0.070	0.014
4	revolute	90	q_4	-0.203	0.433	0.8	0	0	-0.019
5	revolute	-90	q_5	0	0	0.3	0	0	0
6	revolute	90	q_6	0	0	0.1	0	0	0.032

Table 3.3: Kinematic Link Parameters of the PUMA560

$$(a) \quad \bar{d}_{2,3} = \|\mathbf{z}_2 \times (R_0^2 \mathbf{d}'_3)\| = |a_2| = 0.432, \bar{d}_{2,4} = \|\mathbf{d}'_4\| = \sqrt{(a_3)^2 + (d_4)^2} = 0.433,$$

$$\bar{d}_{2,5} = \|\mathbf{d}'_5\| = 0, \bar{d}_{2,6} = \|\mathbf{d}'_6\| = 0.$$

$$(b) \quad \bar{d}_{3,4} = \|\mathbf{d}'_4\| = 0.433, \bar{d}_{3,5} = \|\mathbf{d}'_5\| = 0, \bar{d}_{3,6} = \|\mathbf{d}'_6\| = 0.$$

$$(c) \quad \bar{d}_{4,5} = \|\mathbf{d}'_5\| = 0, \bar{d}_{4,6} = \|\mathbf{d}'_6\| = 0.$$

$$(d) \quad \bar{d}_{5,6} = \|\mathbf{d}'_6\| = 0.$$

We next use (3.21) to calculate $\bar{c}_{k,i}$ for $k = 2, \dots, 6$, as follows:

$$(a) \quad \bar{c}_{2,2} = \|\mathbf{z}_2 \times (R_0^2 \mathbf{c}_2)\| = \sqrt{(c_{2x})^2 + (c_{2y})^2} = 0.068, \bar{c}_{2,3} = \|\mathbf{z}_2 \times R_0^3 \mathbf{c}_3\| = \sqrt{(c_{3x})^2 + (c_{3y})^2} = 0.07, \bar{c}_{2,4} = \|\mathbf{c}_4\| = 0.019, \bar{c}_{2,5} = \|\mathbf{c}_5\| = 0, \bar{c}_{2,6} = \|\mathbf{c}_6\| = 0.032.$$

$$(b) \quad \bar{c}_{3,3} = \|\mathbf{z}_3 \times R_0^3 \mathbf{c}_3\| = \sqrt{(c_{3x})^2 + (c_{3y})^2} = 0.07, \bar{c}_{3,4} = \|\mathbf{c}_4\| = 0.019, \bar{c}_{3,5} = \|\mathbf{c}_5\| = 0, \bar{c}_{3,6} = \|\mathbf{c}_6\| = 0.032.$$

$$(c) \quad \bar{c}_{4,4} = \|\mathbf{z}_4 \times R_0^4 \mathbf{c}_4\| = \sqrt{(c_{4x})^2 + (c_{4y})^2} = 0, \bar{c}_{4,5} = \|\mathbf{c}_5\| = 0, \bar{c}_{4,6} = \|\mathbf{c}_6\| = 0.032.$$

$$(d) \quad \bar{c}_{5,5} = \|\mathbf{z}_5 \times R_0^5 \mathbf{c}_5\| = \sqrt{(c_{5x})^2 + (c_{5y})^2} = 0, \bar{c}_{5,6} = \|\mathbf{c}_6\| = 0.032.$$

$$(e) \quad \bar{c}_{6,6} = \|\mathbf{z}_6 \times R_0^6 \mathbf{c}_6\| = \sqrt{(c_{6x})^2 + (c_{6y})^2} = 0.$$

3.4. Now we substitute the above values into (3.19) to obtain the value of each \bar{b}_k .

From (3.4), we see that $\bar{m}_2 = 23.4$, $\bar{m}_3 = 6.0$, $\bar{m}_4 = 1.2$, $\bar{m}_5 = 0.4$ and $\bar{m}_6 = 0.1$.

$$(a) \quad \bar{b}_2 = (\bar{m}_3 \bar{d}_{2,3} + \bar{m}_4 \bar{d}_{2,4} + \bar{m}_5 \bar{d}_{2,5} + \bar{m}_6 \bar{d}_{2,6} + m_2 \bar{c}_{2,2} + m_3 \bar{c}_{2,3} + m_4 \bar{c}_{2,4} + m_5 g \bar{c}_{2,5} + m_6 \bar{c}_{2,6})g = 45.56.$$

$$(b) \quad \bar{b}_3 = (\bar{m}_4 g \bar{d}_{3,4} + \bar{m}_5 \bar{d}_{3,5} + \bar{m}_6 \bar{d}_{3,6} + m_3 \bar{c}_{3,3} + m_4 \bar{c}_{3,4} + m_5 \bar{c}_{3,5} + m_6 \bar{c}_{3,6})g = 8.58.$$

$$(c) \quad \bar{b}_4 = (\bar{m}_5 \bar{d}_{4,5} + \bar{m}_6 \bar{d}_{4,6} + m_4 \bar{c}_{4,4} + m_5 \bar{c}_{4,5} + m_6 \bar{c}_{4,6})g = 0.03.$$

$$(d) \quad \bar{b}_5 = (\bar{m}_6 \bar{d}_{4,6} + m_5 \bar{c}_{5,5} + m_6 \bar{c}_{5,6})g = 0.03.$$

$$(e) \quad \bar{b}_6 = (m_6 \bar{c}_{6,6})g = 0.$$

Now by substituting in (3.41), we get $\beta = b = 47.92$.

In order to compare the uniform bound just computed with the exact value of $\|\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}\|$, we derived an exact expression for $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial \mathbf{g}(q_2, q_3, q_4, q_5)}{\partial \mathbf{q}}$. It is independent of q_1 as expected since the axis of rotation of joint 1 is always parallel to the gravitational field. In order to locate the point where $\|\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}\|$ reaches a maximum, a complete enumeration of the robot workspace at a relatively low resolution (around 50 points for each joint resulting in a resolution of around 7 degrees) was done using Matlab [54]. We found $\|\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}\|$ to be maximum when $q_2 = q_3 = 90^\circ$ and $q_4 = q_5 = 180^\circ$ degrees. Figure 3.6 shows a plot of $\|\frac{\partial \mathbf{g}}{\partial \mathbf{q}}\|$ and the uniform bound β as q_2 is varied between 0° and 360° degrees while $q_3 = 90^\circ$ and $q_4 = q_5 = 180^\circ$.

3.4 A Bound for Class $\overline{\mathcal{BGJ}}$ Robots

In this section, we present a bound for the Hessian, that satisfies (1.4), for robots that belong to Class $\overline{\mathcal{BGJ}}$.

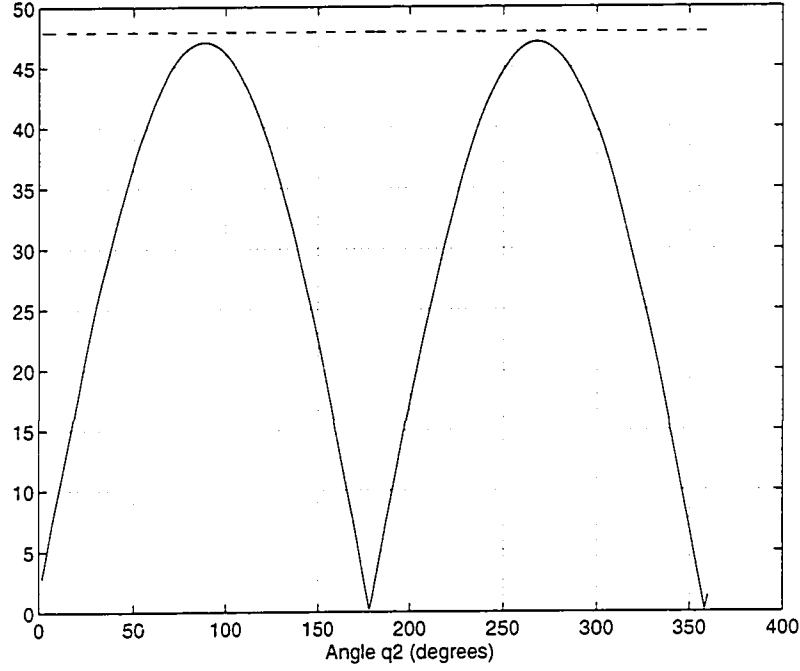


Figure 3.6: Uniform Bound β (dashed line) and $\left\| \frac{\partial \mathbf{g}}{\partial \mathbf{q}} \right\|$ (solid curve) for the PUMA560

3.4.1 An Explicit Expression for the Bound c_4 satisfying (1.4)

Theorem 3.3 For all robots considered in this thesis, a bound c_4 satisfying (1.4) is given by,

$$c_4 = b + \sum_{i=1}^n u_i, \quad (3.42)$$

where b is defined in (3.17) and

$$u_i = \begin{cases} \sqrt{\sum_{j=1}^i \left(2 \sum_{k=j+1}^i (\bar{u}_{j,k}^i)^2 + (\bar{u}_{j,j}^i)^2 \right)} & : \text{ if joint } i \text{ is prismatic} \\ 0 & : \text{ if joint } i \text{ is revolute.} \end{cases} \quad (3.43)$$

Here,

$$\bar{u}_{j,k}^i = \begin{cases} \bar{m}_i g \mu_j \mu_{j,i} & : \text{ if } k = i \text{ and joint } j \text{ is revolute} \\ \bar{m}_i g \mu_j \mu_{j,i} \bar{q}_i & : \text{ if both joints } j \text{ and } k \text{ are revolute and } j \leq k < i \\ 0 & : \text{ otherwise,} \end{cases} \quad (3.44)$$

where \bar{m}_i is defined in (3.4), μ_j is defined in (3.18), $\mu_{j,i}$ is defined in (3.31)

and \bar{q}_i is an upper bound for $|q_i|$ as defined previously. \diamond

Proof of Theorem 3.3: Following the same reasoning that is used to derive (3.23), we have

$$\begin{aligned} \text{Trace}(U_i^T U_i) &= \sum_{j=1}^n \left(2 \sum_{k=j+1}^n (u_{j,k}^i)^2 + (u_{j,j}^i)^2 \right) \\ &= \sum_{j=1}^i \left(2 \sum_{k=j+1}^i (\bar{u}_{j,k}^i)^2 + (\bar{u}_{j,j}^i)^2 \right), \end{aligned} \quad (3.45)$$

since from Lemma 3.6, $u_{j,k}^i = 0$ when $k > i$. Now by using Lemma 3.6 again, we see that $(u_{j,k}^i)^2 \leq (\bar{u}_{j,k}^i)^2$ where $\bar{u}_{j,k}^i$ is defined in (3.44). Now since $\|U_i\| \leq \sqrt{\text{Trace}(U_i^T U_i)}$, by substituting in (3.45), we have the following:

$$\|U_i\| \leq \sqrt{\sum_{j=1}^i \left(2 \sum_{k=j+1}^i (\bar{u}_{j,k}^i)^2 + (\bar{u}_{j,j}^i)^2 \right)} = u_i, \quad (3.46)$$

where u_i is defined in (3.43). By applying the triangle inequality to (3.9) we get,

$$\left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\| = \|B\| + \sum_{i=1}^n \|U_i\|.$$

The result follows by substituting from (3.46) and Lemma 3.5. \square

3.4.2 Example: The Bound c_4 for the \mathcal{RP} Robot of Section 3.2.5.1

In this section we choose a robot that does not belong to Class \mathcal{BGJ} to illustrate the computational procedure of the bound of Theorem 3.3. Consider the 2 d.o.f \mathcal{RP} robot of Section 3.2.5.1 (see Figure 3.2). As noted previously, this robot belongs to

Class $\overline{\mathcal{BGJ}}$. We compute the bound c_4 of (3.42) for this robot as follows: It can be seen that in order to compute the bound c_4 , we need to calculate b and u_i , $i = 1, 2$. We will first calculate b following the procedure suggested in Section 3.3.2.

3.1. Since joint 2 is prismatic, $\mu_2 = 0$ and since joint 1 is revolute and $\gamma_1 = 90^\circ$, $\mu_1 = 1$.

3.2. By substituting the values above into (3.17), we get the following expression:

$$b = \sqrt{(\bar{b}_1)^2 + 2(\bar{b}_2)^2} \quad (3.47)$$

3.3. From (3.20) we get $\bar{d}_{1,2} = \|\mathbf{d}_i\| = 0$. We use (3.21) next to obtain $\bar{c}_{1,1} = \|\mathbf{z}_1 \times \mathbf{c}_1\| = |c_{1y}| = c_{1y}$, $\bar{c}_{1,2} = \|\mathbf{z}_1 \times \mathbf{c}_2\| = |c_{2z}| = c_{2z}$ and $\bar{c}_{2,2} = \|\mathbf{z}_2 \times \mathbf{c}_2\| = 0$.

3.4. Now we substitute the values above into (3.19) to obtain $\bar{b}_1 = \bar{m}_2 g \bar{d}_{1,2} + m_1 g \bar{c}_{1,1} + m_2 g \bar{c}_{1,2} = m_1 g c_{1y} + m_2 g c_{2z}$ and $\bar{b}_2 = \bar{m}_2 g \bar{c}_{2,2} = 0$. Now by substituting in (3.47), we get

$$b = \sqrt{(m_1 g c_{1y} + m_2 g c_{2z})^2} = m_1 g c_{1y} + m_2 g c_{2z}.$$

Since joint 1 is revolute $u_1 = 0$. We calculate u_2 next using (3.43). From (3.31) we see that $\mu_{1,2} = 1$ and $\mu_{2,2} = 0$. Using (3.44) we obtain the following:

$$3.1. \quad \bar{u}_{1,1}^2 = \bar{m}_2 g \mu_1 \mu_{1,2} \bar{q}_2 = \bar{m}_2 g \bar{q}_2.$$

$$3.2. \quad \bar{u}_{1,2}^2 = \bar{m}_2 g \mu_1 \mu_{2,2} = 0.$$

$$3.3. \quad \bar{u}_{2,2}^2 = \bar{m}_2 g \mu_2 \mu_{2,2} = 0.$$

Now by substituting in (3.43), we get

$$u_2 = \sqrt{(\bar{u}_{1,1}^2)^2 + 2(\bar{u}_{1,2}^2)^2 + (\bar{u}_{2,2}^2)^2} = \sqrt{(\bar{m}_2 g \bar{q}_2)^2} = \bar{m}_2 g \bar{q}_2.$$

Finally by substituting in (eqdefbet'), we obtain

$$c_4 = m_1 g c_{1y} + m_2 g c_{2z} + \bar{m}_2 g \bar{q}_2. \quad (3.48)$$

As in the previous example, in order to compare the bound with the exact value of $\left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\|$, we use the expression for $\frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}}$ in (3.40) to generate a plot of $\left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\|$ and the bound c_4 given in (3.48) as q_1 is varied from -180° to 180° and q_2 is varied between 0 and 1.5 *meters* (See Figure 3.7). We chose $m_1 = 5$, $m_2 = 4$, $c_{1y} = 0.25$, $c_{2z} = 0.2$ and $\bar{q}_2 = 1.5$. Note that this bound is only valid locally in the region satisfying $|q_2| \leq \bar{q}_2$.

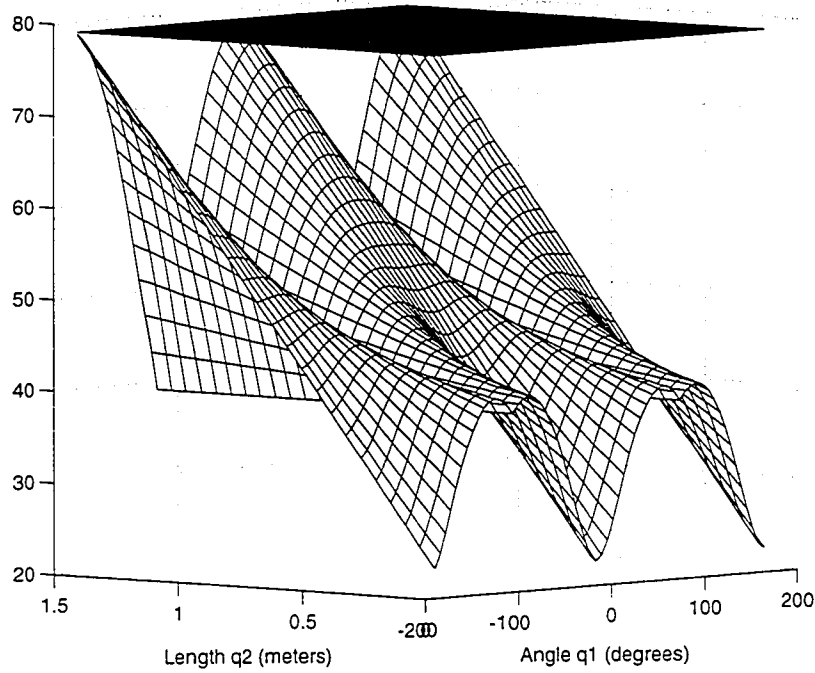


Figure 3.7: Uniform Bound β and $\left\| \frac{\partial \mathbf{g}}{\partial \mathbf{q}} \right\|$ for the \mathcal{RP} Robot

3.5 Summary

Many theoretical results on motion regulation and tracking require a uniform bound for the Hessian. However, not all joint configurations of robots insure the existence of such a uniform bound. Therefore, for the implementation of these control laws it is important to characterize the class of robots for which the Hessian is uniformly bounded (Class \mathcal{BGJ}). The first contribution of this chapter was the full characterization of Class \mathcal{BGJ} . The resulting requirement for belonging to Class \mathcal{BGJ} depended on both the relative orientations of the joints as well as the orientation of the robot in the gravity field. Three examples were presented to illustrate this result.

The selection of parameters of those control laws that require a uniform bound for the Hessian, is based on the value of the uniform bound. Therefore, an explicit expression for the uniform bound will be useful in the implementation of these control laws. The second contribution of this chapter was the derivation of an explicit expression for the uniform bound for Class \mathcal{BGJ} robots. This bound is valid globally and is given in terms of the constant link parameters of the robot. We presented a simple algorithm for computing this bound and illustrated it by computing the uniform bound for the PUMA560 robot.

For robots belonging to Class \mathcal{BGJ} , the results that use the uniform bound for the Hessian will be valid globally. For robots outside Class \mathcal{BGJ} , these results will only be valid locally. Nevertheless, they can still be applied provided the system is kept within the valid domain for the corresponding result. The third contribution of this chapter was an explicit expression of a bound for the Hessian for Class $\overline{\mathcal{BGJ}}$ robots that is valid within any given compact subspace. This bound is given in terms of constant link parameters and upper bounds for prismatic joint variables. This result was also illustrated with an example.

Chapter 4

Implementing PD control with Simple Gravity Compensation in Parallel Robots for Set Point Tracking Applications

In this chapter we implement PD control with simple gravity compensation in a parallel robot based on the dynamics formulation of [24]. We begin by reviewing in Section 4.1 the formulation of the equations of motion and the stability properties of the PD plus simple gravity control law. This is followed by Section 4.2 where we introduce the Rice Planar Delta Robot (R.P.D.R.), which was designed and built at Rice University to perform control experiments, and derive the equations of motion of the R.P.D.R. In Section 4.3, we implement the PD plus simple gravity control law (1.10) on the R.P.D.R. We also set up a simulation of the experiment in Section 4.3 using the equations of motion derived in Section 4.2 and present comparisons of the experimental results with the simulation. Finally in Section 4.4, we summarize the contributions of this chapter.

4.1 PD Control of Closed-Chain Mechanisms

In this section we first review the dynamics model for closed-chain mechanisms originally presented in [24], and second summarize the PD plus simple gravity control result for this class of systems originally developed in [22]. This will be used as the basis to develop the equations of motion for the R.P.D.R. and implement the control result experimentally in later sections.

4.1.1 The Reduced Model for Parallel Robots

The parallel robot referred to here as the constrained system Σ , can be thought of as consisting of a free system Σ' to which constraints \mathcal{C} are applied as shown in Figure 1.3. The free system Σ' is an n' degree-of-freedom (d.o.f) holonomic system which consists of a collection of rigid bodies described by (1.7). The constraints applied to the free system are represented by $(n' - n)$ independent scleronomic holonomic constraints given by

$$\mathcal{C} : \phi(\mathbf{q}') = 0 \quad (4.1)$$

where $\phi(\mathbf{q}')$ is at least twice continuously differentiable, a consequence of which is that $\phi_{q'}(\mathbf{q}') \triangleq \frac{\partial \phi}{\partial \mathbf{q}'}(\mathbf{q}')$ is of full $(n' - n)$ rank. With the introduction of the constraints (4.1), the generalized coordinates \mathbf{q}' are restricted to a subspace of Ω' , namely, $\mathbf{q}' \in \mathbf{U}'$ where

$$\mathbf{U}' \triangleq \{\mathbf{q}' \in \Omega' : \phi(\mathbf{q}') = 0\} \subset \Omega'. \quad (4.2)$$

It follows from standard results in dynamics [33] that the free system Σ' (1.7), with imposed constraints \mathcal{C} (4.1), has n -d.o.f, and hence there exists a minimum set of n -independent generalized coordinates $\mathbf{q} \in \Omega \subset \mathbb{R}^n$ such that the system can be written in terms of \mathbf{q} as follows

$$\Sigma : D(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = 0 \quad (4.3)$$

where $D(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \in \mathbb{R}^n$ is the centrifugal and Coriolis vector, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity vector. This formulation is sometimes referred to as formulation in reduced form [72]. In many situations, such as the case of the R.P.D.R., we may be able to choose the generalized coordinates to coincide with the variables of the actuated joints in which case the equations of motion (4.3) can be written as

$$\Sigma : D(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u} \quad (4.4)$$

where $\mathbf{u} \in \mathbb{R}^n$ is the applied generalized force vector.

Using the given information about the free system Σ' and the constraints \mathcal{C} , a reduced model Σ was developed in [24] and summarized in the following steps:

- The independent generalized coordinates \mathbf{q} with which we would like to describe the constrained system (4.3) can be chosen to satisfy the following twice continuously differentiable parameterization

$$\begin{aligned}\alpha : \mathbf{U}' &\longrightarrow \mathbf{U} \subset \mathbb{R}^n \\ \mathbf{q}' &\longmapsto \mathbf{q} = \alpha(\mathbf{q}').\end{aligned}$$

- Define the following quantities

$$\psi(\mathbf{q}') \triangleq \begin{bmatrix} \phi(\mathbf{q}') \\ \alpha(\mathbf{q}') \end{bmatrix}, \quad (4.5)$$

$$\psi_{q'}(\mathbf{q}') \triangleq \frac{\partial \psi}{\partial \mathbf{q}'}, \quad (4.6)$$

In relation to (4.5), define

$$\bar{\psi}(\mathbf{q}', \mathbf{q}) \triangleq \begin{bmatrix} \phi(\mathbf{q}') \\ \alpha(\mathbf{q}') \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{q} \end{bmatrix}, \quad (4.7)$$

and

$$\bar{\psi}_{q'}(\mathbf{q}') \triangleq \frac{\partial \bar{\psi}}{\partial \mathbf{q}'}. \quad (4.8)$$

Note that $\bar{\psi}_{q'} = \psi_{q'}$.

- We now define the set $\mathbf{V}' \triangleq \{\mathbf{q}' \in \mathbf{U}' : \det[\psi_{q'}(\mathbf{q}')] \neq 0\} \subset \mathbf{U}'$. We say that the system is in a singular configuration when \mathbf{q}' is an element of \mathbf{U}' but is not an element of \mathbf{V}' . It follows that \mathbf{V}' is the workspace region in the \mathbf{q}' coordinates where the constrained system satisfies the constraints and in addition is not in a singular configuration.

- For any given point $\mathbf{q}'_\star \in \mathbf{V}'$, let $\bar{\psi}(\mathbf{q}'_\star, \mathbf{q}_\star) = 0$. Using the Implicit Function Theorem [69], we conclude that there exist a neighborhood $\mathcal{N}_{q'_\star}$ of \mathbf{q}'_\star , and a neighborhood \mathcal{N}_{q_\star} of \mathbf{q}_\star , such that for any $\mathbf{q} \in \mathcal{N}_{q_\star}$, there exists a unique $\mathbf{q}' \in \mathcal{N}_{q'_\star}$, such that $\mathbf{q}' = \sigma(\mathbf{q})$. Furthermore, $\dot{\mathbf{q}}' = \rho(\mathbf{q}') \dot{\mathbf{q}}$ where

$$\rho(\mathbf{q}') = \psi_{q'}^{-1}(\mathbf{q}') \begin{bmatrix} 0_{(n'-n) \times n} \\ I_{n \times n} \end{bmatrix}. \quad (4.9)$$

- We now let $\mathbf{W}' \subset \mathbf{V}'$ denote the *largest* subset of \mathbf{V}' containing \mathbf{q}'_\star for which the unique parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ holds. An explicit characterization of \mathbf{W}' for the general case is reported in [11]. We now denote the corresponding domain of σ by Ω . Hence, we have a diffeomorphism from \mathbf{W}' to Ω as follows: $\mathbf{W}' \xrightarrow{\alpha} \Omega \xrightarrow{\sigma} \mathbf{W}'$.
- Finally, the equations of motion of the constrained system expressed in terms of independent generalized coordinates $\mathbf{q} \in \Omega$, as given in (4.4) and repeated here for convenience,

$$\Sigma: D(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \quad (4.10)$$

are obtained by combining

$$\begin{cases} D(\mathbf{q}')\ddot{\mathbf{q}} + C(\mathbf{q}', \dot{\mathbf{q}}')\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}') = \mathbf{u} \\ \dot{\mathbf{q}}' = \rho(\mathbf{q}') \dot{\mathbf{q}} \\ \mathbf{q}' = \sigma(\mathbf{q}) \end{cases} \quad (4.11)$$

where $\forall \mathbf{q}' \in \mathbf{W}'$

$$D(\mathbf{q}') = \rho(\mathbf{q}')^T D'(\mathbf{q}') \rho(\mathbf{q}') \quad (4.12)$$

$$C(\mathbf{q}', \dot{\mathbf{q}}') = \rho(\mathbf{q}')^T C'(\mathbf{q}', \dot{\mathbf{q}}') \rho(\mathbf{q}') + \rho(\mathbf{q}')^T D'(\mathbf{q}') \dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}') \quad (4.13)$$

$$\mathbf{g}(\mathbf{q}') = \rho(\mathbf{q}')^T \mathbf{g}'(\mathbf{q}') \quad (4.14)$$

$\rho(\mathbf{q}')$ is given by equation (4.9)

$\sigma(\mathbf{q})$ its existence is insured by the Implicit Function Theorem.

The reduced model described above has two special characteristics which make it different from regular models of open-chain mechanical systems. First, the above reduced model is valid only (locally) in a compact set Ω . This means that control strategies can only insure local stability results at best. Second, since the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ is implicit, it is an implicit model. It follows that in order to apply model based control strategies, there is a need to instantaneously compute the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ which is a difficult task in general even though its existence is insured by the Implicit Function Theorem. When implementing model based control laws using digital computers, this means that at each fraction of the control sampling period, a numerical technique such as Newton-Raphson has to successfully compute this parameterization which, even with the current advances in digital hardware, could be very difficult to achieve. This difficulty is avoided with a PD plus simple gravity compensation control as discussed next.

4.1.2 PD Plus Simple Gravity Compensation

The applicability of PD-based control strategies for closed-chain mechanisms was investigated in [22]. It was proposed that PD with full as well as with simple gravity compensation were possible thanks to the skew symmetry property of $\dot{D} - 2C$, a property that was established for closed-chain mechanisms in [22]. In particular, PD plus simple gravity compensation is very attractive because it avoids the online computation of the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$.

Consider the following PD plus simple gravity compensation control law (1.10). In reference to the equations of motion of the closed-chain mechanism, equation (4.11), note that

$$\mathbf{g}(\mathbf{q}^d) = \rho^T \left(\sigma(\mathbf{q}^d) \right) \mathbf{g}' \left(\sigma(\mathbf{q}^d) \right).$$

Since the parameterization $\mathbf{q}' = \sigma(\mathbf{q}^d)$ is now a constant value, it could be computed *off-line* to any degree of accuracy using any of the many well established numerical techniques. It follows that the constant term $\mathbf{g}(\mathbf{q}^d)$ in (1.10) could also be accurately computed off-line.

Suppose now that for a particular closed-chain system, there is a simply connected set $\mathcal{D} \subset \Omega$ and consider a set-point (regulation) problem in which it is required that the vector \mathbf{q} reach a constant desired value $\bar{\mathbf{q}}_d \in \mathcal{D}$. Define $\tilde{\mathbf{q}} \triangleq \bar{\mathbf{q}}_d - \mathbf{q}$, and $\mathcal{B} \triangleq \{(\tilde{\mathbf{q}}, \dot{\mathbf{q}}) : \mathbf{q}, \bar{\mathbf{q}}_d \in \mathcal{D}\}$. The PD plus simple gravity compensation control result is summarized as follows:

Theorem 4.1 [22] Define the set $\mathcal{B}_2 \triangleq \{(\tilde{\mathbf{q}}, \dot{\mathbf{q}}) : V_2(\tilde{\mathbf{q}}, \dot{\mathbf{q}}) \leq c_2\}$ where c_2 is the largest positive real number such that $\mathcal{B}_2 \subset \mathcal{B}$. Let the initial conditions $(\tilde{\mathbf{q}}_0, \dot{\mathbf{q}}_0) \in \mathcal{B}_2$, and let $\left\| \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq \beta$ in \mathcal{B}_2 where β is a positive constant. Choose $k_{pi} > \beta$, $i = 1, 2, \dots, n$, where k_{pi} are the diagonal elements of K_p . Then the equilibrium $\tilde{\mathbf{q}} = 0$ and $\dot{\mathbf{q}} = 0$ of the closed-chain mechanism model (4.11) with the control law (1.10) is (locally) asymptotically stable and $\mathbf{q} \longrightarrow \bar{\mathbf{q}}_d$, $\dot{\mathbf{q}} \longrightarrow 0$, as $t \longrightarrow \infty$.

Note that the existence of the constant β is insured by the continuity of the gravity vector \mathbf{g} since the set \mathcal{B}_2 is compact.

4.2 The Rice Planar Delta Robot (R.P.D.R.)

In order to experimentally verify the results on control of parallel robots developed in this project there was a need to have a programmable parallel robot. We designed and built the Rice Planar Delta Robot (R.P.D.R.) for this purpose. We chose to construct a planar delta robot since it's one of the simplest parallel robots and it can be modeled easily. The R.P.D.R. has four links connected through revolute joints (see Figure 4.1). Two of the joints (Joint 1 and Joint 2) are actuated while the other

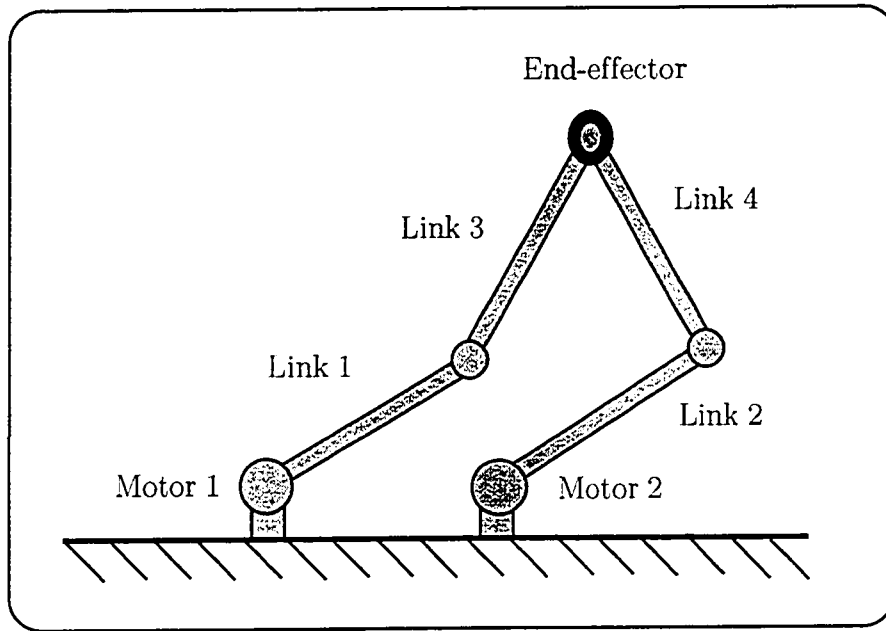


Figure 4.1: Link and Joint Configuration of the R.P.D.R.

three joints are passive. It can be seen that the R.P.D.R. has two degrees of freedom. Thus, the number of inputs are selected to match the number of degrees of freedom.

In the R.P.D.R. joints 1 and 2 are actuated by permanent magnet DC motors. In order to provide sensory feedback for control, we used optical encoders having a resolution of 10,000 pulses per revolution. In order to freeze the robot once it reaches a desired configuration the R.P.D.R. is equipped with electromagnetic brakes. Since the DC motors provide high speed and low torque, in order to drive the robot there was a need to include a transmission system that reduce speed (and increase torque). Typical transmission systems have undesirable properties such as backlash and joint flexibility. For the R.P.D.R. we chose a special timing chain and sprocket system that dramatically reduce these effects. For joints 1 and 2 the actuator/sensor assembly consists of the motor, the encoder, the electromagnetic brake, and the transmission system (see Figure 4.2) and they are identical. The passive joints were designed to

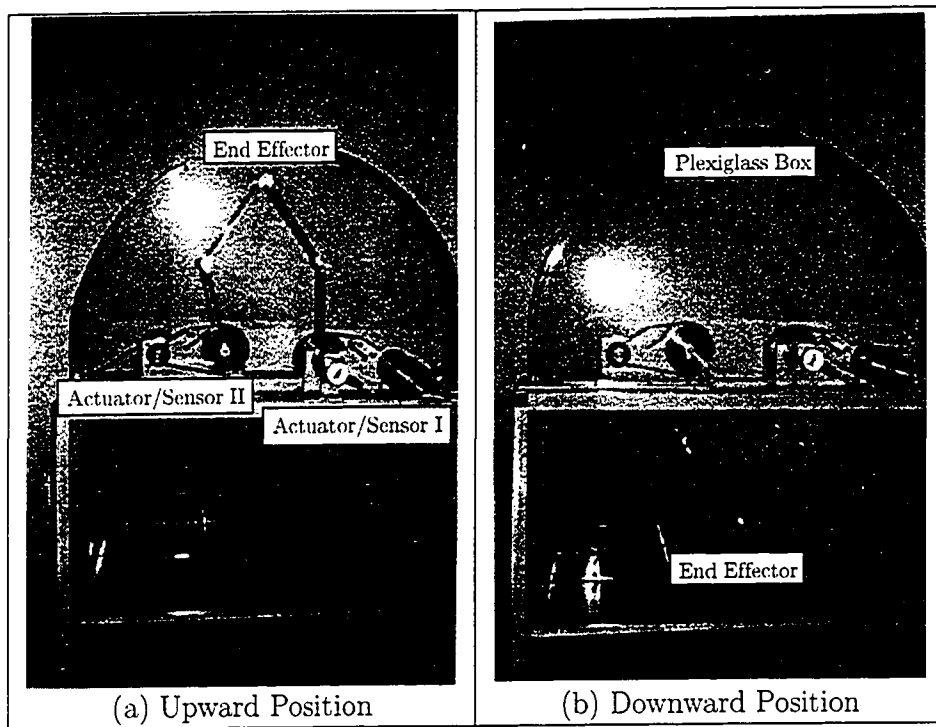


Figure 4.2: Rice Planar Delta Robot (R.P.D.R.)

minimize friction using roller bearings. They were made with Aluminum alloys to reduce weight. The links were made of 0.5 inch square hollow steel tubing which provide high rigidity and relatively light weight. The link train starts from Joint 1 and ends in Joint 2 (see Figure 4.2). The base of the robot is made of steel and constructed to be rigid. The actuator/sensor assemblies corresponding to joints 1 and 2 are mounted on the base. For safety reasons the moving parts of the robot is enclosed by a plexi-glass frame.

The primary objective of the R.P.D.R. is to serve as a test bed to perform experiments on control. For this reason it was designed such that the dynamic parameters can be adjusted. The two actuator/sensor assemblies corresponding to joints 1 and 2 are designed such that they can be mounted to the base of the robot with varying dis-

tance between them. If there is a need to vary the link lengths (and weight), this can also be achieved in the R.P.D.R. by using a new set of tubing (which is inexpensive). In order to study singularities the RPDR was designed such that all of the joints can rotate through a full circle (360 degrees). This allows the RPDR to be driven to a singular configuration if needed for some experiment.

The R.P.D.R. is controlled with an IBM PC to which it is interfaced through a DSP board made by dSPACE [16]. The DSP board controls the motors by sending a pulse width modulated signal in which the pulse width is proportional to the desired torque. This signal is amplified by an H-bridge circuit and fed to the motors. The DSP board has two encoder channels which is used to interface the optical encoders. The electromagnetic brakes are controlled by two of the digital I/O lines of the DSP board. In order to implement most control laws there is a need for velocity feedback in addition to position feedback. In the R.P.D.R. velocity feedback is provided by differentiating the position and filtering with a low pass filter implemented in the DSP. In addition to providing the interface between the robot and the computer the dSPACE board and software performs the important function of real time data acquisition. This enables us to monitor the sensory data as well as any other variable associated with the experiment during the implementation.

In the implementation of control the dSPACE board computes the control law in real time based on the position/velocity feedback and outputs the desired torque to the motors. This is done in an interrupt routine which is activated at a fixed sampling rate. The sampling rate can be programmed and we selected 5 KHz. (a step size of 200 microseconds) for the experiments reported in this chapter and a sampling rate of 1 KHz for the experiments reported in Chapter 5. In each experiment in addition to the interrupt routine implemented in the DSP, another program implemented in windows communicates with the interrupt routine. This program allows the operator

to control the robot during experiments. For a full description of the design process of the R.P.D.R. refer to Appendix A. All of these programs were written in C. Please refer to [34] for a full listing of the programs corresponding to each experiment.

4.2.1 Equations of Motion of the R.P.D.R.

In this section, we will derive the equations of motion for the R.P.D.R. using the formulation summarized in Section 4.1.1. The first step in deriving the equations of motion is selecting the “free system”. In our free system, the robot is virtually cut open at the end-effector, resulting in two serial robots each having two degrees of freedom (see Figure 4.3). As defined in the figure, m_i , c_i , and a_i are respectively the mass, distance to the center of mass, and length of link i . The inertia of link i about the line through the center of mass parallel to the axis of rotation is denoted by I_i . The parameters corresponding to Link 1 and Link 3 are defined similar to the parameters of Link 2 and Link 4 even though they are not shown in the figure. Thus

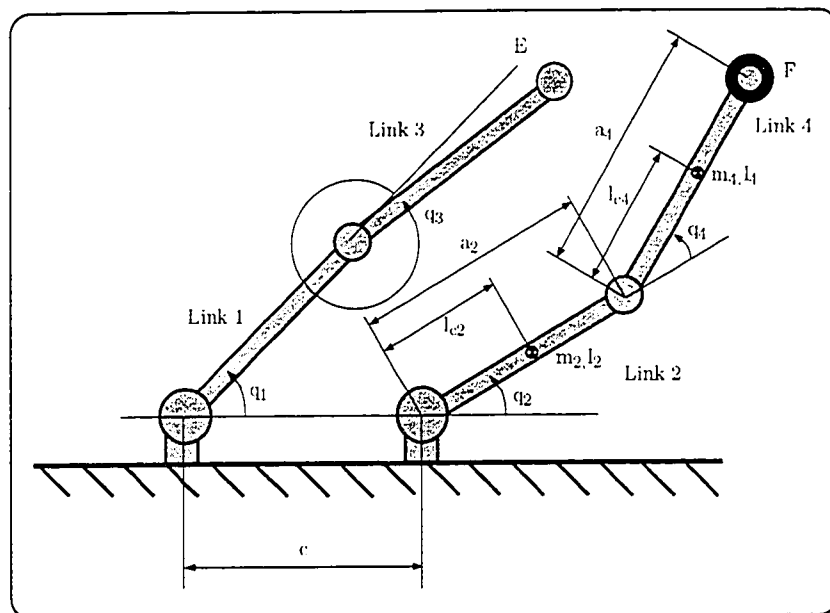


Figure 4.3: The “free system”

the constraint equations are due to point E being coincident with point F and are given by,

$$\phi(\mathbf{q}') = \begin{bmatrix} a_1 \cos(q_1) + a_3 \cos(q_1 + q_3) - c - a_2 \cos(q_2) - a_4 \cos(q_2 + q_4) \\ a_1 \sin(q_1) + a_3 \sin(q_1 + q_3) - a_2 \sin(q_2) - a_4 \sin(q_2 + q_4) \end{bmatrix} = \mathbf{0}, \quad (4.15)$$

where $\mathbf{q}' = [q_1 \ q_2 \ q_3 \ q_4]^T$ is the generalized coordinate vector of the free system. Note that since the actuated joints are joints 1 and 2 for the R.P.D.R., we choose the generalized coordinate vector of the constrained system to be $\mathbf{q} = [q_1 \ q_2]^T$. Our next objective will be to derive expressions for each of the terms appearing in (4.11) for the R.P.D.R. to obtain the equations of motion. We begin with the parameterization $\alpha(\mathbf{q}') = \mathbf{q}$, which is given by,

$$\alpha(\mathbf{q}') = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{q}' = \mathbf{q}. \quad (4.16)$$

Now by combining (4.15) and (4.16) and differentiating with respect to \mathbf{q}' , we obtain the following expression for $\psi_{\mathbf{q}'}(\mathbf{q}')$:

$$\psi_{\mathbf{q}'}(\mathbf{q}') = \begin{bmatrix} \psi_{q'}(1, 1) & \psi_{q'}(1, 2) & -a_3 \sin(q_1 + q_3) & a_4 \sin(q_2 + q_4) \\ \psi_{q'}(2, 1) & \psi_{q'}(2, 2) & a_3 \cos(q_1 + q_3) & -a_4 \cos(q_2 + q_4) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (4.17)$$

where, $\psi_{q'}(1, 1) = -a_1 \sin(q_1) - a_3 \sin(q_1 + q_3)$, $\psi_{q'}(1, 2) = a_2 \sin(q_2) + a_4 \sin(q_2 + q_4)$, $\psi_{q'}(2, 1) = a_1 \cos(q_1) + a_3 \cos(q_1 + q_3)$, and $\psi_{q'}(2, 2) = -a_2 \cos(q_2) - a_4 \cos(q_2 + q_4)$. Now from (4.9) we have the following expression for $\rho(\mathbf{q}')$:

$$\rho(\mathbf{q}') = \psi_{\mathbf{q}'}^{-1}(\mathbf{q}') \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4.18)$$

Since $\rho(\mathbf{q}')$ is in terms of an inverse matrix, it is not easy to take the time derivative. Therefore, we use the following expression for $\dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}')$, which can be easily obtained

by pre-multiplying (4.18) with $\psi_{\mathbf{q}'}(\mathbf{q}')$ and taking the time derivative:

$$\dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}') = -\psi_{\mathbf{q}'}^{-1}(\mathbf{q}') \dot{\psi}_{\mathbf{q}'}(\mathbf{q}', \dot{\mathbf{q}}') \rho(\mathbf{q}'), \quad (4.19)$$

where $\dot{\psi}_{\mathbf{q}'}(\mathbf{q}', \dot{\mathbf{q}}')$ can be obtained by differentiating (4.17) with respect to time. Next we will derive expressions for $D'(\mathbf{q}')$, $C'(\mathbf{q}', \dot{\mathbf{q}}')$, and $\mathbf{g}'(\mathbf{q}')$ that appear in (4.12), (4.13), and (4.14). The following expressions were derived using the Lagrangian method:

$$D'(\mathbf{q}') = \begin{bmatrix} d_{1,1} & 0 & d_{1,3} & 0 \\ 0 & d_{2,2} & 0 & d_{2,4} \\ d_{3,1} & 0 & d_{3,3} & 0 \\ 0 & d_{4,2} & 0 & d_{4,4} \end{bmatrix}, \quad (4.20)$$

where, $d_{1,1} = m_1 l_{c1}^2 + m_3(a_1^2 + l_{c3}^2 + 2a_1 l_{c3} \cos(q_3)) + I_1 + I_3$, $d_{1,3} = m_3(l_{c3}^2 + a_1 l_{c3} \cos(q_3)) + I_3$, $d_{2,2} = m_2 l_{c2}^2 + m_4(a_2^2 + l_{c4}^2 + 2a_2 l_{c4} \cos(q_4)) + I_2 + I_4$, $d_{2,4} = m_4(l_{c4}^2 + a_2 l_{c4} \cos(q_4)) + I_4$, $d_{3,1} = d_{1,3}$, $d_{3,3} = m_3 l_{c3}^2 + I_3$, $d_{4,2} = d_{2,4}$, $d_{4,4} = m_4 l_{c4}^2 + I_4$,

$$C'(\mathbf{q}', \dot{\mathbf{q}}') = \begin{bmatrix} h_1 \dot{q}_3 & 0 & h_1(\dot{q}_1 + \dot{q}_3) & 0 \\ 0 & h_2 \dot{q}_4 & 0 & h_2(\dot{q}_2 + \dot{q}_4) \\ -h_1 \dot{q}_1 & 0 & 0 & 0 \\ 0 & -h_2 \dot{q}_2 & 0 & 0 \end{bmatrix}, \quad (4.21)$$

where $h_1 = -m_3 a_1 l_{c3} \sin(q_3)$, and $h_2 = -m_4 a_2 l_{c4} \sin(q_4)$, and

$$\mathbf{g}'(\mathbf{q}') = \begin{bmatrix} (m_1 l_{c1} + m_3 a_1) \cos(q_1) + m_3 l_{c3} \cos(q_1 + q_3) \\ (m_2 l_{c2} + m_4 a_2) \cos(q_2) + m_4 l_{c4} \cos(q_2 + q_4) \\ m_3 l_{c3} \cos(q_1 + q_3) \\ m_4 l_{c4} \cos(q_2 + q_4) \end{bmatrix} g, \quad (4.22)$$

where $g = 9.81 \text{ m/sec}^2$ is the gravitational acceleration constant. At this point we have derived the equations of motion of the R.P.D.R. in terms of \mathbf{q}' . The only remaining issue is the derivation of the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$. In general, it is not possible to derive an analytic expression for $\sigma(\mathbf{q})$, and it must be computed using numerical methods. For the R.P.D.R. however, it is possible to solve (4.15) to obtain

the following:

$$q_4 = \tan^{-1} \left[\frac{B(q_1, q_2)}{A(q_1, q_2)} \right] + \tan^{-1} \left[\frac{\pm \sqrt{A(q_1, q_2)^2 + B(q_1, q_2)^2 - C(q_1, q_2)^2}}{C(q_1, q_2)} \right] - q_2, \quad (4.23)$$

where $A(q_1, q_2) = 2a_4\lambda(q_1, q_2)$, $B(q_1, q_2) = 2a_4\mu(q_1, q_2)$, and $C(q_1, q_2) = a_3^2 - a_4^2 - \lambda(q_1, q_2)^2 - \mu(q_1, q_2)^2$, and $\lambda(q_1, q_2) = a_2\cos(q_2) - a_1\cos(q_1) + c$, $\mu(q_1, q_2) = a_2\sin(q_2) - a_1\sin(q_1)$. Finally,

$$q_3 = \tan^{-1} \left[\frac{\mu(q_1, q_2) + a_4\sin(q_2 + q_4)}{\lambda(q_1, q_2) + a_4\cos(q_2 + q_4)} \right] - q_1. \quad (4.24)$$

Hence, (4.23) and (4.24) combined represent the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$. This completes the derivation of the equations of motion of the R.P.D.R. In summary, the equations of motion of the R.P.D.R. are given by

$$\begin{cases} D(\mathbf{q}')\ddot{\mathbf{q}} + C(\dot{\mathbf{q}}', \mathbf{q}')\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}') = \mathbf{u}, \\ \dot{\mathbf{q}}' = \rho(\mathbf{q}')\dot{\mathbf{q}}, \\ \mathbf{q}' = \sigma(\mathbf{q}), \end{cases}$$

where

$$\begin{aligned} D(\mathbf{q}') &= \rho(\mathbf{q}')^T D'(\mathbf{q}') \rho(\mathbf{q}'), \\ C(\mathbf{q}', \dot{\mathbf{q}}') &= \rho(\mathbf{q}')^T C'(\mathbf{q}', \dot{\mathbf{q}}') \rho(\mathbf{q}') + \rho(\mathbf{q}')^T D'(\mathbf{q}') \dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}'), \\ \mathbf{g}(\mathbf{q}') &= \rho(\mathbf{q}')^T \mathbf{g}'(\mathbf{q}'), \end{aligned}$$

$D'(\mathbf{q}')$, $C'(\mathbf{q}', \dot{\mathbf{q}}')$, $\mathbf{g}'(\mathbf{q}')$, $\dot{\rho}(\mathbf{q}', \dot{\mathbf{q}}')$, $\rho(\mathbf{q}')$, and $\sigma(\mathbf{q})$ are defined in (4.20), (4.21) (4.22), (4.19), (4.18), and (4.23) plus (4.24) respectively.

4.3 Implementation of the PD Control Law

Our next objective is to implement the PD plus simple gravity compensation control discussed in Section 4.1.2 using the computer controlled R.P.D.R.. First, we need to characterize a compact domain, a subset of Ω , where the parameterization $\sigma(\mathbf{q})$

exists. For the R.P.D.R., if we select a simply connected region $\mathcal{D} \subset \Omega$ that is free of singularities, the parameterization $\sigma(\mathbf{q})$ will exist in that region. Therefore, our first objective was to characterize the singular points of the R.P.D.R. From (4.17), we see that $\det[\psi_{\mathbf{q}'}(\mathbf{q}')] = \sin(q_1 + q_3 - q_2 - q_4)$. Consequently, the singular points of the R.P.D.R. are those for which $(q_1 + q_3 - q_2 - q_4) = n\pi$, for $n = 0, \pm 1, \pm 2, \dots$. By substituting these into the constraint equations (4.15), we were able to solve for all of the singular points of the R.P.D.R. These are plotted in Figure 4.4 which shows the entire workspace of the R.P.D.R. (corresponding to the range -180° to 180° for Joint 1 and Joint 2). We considered two cases corresponding to n being even and n being odd. The two cases led to two types of singularities. There are infinitely many

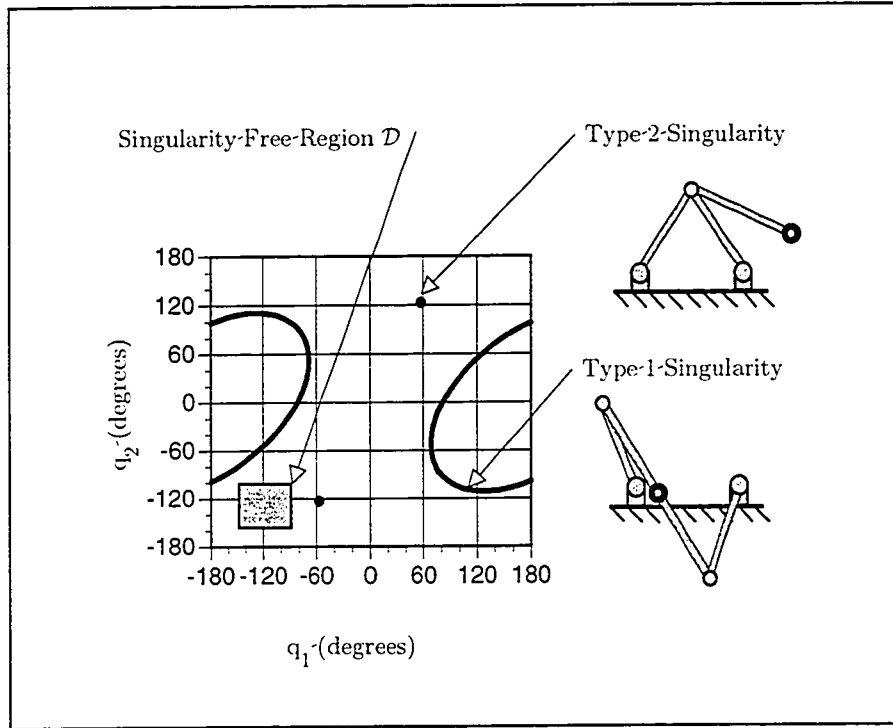


Figure 4.4: Singular Configurations of the R.P.D.R. and the Region \mathcal{D}

singularities of the first type which occur when n is odd. The second type occurs when

n is even and the robot loses control of Link 3 and Link 4 in these configurations. There are two singularities of this type for the R.P.D.R. as shown in Figure 4.4. For our experiments we chose the singularity free region \mathcal{D} where $-150^\circ \leq q_1 \leq -90^\circ$ and $-160^\circ \leq q_2 \leq -100^\circ$.

We next present some of the initial experiments that were performed to identify the system parameters and in particular frictional effects. We then present simulation as well as experimental results on a simple point to point motion. Finally we present experimental and simulation results on a repetitive point to point motion. This type of motion is typical in pick and place applications of parallel robots.

4.3.1 Evaluation of the Parameters in the Equations of Motion

Before implementing control, initial experiments were performed to identify the parameters in the equations of motion derived in Section 4.2.1. The link parameters were calculated and verified by measurement. These are presented in Table 4.1. In

Link i	m_i (kg)	a_i (m)	l_{ci} (m)	I_i (kg m ²)
1	0.2451	0.2794	0.1466	2.779×10^{-3}
2	0.2352	0.2794	0.141	2.607×10^{-3}
3	0.2611	0.3048	0.1581	3.476×10^{-3}
4	0.2611	0.3048	0.1467	3.476×10^{-3}

Table 4.1: Kinematic Link Parameters of the Rice Planar Delta Robot

addition to the inertia represented in $D(\mathbf{q})$, there is a constant inertia due to the motors and the transmission on Joint 1 and Joint 2. These constant inertias were estimated to be 3.3263×10^{-3} kg m². The distance between the axes of Joint 1 and Joint 2 was measured to be $c = 0.3048$ m.

Our next objective was to identify the friction in the system. We considered Coulomb friction, viscous friction and static friction at the initiation of motion. Using

the model of the R.P.D.R. derived in Section 4.1.1 together with the parameters which we calculated, we were able to setup a simulation. Friction was also included in the model with an initial guess of the friction parameters. The real values of the friction parameters were identified by comparing the actual step response of the system with the predicted response of the simulation for different step inputs. These experiments showed us that the frictional effects cannot be neglected in the R.P.D.R.

4.3.2 Experimental Results and Simulations: Point to Point Motion

For our experiment, we chose the point to point trajectory shown in Figure 4.5 which is within the singularity free region \mathcal{D} . The initial configuration is given by $q_1 = -150^\circ$ and $q_2 = -160^\circ$ while the final configuration is given by $q_1 = -90^\circ$ and $q_2 = -100^\circ$. Our next objective was to compute the parameters of the PD with simple gravity

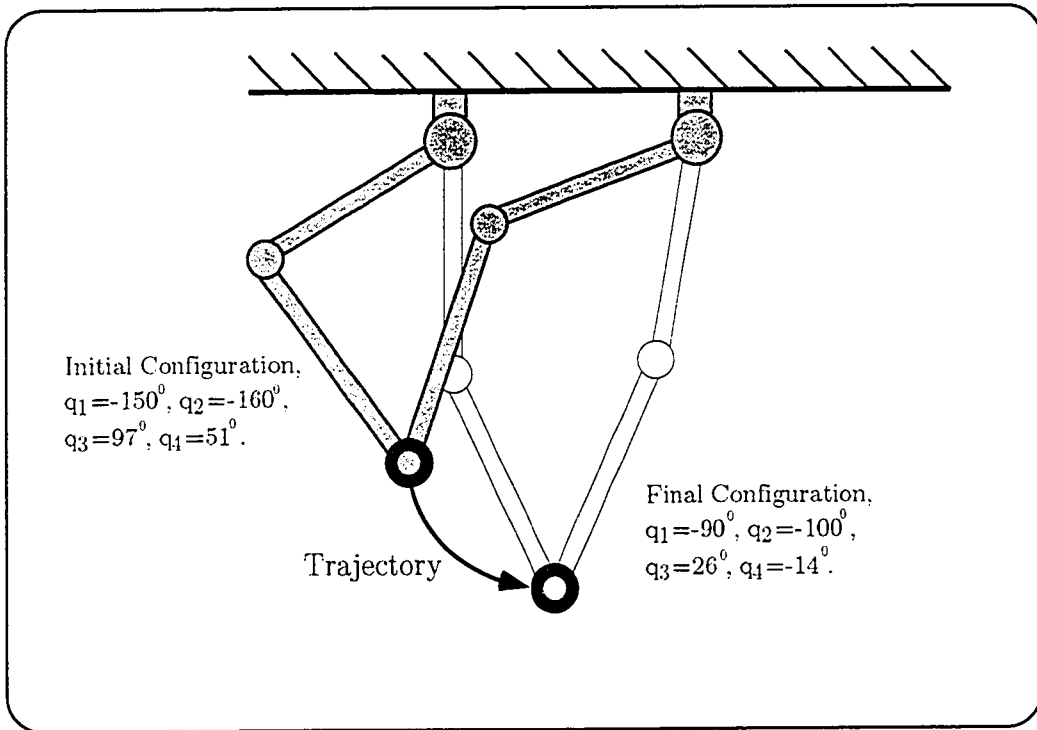


Figure 4.5: Point-To-Point Trajectory

compensation control law (1.10), namely,

$$\mathbf{u} = \mathbf{g}(\mathbf{q}^d) + K_P(\mathbf{q}^d - \mathbf{q}) - K_V\dot{\mathbf{q}}.$$

Based on the system parameters identified in Section 4.3.1, we computed the gravity vector at the desired configuration $\mathbf{g}(\mathbf{q}^d)$ off-line. In order to account for the frictional effects, we estimated the coulomb friction at the desired configuration and added that to $\mathbf{g}(\mathbf{q}^d)$. Since the effect of viscous friction is to increase K_V , there is no need to account for it. We selected K_P to be diagonal with each element equal to 11 N-m/rad which was sufficient to satisfy the requirements of Theorem 4.1 for this experiment. The matrix K_V was also selected diagonal with the first element 0.65 and the second element 0.6 N-m-s/rad. The values of the elements of K_V were selected by trial and error to make the system close to critically damped.

In order to compare the experimental results with the theoretical predictions, we performed a simulation. During the simulation we integrated the equations of motion derived in Section 4.2.1 combined with the control law (1.10), to generate the solution trajectories. We used the Runge-Kutta algorithm in Matlab [54] to perform the integration. For the simulation, we had the same initial configuration and final desired configuration that was used for the experiments. We also included friction (identified in Section 4.3.1) in our model. The simulation and experimental results are presented in Figure 4.6. The dashed curves correspond to the simulation results while the solid curves correspond to the experimental results. It can be seen that the experimental and simulation results match well. The slight deviations can be attributed to inexact modeling of friction and other unmodeled dynamics in the system. Furthermore, it can be seen that the robot achieved the desired goal within approximately 0.3 seconds. It can be seen that the experimental results confirmed the theoretical predictions. A full listing of the C programs corresponding to this experiment can be found in [34].

4.3.3 Experimental Results and Simulations: Repetitive Point to Point Motion

In some industrial applications, parallel robots are used to perform rapid pick and place routines. In a pick and place routine, a robot moves to one point, picks up an object, carries it to another point, and places the object at that point. In this section we present experimental and simulation results of the R.P.D.R. performing rapid pick and place type motions.

The R.P.D.R. was programmed to move repetitively between the initial configuration of Figure 4.5 (say \mathbf{q}^i) and the final configuration (say \mathbf{q}^f). The algorithm performs the following routine:

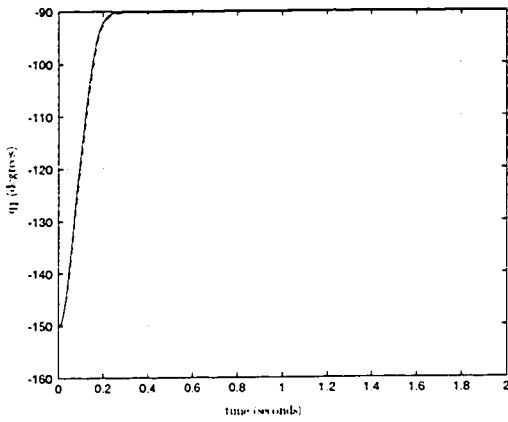
- 4.1. Implement control law (1.10) with $\mathbf{q}^d = \mathbf{q}^f$.
- 4.2. After steady state is achieved wait 2.5 seconds with the control law unchanged.
- 4.3. Implement control law (1.10) with $\mathbf{q}^d = \mathbf{q}^i$.
- 4.4. After steady state is achieved wait 2.5 seconds with the control law unchanged.
- 4.5. Go to Step 4.1.

As in the point to point motion experiment of the previous section, we compensated for the coulomb friction at the desired configuration. The matrices K_P and K_V were also selected to be the same as in the previous experiment. A simulation was also performed using the model developed in previous sections. The experimental results and the simulation results are presented in Figure 4.7. As in the previous case, the dashed curves correspond to the simulation results while the solid curves correspond to the experimental results. The results show that the robot completed the motion in under 0.3 seconds and achieved the desired configuration within reasonable tolerances. The simulation and the experimental results seems to agree quite well.

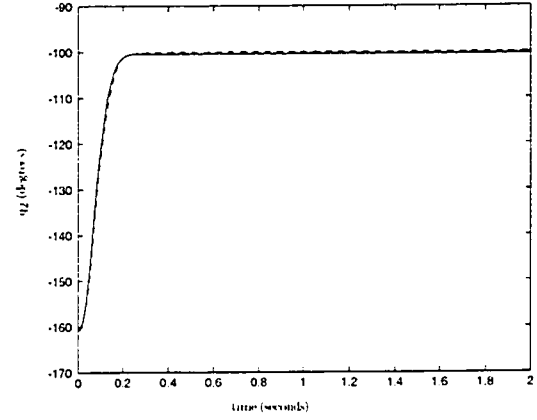
These results show us that the control law (1.10) can be used successfully for pick and place routines provided the coulomb friction is exactly compensated. If a system is developed for a specific routine (where the programming is done for one particular pick and place routine), even if the friction is not known precisely the compensation term can be modified until the error reaches acceptable levels. However, if the system is to be developed for any general pick and place routine, the coulomb friction must be predictable and known exactly in order to successfully implement the control law (1.10). A full listing of the C programs corresponding to this experiment can be found in [34].

4.4 Summary

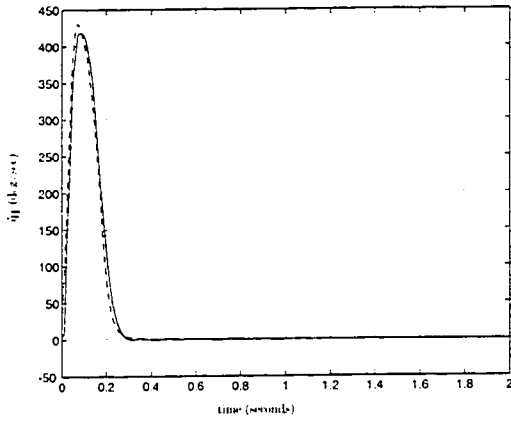
In this chapter, we first reviewed a new formulation of the equations of motion of closed chain mechanisms of [24] and the local asymptotic stability of the PD plus simple gravity compensation control law that was established in [22]. We then experimentally validated this result using the Rice Planar Delta Robot (R.P.D.R.) and evaluated the applicability of this control law for pick-and-place applications of parallel robots. A detailed derivation of the equations of motion was also presented in this chapter which enabled setting up a simulation of the motion. Both simulation and experimental results were presented and showed good agreement.



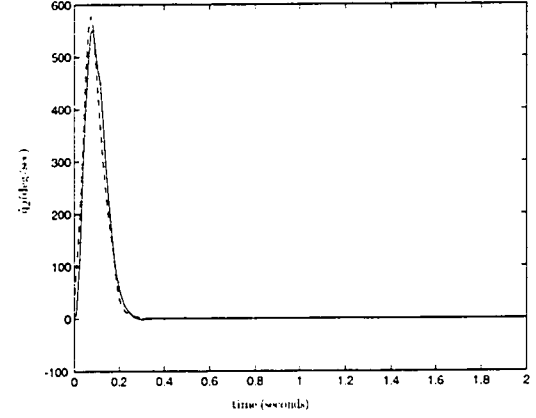
(a) Trajectory Profile of Joint 1



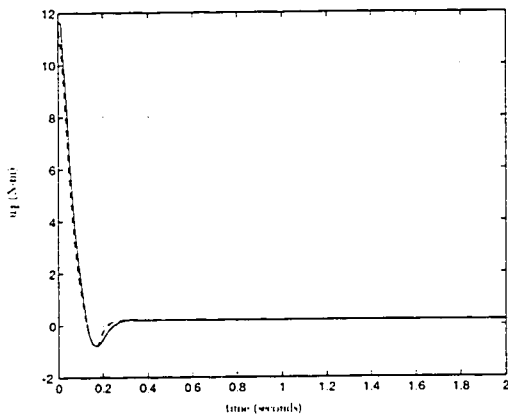
(b) Trajectory Profile of Joint 2



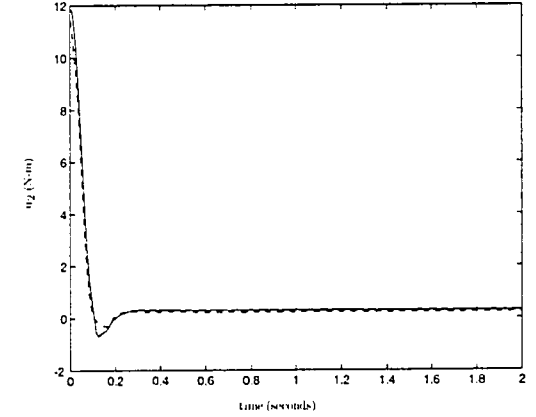
(c) Velocity Profile of Joint 1



(d) Velocity Profile of Joint 2

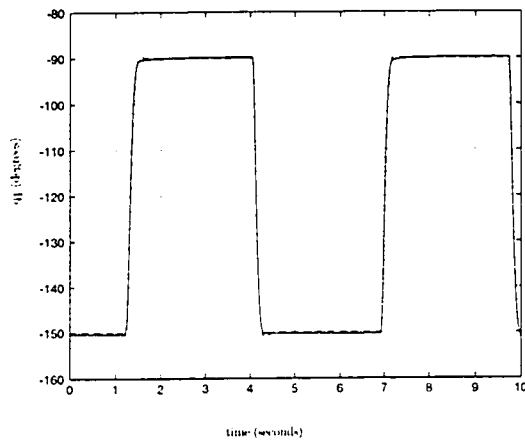


(e) Torque Profile of Joint 1

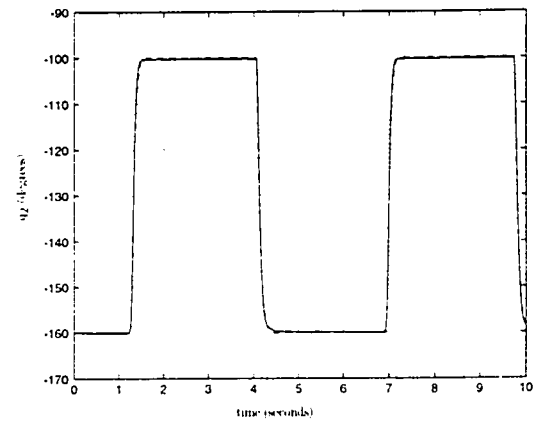


(f) Torque Profile of Joint 2

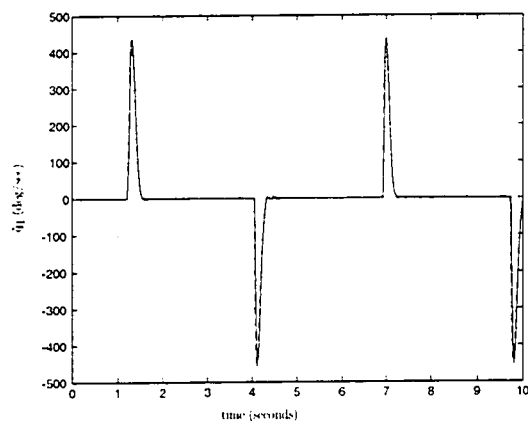
Figure 4.6: Simulation (Dashed) and Experimental (Solid) Results



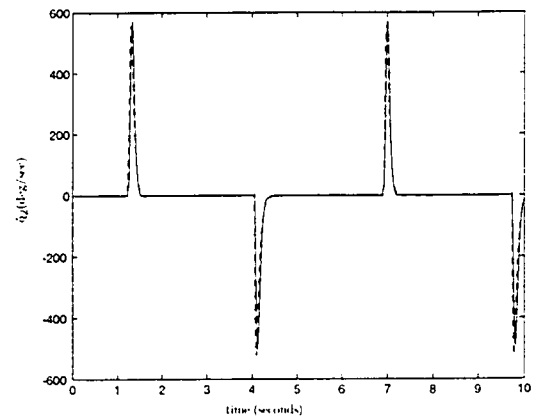
(a) Trajectory Profile of Joint 1



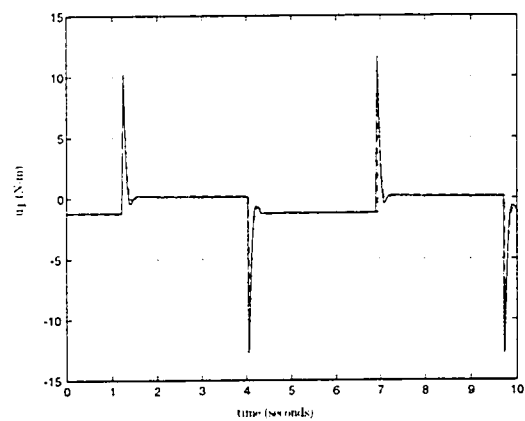
(b) Trajectory Profile of Joint 2



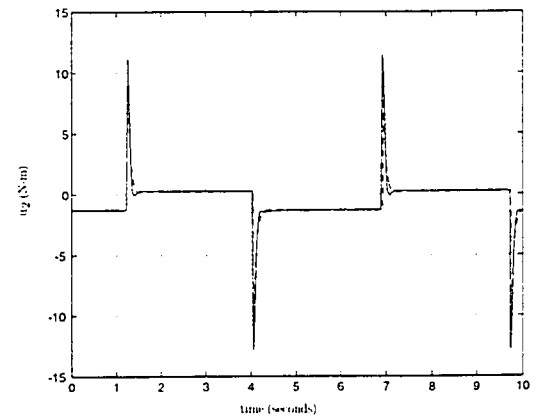
(c) Velocity Profile of Joint 1



(d) Velocity Profile of Joint 2



(e) Torque Profile of Joint 1



(f) Torque Profile of Joint 2

Figure 4.7: Pick and Place Routine: Simulation (Dashed) and Experimental (Solid) Results

Chapter 5

Model Based Control of Parallel Robots for Trajectory Tracking Applications

In this chapter we address the two issues related to using the dynamics formulation of [24] to implement control, namely, the issue of characterizing the domain of the parameterization (1.9) and the issue of computing $\mathbf{q}' = \sigma(\mathbf{q})$ in real time for model based control. This chapter is organized as follows: In Section 5.1 and Section 5.2 we address the issue of characterizing the domain of (1.9). This is followed by Section 5.3 and Section 5.4 where we address the issue of computing $\mathbf{q}' = \sigma(\mathbf{q})$ in real time. We propose a procedure to implement control that ensures convergence in Section 5.5. In Section 5.7 we illustrate this procedure with an example, namely, the Rice Planar Delta Robot (R.P.D.R.) and compute all of the parameters associated with this procedure. Experimental results on the implementation of the procedure are presented in Section 5.8. In this experiment the R.P.D.R. is made to follow a trajectory which makes the end-effector travel in a circle. Finally, in Section 5.9 we summarize the results of this chapter.

5.1 The Domain of the Parameterization $\mathbf{q}' = \sigma(\mathbf{q})$

Given any point $\mathbf{q}'_* \in \mathbf{V}'$, let $\mathbf{q}_* = \alpha(\mathbf{q}'_*)$. It was proved in [24] that there exists neighborhoods $\mathcal{N}_{\mathbf{q}'_*}$ and $\mathcal{N}_{\mathbf{q}_*}$ of \mathbf{q}'_* and \mathbf{q}_* respectively, such that for each $\mathbf{q} \in \mathcal{N}_{\mathbf{q}_*}$ there exists a unique $\mathbf{q}' \in \mathcal{N}_{\mathbf{q}'_*}$ satisfying (1.9). The equations of motion (1.1) were proved to be valid in any subset of \mathbf{V}' for which the parameterization (1.9) is valid. Obviously, the establishment of the existence of a neighborhood where (1.9) is valid

is not sufficient by itself to use this result for control. Our objective will be to characterize a region of the robot workspace, that is sufficiently large to enable control, where the parameterization (1.9) is valid and propose an algorithm to compute it.

Is it possible to prove that the parameterization (1.9) is valid globally? We can show by way of example that this is impossible. Consider the planar delta robot of Figure 1.1. In this robot joints one and two are actuated and hence, $\mathbf{q} = [q_1 \ q_2]^T$. Let $\mathbf{q}' = [q_1 \ q_2 \ q_3 \ q_4]^T$. Clearly, if the parameterization (1.9) is valid globally, for each value of \mathbf{q} there should be a unique \mathbf{q}' satisfying the constraint $\bar{\psi}(\mathbf{q}', \mathbf{q}) = \mathbf{0}$. In the planar delta robot, for most values of \mathbf{q}' , there are two different values of \mathbf{q}' satisfying the constraints $\bar{\psi}(\mathbf{q}', \mathbf{q}) = \mathbf{0}$ (See Figure 5.1). For most parallel robots

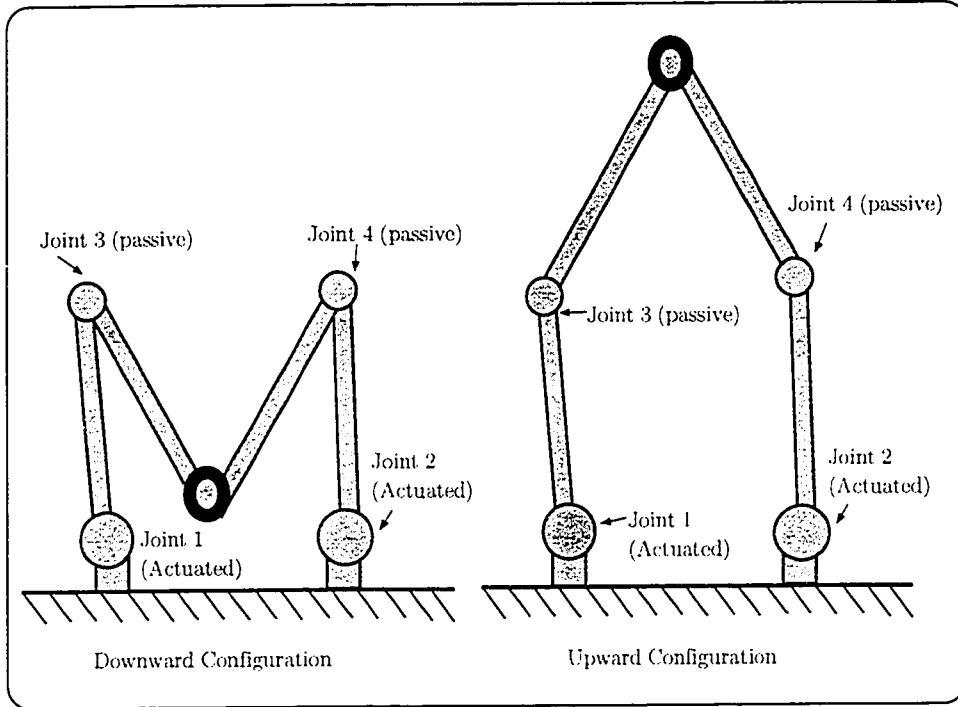


Figure 5.1: Two different configurations of the planar delta corresponding to the same positions of the actuated joints (joints 1 and 2)

the actuated joints do not uniquely determine the entire configuration of the robot. Hence the parameterization (1.9) is not valid globally in general. Our objective will be to characterize a closed set where (1.9) is valid.

One approach that can be taken to solve this problem is to regard it as solving for the zero of a non linear function. That is, for a given point $\mathbf{q} = \mathbf{q}_1$ of the robot workspace, the corresponding value of the dependent coordinates $\mathbf{q}' = \mathbf{q}'_1$ can be found by solving (4.7),

$$\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) = \begin{bmatrix} \phi(\mathbf{q}'_1) \\ \alpha(\mathbf{q}'_1) \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{q}_1 \end{bmatrix} = \mathbf{0}, \quad (5.1)$$

for \mathbf{q}'_1 . For a given value of \mathbf{q}_1 (5.1) is a function of the form $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{f} : \mathbb{R}^{n'} \Rightarrow \mathbb{R}^{n'}$. The issue of solving this problem received much attention in the past and several algorithms have been proposed. Among these, the Newton algorithm is well known. Among the results in this area, the Kantorovich theorem characterizes a set where the Newton algorithm is guaranteed to converge and provides an estimate for the error after m iterations. Therefore, the Kantorovich theorem can be used to characterize a region where a solution to (5.1) exists. This is the approach taken in [10].

An important feature of the problem at hand that is not taken into account in this approach is the variation in \mathbf{q} . A question that is not answered directly in this approach is, how does the variation in \mathbf{q} affect the existence of a solution to (5.1)? This aspect is important in our application since the parallel robot will always start from a known configuration $\mathbf{q} = \mathbf{q}_*$, where the corresponding solution to (5.1), $\mathbf{q}' = \mathbf{q}'_*$ will be known from the previous computation. At the start of the numerical computation we will know the new value of \mathbf{q} ($= \mathbf{q}_1$ say) and we will need to find \mathbf{q}'_1 such that $\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) = \mathbf{0}$. The fact that we are starting from a solution to (5.1) is also not exploited in the above approach. In our approach we will exploit this structure in the problem at hand and pose the following questions: How much can \mathbf{q} be varied from

the starting configuration, $\mathbf{q} = \mathbf{q}_\star$ and $\mathbf{q}' = \mathbf{q}'_\star$, where $\bar{\psi}(\mathbf{q}'_\star, \mathbf{q}_\star) = \mathbf{0}$, while ensuring the existence of a solution to (5.1)? What is the corresponding maximum variation in \mathbf{q}' ? Can a numerical scheme be used to solve for (5.1)? We will answer these questions in the following lemma and propose a numerical scheme that is different from the popular Newton algorithm. The method used to prove this Lemma is very similar to the approach taken in proving the Implicit Function Theorem [40].

Lemma 5.1 For any given point $\mathbf{q}'_\star \in \mathbb{R}^{n'}$, let $\mathbf{q}_\star = \alpha(\mathbf{q}'_\star)$, suppose $\bar{\psi}(\mathbf{q}'_\star, \mathbf{q}_\star) = \mathbf{0}$ and the following conditions are satisfied throughout the closed balls¹ $B_r(\mathbf{q}_\star)$ and $B_{r'}(\mathbf{q}'_\star)$ for some $r, r' > 0$:

- 5.1. There exists a constant bound c_1 such that $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1}\| \leq c_1$.
- 5.2. $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1}[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]\| \leq \zeta_1 < 1$.
- 5.3. $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1}[\bar{\psi}(\mathbf{q}', \mathbf{q}) - \bar{\psi}(\mathbf{q}', \mathbf{q}_\star)]\| \leq \zeta_2$, where $\zeta_1 r' + \zeta_2 \leq r'$.

Then we have the following conclusions:

- 5.1. There exists a differentiable function $\sigma : B_r(\mathbf{q}_\star) \rightarrow B_{r'}(\mathbf{q}'_\star)$ such that

$$\bar{\psi}(\sigma(\mathbf{q}), \mathbf{q}) = \mathbf{0}, \text{ and}$$

$$\frac{\partial \sigma(\mathbf{q})}{\partial \mathbf{q}} = [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')^{-1} C,$$

where

$$C = \begin{bmatrix} 0_{(n'-n) \times n} \\ I_{n \times n} \end{bmatrix}. \quad (5.2)$$

- 5.2. For each $\mathbf{q} \in B_r(\mathbf{q}_\star)$, the sequence

$$\mathbf{q}'_k = \mathbf{q}'_{k-1} - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1} \bar{\psi}(\mathbf{q}'_{k-1}, \mathbf{q}), k = 1, 2, \dots \quad (5.3)$$

will converge to $\mathbf{q}' = \sigma(\mathbf{q})$.

¹The notation $B_r(\mathbf{x})$ is used to denote a closed ball of radius r centered at \mathbf{x} .

5.3. The error after m iterations will satisfy the following:

$$\|\mathbf{q}' - \mathbf{q}'_m\| \leq \zeta_1^m (1 - \zeta_1)^{-1} \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1} \bar{\psi}(\mathbf{q}', \mathbf{q})\|. \quad (5.4)$$

◇

Proof of Lemma 5.1: Corresponding to each $\mathbf{q} \in B_r(\mathbf{q}_*)$, consider the mapping $\mathbf{h}^q : B_{r'}(\mathbf{q}') \rightarrow \mathbb{R}^{n'}$ given by,

$$\mathbf{h}^q(\mathbf{q}') = \mathbf{q}' - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1} \bar{\psi}(\mathbf{q}', \mathbf{q}). \quad (5.5)$$

Our next objective will be to prove that the mapping (5.5) is a contraction mapping from $B_{r'}(\mathbf{q}')$ into itself. By differentiating (5.5) with respect to \mathbf{q}' ,

$$\begin{aligned} \mathbf{h}_{\mathbf{q}'}^q(\mathbf{q}') &= I - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1} \bar{\psi}_{\mathbf{q}'}(\mathbf{q}') \\ &= [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1} [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*) - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}')]. \end{aligned}$$

From condition 5.2,

$$\|\mathbf{h}_{\mathbf{q}'}^q(\mathbf{q}')\| \leq \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1} [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*) - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}')]\| \leq \zeta_1 < 1, \quad (5.6)$$

for all $\mathbf{q} \in B_r(\mathbf{q}_*)$ and $\mathbf{q}' \in B_{r'}(\mathbf{q}')$. If we define $\mathbf{p}(\mathbf{q}, \mathbf{q}') = [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1} \bar{\psi}(\mathbf{q}', \mathbf{q})$, we have

$$\begin{aligned} \mathbf{h}^q(\mathbf{q}') &= \mathbf{q}' - \mathbf{p}(\mathbf{q}, \mathbf{q}') = \mathbf{q}' - \mathbf{p}(\mathbf{q}_*, \mathbf{q}') + \mathbf{p}(\mathbf{q}_*, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}') \\ &= \mathbf{q}' - \mathbf{p}(\mathbf{q}_*, \mathbf{q}') + \mathbf{p}(\mathbf{q}_*, \mathbf{q}_*) + \mathbf{p}(\mathbf{q}_*, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}'), \end{aligned}$$

since $\mathbf{p}(\mathbf{q}_*, \mathbf{q}_*) = \mathbf{0}$. Now by subtracting \mathbf{q}'_* from both sides

$$\begin{aligned} \mathbf{h}^q(\mathbf{q}') - \mathbf{q}'_* &= \mathbf{q}' - \mathbf{p}(\mathbf{q}_*, \mathbf{q}') - \mathbf{q}'_* + \mathbf{p}(\mathbf{q}_*, \mathbf{q}_*) + \mathbf{p}(\mathbf{q}_*, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}') \\ &= [\mathbf{q}' - \mathbf{p}(\mathbf{q}_*, \mathbf{q}')] - [\mathbf{q}'_* - \mathbf{p}(\mathbf{q}_*, \mathbf{q}_*)] + [\mathbf{p}(\mathbf{q}_*, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}')] \\ &= [\mathbf{h}^{q_*}(\mathbf{q}') - \mathbf{h}^{q_*}(\mathbf{q}'_*)] + [\mathbf{p}(\mathbf{q}_*, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}')] \end{aligned}$$

Now by applying the properties of the norm,

$$\|\mathbf{h}^q(\mathbf{q}') - \mathbf{q}'_\star\| \leq \|\mathbf{h}^{q_\star}(\mathbf{q}') - \mathbf{h}^{q_\star}(\mathbf{q}'_\star)\| + \|\mathbf{p}(\mathbf{q}_\star, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}')\| \quad (5.7)$$

Now since $\|\mathbf{h}^{q_\star}(\mathbf{q}')\| \leq \zeta_1$, by applying the generalization of the Mean Value Theorem [40], we have $\|\mathbf{h}^{q_\star}(\mathbf{q}') - \mathbf{h}^{q_\star}(\mathbf{q}'_\star)\| \leq \zeta_1 \|\mathbf{q}' - \mathbf{q}'_\star\| \leq \zeta_1 r'$ for all $\mathbf{q}' \in B_{r'}(\mathbf{q}'_\star)$. From condition 5.3 we have $\|\mathbf{p}(\mathbf{q}_\star, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}')\| \leq \zeta_2$. By substituting these into (5.7) we have,

$$\|\mathbf{h}^q(\mathbf{q}') - \mathbf{q}'_\star\| \leq \|\mathbf{h}^{q_\star}(\mathbf{q}') - \mathbf{h}^{q_\star}(\mathbf{q}'_\star)\| + \|\mathbf{p}(\mathbf{q}_\star, \mathbf{q}') - \mathbf{p}(\mathbf{q}, \mathbf{q}')\| \leq \zeta_1 r' + \zeta_2 \leq r', \quad (5.8)$$

for all $\mathbf{q} \in B_r(\mathbf{q}_\star)$ and $\mathbf{q}' \in B_{r'}(\mathbf{q}'_\star)$. Therefore, we see from (5.6) and (5.8) that the mapping (5.5) is a contraction mapping from $B_{r'}(\mathbf{q}'_\star)$ into itself. Therefore, from the Contraction Mapping Theorem [40], there exists a fixed point $\mathbf{q}' \in B_{r'}(\mathbf{q}'_\star)$ corresponding to each $\mathbf{q} \in B_r(\mathbf{q}_\star)$ such that

$$\mathbf{q}' = \mathbf{h}^q(\mathbf{q}') = \mathbf{q}' - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1} \bar{\psi}(\mathbf{q}', \mathbf{q}),$$

which implies $\bar{\psi}(\mathbf{q}', \mathbf{q}) = \mathbf{0}$. For each $\mathbf{q} \in B_r(\mathbf{q}_\star)$ we set $\sigma(\mathbf{q})$ to be the fixed point \mathbf{q}' corresponding to the contraction mapping $\mathbf{h}^q(\mathbf{q}')$. Since the existence and uniqueness is guaranteed by the Contraction Mapping Theorem, we see that $\sigma : B_r(\mathbf{q}_\star) \rightarrow B_{r'}(\mathbf{q}'_\star)$ is a function that satisfies $\bar{\psi}(\sigma(\mathbf{q}), \mathbf{q}) = \mathbf{0}$. We see further that conclusion 5.2 follows immediately since the sequence (5.3) is the same as the contraction mapping (5.5). The error estimate of conclusion 3 is derived as follows: from the Contraction Mapping Theorem, since \mathbf{q}' is the fixed point of the contraction mapping (5.5),

$$\begin{aligned} \|\mathbf{q}' - \mathbf{q}'_\star\| &\leq \zeta_1^m (1 - \zeta_1)^{-1} \|\mathbf{h}^q(\mathbf{q}'_\star) - \mathbf{q}'_\star\| \\ &= \zeta_1^m (1 - \zeta_1)^{-1} \|\mathbf{q}'_\star - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1} \bar{\psi}(\mathbf{q}'_\star, \mathbf{q}) - \mathbf{q}'_\star\| \\ &= \zeta_1^m (1 - \zeta_1)^{-1} \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1} \bar{\psi}(\mathbf{q}'_\star, \mathbf{q})\|, \end{aligned}$$

is true for all \mathbf{q}'_m , $m = 0, 1, \dots$. We have left to prove the differentiability of $\mathbf{q}' = \sigma(\mathbf{q})$. We begin by proving continuity. Let $\mathbf{q}_1, \mathbf{q}_2 \in B_r(\mathbf{q}_*)$ and $\mathbf{q}'_1 = \sigma(\mathbf{q}_1)$ and $\mathbf{q}'_2 = \sigma(\mathbf{q}_2)$. Since \mathbf{q}'_1 and \mathbf{q}'_2 are the fixed points of the contractions $\mathbf{h}^{q_1}(\mathbf{q}')$ and $\mathbf{h}^{q_2}(\mathbf{q}')$,

$$\mathbf{q}'_1 - \mathbf{q}'_2 = \mathbf{h}^{q_1}(\mathbf{q}'_1) - \mathbf{h}^{q_2}(\mathbf{q}'_2) = \mathbf{h}^{q_1}(\mathbf{q}'_1) - \mathbf{h}^{q_1}(\mathbf{q}'_2) + \mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2).$$

Therefore,

$$\begin{aligned} \|\mathbf{q}'_1 - \mathbf{q}'_2\| &\leq \|\mathbf{h}^{q_1}(\mathbf{q}'_1) - \mathbf{h}^{q_1}(\mathbf{q}'_2)\| + \|\mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2)\| \\ &\leq \zeta_1 \|\mathbf{q}'_1 - \mathbf{q}'_2\| + \|\mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2)\|, \end{aligned}$$

since $\|\mathbf{h}^{q_1}(\mathbf{q}'_1) - \mathbf{h}^{q_1}(\mathbf{q}'_2)\| \leq \zeta_1 \|\mathbf{q}'_1 - \mathbf{q}'_2\|$ for any two points \mathbf{q}'_1 and \mathbf{q}'_2 (since $\mathbf{h}^{q_1}(\mathbf{q}')$ is a contraction). By re-organizing the terms,

$$(1 - \zeta_1) \|\mathbf{q}'_1 - \mathbf{q}'_2\| \leq \|\mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2)\| \quad (5.9)$$

It can be seen from (5.5), that

$$\begin{aligned} \mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2) &= [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]^{-1} (\bar{\psi}(\mathbf{q}'_2, \mathbf{q}_1) - \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2)) \\ &= [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]^{-1} \left(- \begin{bmatrix} 0 \\ \mathbf{q}_1 - \mathbf{q}_2 \end{bmatrix} \right), \end{aligned}$$

by substituting from (4.7). Therefore,

$$\|\mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2)\| \leq \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]^{-1}\| \|\mathbf{q}_1 - \mathbf{q}_2\|.$$

By substituting the above into (5.9),

$$\begin{aligned} \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2)\| = \|\mathbf{q}'_1 - \mathbf{q}'_2\| &\leq \frac{1}{(1 - \zeta_1)} \|\mathbf{h}^{q_1}(\mathbf{q}'_2) - \mathbf{h}^{q_2}(\mathbf{q}'_2)\| \\ &\leq \frac{1}{(1 - \zeta_1)} \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]^{-1}\| \|\mathbf{q}_1 - \mathbf{q}_2\| \\ &\leq \frac{c_1}{(1 - \zeta_1)} \|\mathbf{q}_1 - \mathbf{q}_2\|, \end{aligned} \quad (5.10)$$

by substituting from Condition 5.1. Thus we have proved that given any $\epsilon > 0$, there exists $\delta \left(= \frac{(1-\zeta_1)\epsilon}{c_1} \right) > 0$ such that $\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta \Rightarrow \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2)\| < \epsilon$ proving the continuity of $\mathbf{q}' = \sigma(\mathbf{q})$. Our next objective is to prove differentiability, that is, to prove that given $\epsilon > 0$, there exists $\delta > 0$ such that

$$\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta \Rightarrow \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}C(\mathbf{q}_1 - \mathbf{q}_2)\| \leq \epsilon\|\mathbf{q}_1 - \mathbf{q}_2\|,$$

for all $\mathbf{q}_1, \mathbf{q}_2 \in B_r(\mathbf{q}_*)$, where $\mathbf{q}'_2 = \sigma(\mathbf{q}_2)$, $\mathbf{q}'_1 = \sigma(\mathbf{q}_1)$, and C is defined in (5.2).

Consider the term on the left hand side of the inequality,

$$\begin{aligned} & \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}C(\mathbf{q}_1 - \mathbf{q}_2)\| \\ &= \|\mathbf{q}'_1 - \mathbf{q}'_2 - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}C(\mathbf{q}_1 - \mathbf{q}_2)\| \\ &= \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}([\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2) - C(\mathbf{q}_1 - \mathbf{q}_2))\| \\ &\leq \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}\| \|([\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2) - C(\mathbf{q}_1 - \mathbf{q}_2))\| \\ &\leq c_1 \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2) - C(\mathbf{q}_1 - \mathbf{q}_2)\| \end{aligned} \quad (5.11)$$

by invoking Condition 5.1. Note that since $\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) = \mathbf{0}$ and $\bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) = \mathbf{0}$,

$$\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) + \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) + \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) = \mathbf{0}.$$

By inserting the above zero quantity into the right hand side of (5.11) we have

$$\begin{aligned} & \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}C(\mathbf{q}_1 - \mathbf{q}_2)\| \\ &\leq c_1 \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2) - C(\mathbf{q}_1 - \mathbf{q}_2) \\ &\quad - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) + \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) + \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2)\|, \end{aligned}$$

By reorganizing the terms we have

$$\begin{aligned} & \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1}C(\mathbf{q}_1 - \mathbf{q}_2)\| \\ &\leq c_1 \|-\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) + \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) + [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2)\| \end{aligned}$$

$$\begin{aligned}
& - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) + \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - C(\mathbf{q}_1 - \mathbf{q}_2) \| \\
& \leq c_1 \left\{ \| - \left(\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2) \right) \| \right. \\
& + \left. \| - \left(\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) + C(\mathbf{q}_1 - \mathbf{q}_2) \right) \| \right\} \\
& = c_1 \| \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2) \| \\
& + c_1 \| \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) + C(\mathbf{q}_1 - \mathbf{q}_2) \|. \tag{5.12}
\end{aligned}$$

Since $\bar{\psi}(\mathbf{q}', \mathbf{q})$ is differentiable and $\frac{\partial \bar{\psi}(\mathbf{q}', \mathbf{q})}{\partial \mathbf{q}'} = \bar{\psi}_{\mathbf{q}'}(\mathbf{q}')$, given $\epsilon_1 = \frac{\epsilon(1-\zeta_1)}{2(c_1)^2}$, there exists δ_1 such that

$$\|\mathbf{q}'_1 - \mathbf{q}'_2\| < \delta_1 \Rightarrow \|\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2)\| \leq \epsilon_1 \|\mathbf{q}'_1 - \mathbf{q}'_2\|. \tag{5.13}$$

Now, due to continuity of $\mathbf{q}' = \sigma(\mathbf{q})$, given $\epsilon_2 = \delta_1$, there exists δ_2 such that $\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta_2 \Rightarrow \|\mathbf{q}'_1 - \mathbf{q}'_2\| < \delta_1$ and hence, by substituting into (5.13),

$$\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta_2 \Rightarrow$$

$$\|\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2)\| \leq \epsilon_1 \|\mathbf{q}'_1 - \mathbf{q}'_2\| \leq \epsilon_1 \frac{c_1}{(1-\zeta_1)} \|\mathbf{q}_1 - \mathbf{q}_2\|,$$

by using (5.10). Now by substituting for ϵ_1 ,

$$\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta_2 \Rightarrow$$

$$\|\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) - \bar{\psi}(\mathbf{q}'_2, \mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)](\mathbf{q}'_1 - \mathbf{q}'_2)\| \leq \frac{\epsilon}{2c_1} \|\mathbf{q}_1 - \mathbf{q}_2\|. \tag{5.14}$$

Now since $\frac{\partial \bar{\psi}(\mathbf{q}', \mathbf{q})}{\partial \mathbf{q}} = -C$, given $\epsilon_3 = \frac{\epsilon}{2c_1}$, there exists δ_3 such that

$$\begin{aligned}
\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta_3 \Rightarrow \|\bar{\psi}(\mathbf{q}'_1, \mathbf{q}_1) - \bar{\psi}(\mathbf{q}'_1, \mathbf{q}_2) + C(\mathbf{q}_1 - \mathbf{q}_2)\| &\leq \epsilon_3 \|\mathbf{q}_1 - \mathbf{q}_2\| \\
&= \frac{\epsilon}{2c_1} \|\mathbf{q}_1 - \mathbf{q}_2\|. \tag{5.15}
\end{aligned}$$

By selecting $\delta = \min\{\delta_2, \delta_3\}$ we have the conclusions of (5.14) and (5.15) valid whenever $\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta$. By substituting these into (5.12), we have

$$\|\mathbf{q}_1 - \mathbf{q}_2\| < \delta \Rightarrow \|\sigma(\mathbf{q}_1) - \sigma(\mathbf{q}_2) - [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_2)]^{-1} C(\mathbf{q}_1 - \mathbf{q}_2)\| \leq \epsilon \|\mathbf{q}_1 - \mathbf{q}_2\|.$$

This completes the proof of this lemma. \square

Our next objective will be to characterize a region of the robot workspace where the condition of Lemma 5.1 will be valid. It will be seen later that these conditions are valid if the quantity $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1}\|$ is bounded for all points satisfying the constraints $\phi(\mathbf{q}') = 0$. This motivates us to define a set \mathcal{D}' as follows:

Definition 5.1 The set \mathcal{D}' is defined to be a closed interval in $\mathbb{R}^{n'}$, characterized by $l_1, l_2, \dots, l_{n'}$ and $h_1, h_2, \dots, h_{n'}$ where $l_i \leq q'_i \leq h_i$, $i = 1, 2, \dots, n'$ for all $\mathbf{q}' \in \mathcal{D}'$, for which a constant c_1 satisfying the following condition exists:

Condition: $\phi(\mathbf{q}') \neq 0$ or $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')]^{-1}\| \leq c_1$ for all $\mathbf{q}' \in \mathcal{D}'$.

\diamond

In the following lemma we prove that the conditions of Lemma 5.1 are satisfied at any point within \mathcal{D}' that satisfies the constraints $\phi(\mathbf{q}') = 0$.

Lemma 5.2 Given $\mathbf{q}'_* \in \mathcal{D}'$, let $\mathbf{q}_* = \alpha(\mathbf{q}'_*)$, if $\phi(\mathbf{q}'_*) = 0$, the following conditions are satisfied throughout $B_{r'}(\mathbf{q}'_*)$ and $B_r(\mathbf{q}_*)$ for any given $0 < \zeta_1 < 1$:

$$5.1. \quad \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]^{-1}[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]\| \leq \zeta_1 < 1.$$

$$5.2. \quad \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_*)]^{-1}[\bar{\psi}(\mathbf{q}', \mathbf{q}) - \bar{\psi}(\mathbf{q}', \mathbf{q}_*)]\| \leq \zeta_2, \text{ where } \zeta_1 r' + \zeta_2 \leq r'.$$

where

$$r' = \frac{\zeta_1}{c_1 c_2}, \quad (5.16)$$

and ζ_1 is a constant that should be selected such that $0 < \zeta_1 < 1$. The constant c_2 is given by

$$c_2 = \|B\|, \quad (5.17)$$

where B is a matrix composed of elements $b_{i,j}$, $i = 1, \dots, n'$, $j = 1, \dots, n'$ satisfying the following:

$$\left\| \frac{\partial [\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')]_{i,j}}{\partial \mathbf{q}'} \right\| \leq b_{i,j}, \quad (5.18)$$

for all $\mathbf{q}' \in \mathcal{D}'$ and $[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')]_{i,j}$ denotes the $(i, j)^{th}$ element of $\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')$. The radius r is given by

$$r = \frac{\zeta_1(1 - \zeta_1)}{c_1^2 c_2}. \quad (5.19)$$

◇

Proof of Lemma 5.2: Due to the assumed differentiability properties, there exists bounds $b_{i,j}$ on the partial derivatives of each of the elements of $\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')$ satisfying (5.18) for all $\mathbf{q}' \in \mathcal{D}'$. Therefore, by applying the generalization of the Mean Value Theorem [6],

$$|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')]_{i,j}| \leq b_{i,j} \|\mathbf{q}' - \mathbf{q}_\star'\| = \bar{b}_{i,j} \text{ (say)}, \quad (5.20)$$

for all $\mathbf{q}' \in \mathcal{D}'$. Here the notation $[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')]_{i,j}$ is used to denote the $(i, j)^{th}$ element of $\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')$. Let \bar{B} be a matrix whose $(i, j)^{th}$ element is $\bar{b}_{i,j} = b_{i,j} \|\mathbf{q}' - \mathbf{q}_\star'\|$. Now by applying Lemma 2.4 to (5.20),

$$\|\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')\| \leq \|\bar{B}\| \leq \|B\| \|\mathbf{q}' - \mathbf{q}_\star'\| = c_2 \|\mathbf{q}' - \mathbf{q}_\star'\|, \quad (5.21)$$

where the second inequality is obtained by factoring out the scalar common factor $\|\mathbf{q}' - \mathbf{q}_\star'\|$. Therefore,

$$\begin{aligned} \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')]^{-1} (\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star'))\| &\leq \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')]^{-1}\| \|\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') - \bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')\| \\ &\leq c_1 c_2 \|\mathbf{q}' - \mathbf{q}_\star'\|, \end{aligned}$$

by substituting from (5.21) since $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')]^{-1}\| \leq c_1$ as $\mathbf{q}_\star' \in \mathcal{D}'$ and $\phi(\mathbf{q}_\star') = \mathbf{0}$. Therefore, by selecting r' as in (5.16), Conclusion 5.1 of this lemma is satisfied

throughout $B'_r(\mathbf{q}'_\star)$. Before deriving Conclusion 5.2, we select ζ_2 to be,

$$\zeta_2 = (1 - \zeta_1)r', \quad (5.22)$$

which will cause the automatic satisfaction of $\zeta_1 r' + \zeta_2 \leq r'$. It can be seen from (4.7) that

$$\|\bar{\psi}(\mathbf{q}', \mathbf{q}) - \bar{\psi}(\mathbf{q}', \mathbf{q}_\star)\| = \|\mathbf{q} - \mathbf{q}_\star\|.$$

and hence,

$$\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1} (\bar{\psi}(\mathbf{q}', \mathbf{q}) - \bar{\psi}(\mathbf{q}', \mathbf{q}_\star))\| \leq c_1 \|\mathbf{q} - \mathbf{q}_\star\|.$$

Therefore, by selecting $\mathbf{r} = \frac{\zeta_2}{c_1} = \frac{(1-\zeta_1)r'}{c_1} = \frac{\zeta_1(1-\zeta_1)}{c_1^2 c_2}$, we have Conclusion 5.2 of this lemma satisfied throughout $B'_r(\mathbf{q}'_\star)$ and $B_r(\mathbf{q}_\star)$. This completes the proof of this lemma. \square

Now we are ready to present the first main result of this chapter.

Theorem 5.1 Let \mathcal{D}' be a closed set satisfying Definition 5.1. For any point $\mathbf{q}'_\star \in \mathcal{D}'$ such that $\phi(\mathbf{q}'_\star) = \mathbf{0}$, let $\mathbf{q}_\star = \alpha(\mathbf{q}'_\star)$, the parameterization (1.9) is valid throughout $B_r(\mathbf{q}_\star)$. \diamond

Proof of Theorem 5.1: Follows from Lemma 5.1 and Lemma 5.2. \square

5.2 Extending the domain to an arbitrary trajectory

Can the domain be extended? We can argue that (1.9) is valid along any arbitrary trajectory that is contained in \mathcal{D}' as follows: Consider any given trajectory, that is contained in \mathcal{D}' , starting from a point $\mathbf{q}'_0 \in \mathcal{D}'$ and ending in $\mathbf{q}'_n \in \mathcal{D}'$. Let's denote $\alpha(\mathbf{q}')$ by \mathbf{q} for any point \mathbf{q}' in \mathcal{D}' . Hence $\alpha(\mathbf{q}'_0) = \mathbf{q}_0$ and $\alpha(\mathbf{q}'_n) = \mathbf{q}_n$. Therefore, the trajectory can be represented in the joint space by a sequence of points starting \mathbf{q}_0 and ending at \mathbf{q}_n . We next select a sequence of points \mathbf{q}_i , $i = 0, 1, \dots, n$ along the

5.3 The Real Time Computation of $\mathbf{q}' = \sigma(\mathbf{q})$

In this section we address the issue of computing $\mathbf{q}' = \sigma(\mathbf{q})$ in real time. We first present sufficient conditions to ensure convergence of the numerical scheme (5.3) and then propose a procedure for implementing control in parallel robots.

It can be seen from Lemma 5.1 that if we start from a solution $\bar{\psi}(\mathbf{q}', \mathbf{q}) = 0$ and \mathbf{q} remains within $B_r(\mathbf{q})$, then the numerical scheme (5.3) will converge. Therefore, if a numerical computation is initiated sufficiently frequently to ensure that \mathbf{q} remains within $B_r(\mathbf{q})$ from the time the previous numerical computation converged, then convergence can be guaranteed. In Theorem 5.2 we derive conditions on the frequency at which the numerical scheme must be initiated in order to ensure convergence. A second important issue that is addressed in Theorem 5.2 is the condition for terminating the numerical computations. If the numerical scheme is made to iterate until convergence to within a desired tolerance, it may hold up other important duties of the controller such as servicing the sensors, updating the motors etc. For this reason it's safer to set up the numerical scheme to terminate after a fixed number of iterations. In this case however, there might still be substantial error even after the numerical computation. This problem is also addressed in Theorem 5.2 where the number of iterations required to reach a desired tolerance level is derived.

Before presenting Theorem 5.2, we define some terms that become important in the remainder of this paper. In the implementation of control in a digital system the control system reads sensory data and update the actuator inputs at a particular frequency commonly referred to as the "sampling frequency" (f_s say). Let's denote the corresponding "sampling period" by T_s . In a parallel robot, in addition to selecting the sampling period we also need to select a frequency for initiating the numerical computation of $\mathbf{q}' = \sigma(\mathbf{q})$. This frequency, which we will refer to as "initiation frequency" and denote by f_i , becomes important in the results of this chapter.

We will denote the corresponding initiation period by T_i . Note that the initiation frequency refers to how often in time the numerical computation must be started and it's different from the step size of iterations during computations.

5.4 Conditions for Convergence

Theorem 5.2 Given a parallel robot following a trajectory contained within a closed interval D' satisfying Definition 5.1, if the initiation period T_i is selected such that the joint velocity vector $\dot{\mathbf{q}}$ satisfies

$$\|\dot{\mathbf{q}}\| \leq \frac{r}{T_i}, \quad (5.23)$$

the numerical scheme (5.3) will converge to any desired tolerance e within m iterations where

$$m \geq \frac{\ln\left(\frac{e(1-\zeta_1)}{c_1 r}\right)}{\ln(\zeta_1)}, \quad (5.24)$$

where r is given by (5.19). The desired tolerance must be selected such that it is negligible when compared to r' (given by (5.16)). \diamond

Proof of Theorem 5.2: We will first prove that condition (5.23) ensures that the trajectory remain within a ball of radius r during each initiation period T_i . We denote by $\mathbf{q}(t)$ the robot position at time t and the corresponding joint velocity by $\dot{\mathbf{q}}(t)$. From the generalization of the Mean Value Theorem [6], there exists a time t_* , $t < t_* < t + T_i$ such that

$$\|\mathbf{q}(t + T_i) - \mathbf{q}(t)\| \leq \|\dot{\mathbf{q}}(t_*)\|T_i.$$

Therefore, if $\dot{\mathbf{q}}(t)$ satisfies (5.23),

$$\|\mathbf{q}(t + T_i) - \mathbf{q}(t)\| \leq \|\dot{\mathbf{q}}(t_*)\|T_i \leq r$$

for any given time t . Hence, the trajectory remains inside a ball of radius r during each initiation period. Now we will prove that the numerical scheme will converge to tolerance level e within m iterations. Note that since $\bar{\psi}(\mathbf{q}_\star, \mathbf{q}'_\star) = \mathbf{0}$,

$$\|\bar{\psi}(\mathbf{q}, \mathbf{q}'_\star)\| = \|\bar{\psi}(\mathbf{q}, \mathbf{q}'_\star) - \bar{\psi}(\mathbf{q}_\star, \mathbf{q}'_\star)\| = \|\mathbf{q} - \mathbf{q}_\star\|,$$

by substituting from (4.7). Now since the trajectory remains within a ball of radius r during each initiation period, when the numerical scheme is initiated $\|\mathbf{q} - \mathbf{q}_\star\| \leq r$ and hence,

$$\|\bar{\psi}(\mathbf{q}, \mathbf{q}'_\star)\| \leq r.$$

Therefore from (5.4), the error after m iterations satisfies

$$\begin{aligned} \|\mathbf{q}' - \mathbf{q}'_m\| &\leq \zeta_1^m (1 - \zeta_1)^{-1} \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1} \bar{\psi}(\mathbf{q}, \mathbf{q}'_\star)\| \\ &\leq \zeta_1^m (1 - \zeta_1)^{-1} \|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}'_\star)]^{-1}\| \|\bar{\psi}(\mathbf{q}, \mathbf{q}'_\star)\| \\ &\leq \frac{\zeta_1^m c_1 r}{1 - \zeta_1}, \end{aligned} \tag{5.25}$$

by invoking condition 1 of Lemma 5.1 and substituting from above. However, if m satisfies (5.24), by multiplying both sides of (5.24) by $\ln(\zeta_1)$ we have

$$m \ln(\zeta_1) \leq \ln \left(\frac{e(1 - \zeta_1)}{c_1 r} \right)$$

(note the change in direction of the inequality since $\ln(\zeta_1) < 0$). By taking the exponential of both sides,

$$\zeta_1^m \leq \frac{e(1 - \zeta_1)}{c_1 r},$$

and hence we have $\|\mathbf{q}' - \mathbf{q}'_m\| \leq e$ by substituting into (5.25). This completes the proof of Theorem 5.2. □

Note that Theorem 5.2 is independent of Theorem 5.1 and hence, if for some other algorithm the region where convergence is assured is derived and an error estimate is available, Theorem 5.2 can be used to obtain conditions for convergence.

5.5 Procedure for Implementing Control in Parallel Robots

Based on the results of this paper we propose the following algorithm for implementing control:

- 5.1. Based on the application select a closed interval \mathcal{D}' in $\mathbb{R}^{n'}$ that is free of singularities (If a potential choice of \mathcal{D}' contains singularities the desired trajectory must be redefined).
- 5.2. Perform a numerical optimization to compute $c_1 = \max \|\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')^{-1}\|$ subject to $\phi(\mathbf{q}') = \mathbf{0}$, in \mathcal{D}' .
- 5.3. Compute bounds $b_{i,j}$ for the partial derivatives of the elements of $[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')]_{i,j}$ satisfying (5.18). Often these bounds can be obtained by inspection. If this is not possible they can also be computed by numerical schemes. Compute c_2 from (5.17).
- 5.4. Select $0 < \zeta < 1$ (let $\zeta = 0.5$) and compute r' and r from equations (5.16) and (5.19) respectively.
- 5.5. Select the initiation frequency, f_i , such that (5.23) is satisfied and set the number of iterations to m such that (5.24) is satisfied.
- 5.6. Program the numerical algorithm (5.3) according to the above requirements and implement control. Note that (5.3) will iterate faster than the popular Newton algorithm since the matrix $[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_*)]^{-1}$ needs to be computed only once during each implementation (it remains constant during iterations). For this reason (5.3) has less computations when compared to the popular Newton algorithm.

5.6 Trajectory Tracking Experiment

In this section we will present experimental results on trajectory tracking of the R.P.D.R. The implementation was based on the procedure derived in the previous section. For this application we chose a trajectory that will make the end effector follow a circle of diameter 6.0 inches that is centered 9 inches above the axes of joints one and two (see Figure 5.3). This translates into ranges of motion $-24^\circ \leq q_1 \leq 13^\circ$, $168^\circ \leq q_2 \leq 204^\circ$, $100^\circ \leq q_3 \leq 137^\circ$, $-138^\circ \leq q_4 \leq -102^\circ$.

5.7 Implementation of the Procedure Defined in Section 5.5

For this motion we followed the procedure proposed in the previous section as follows:

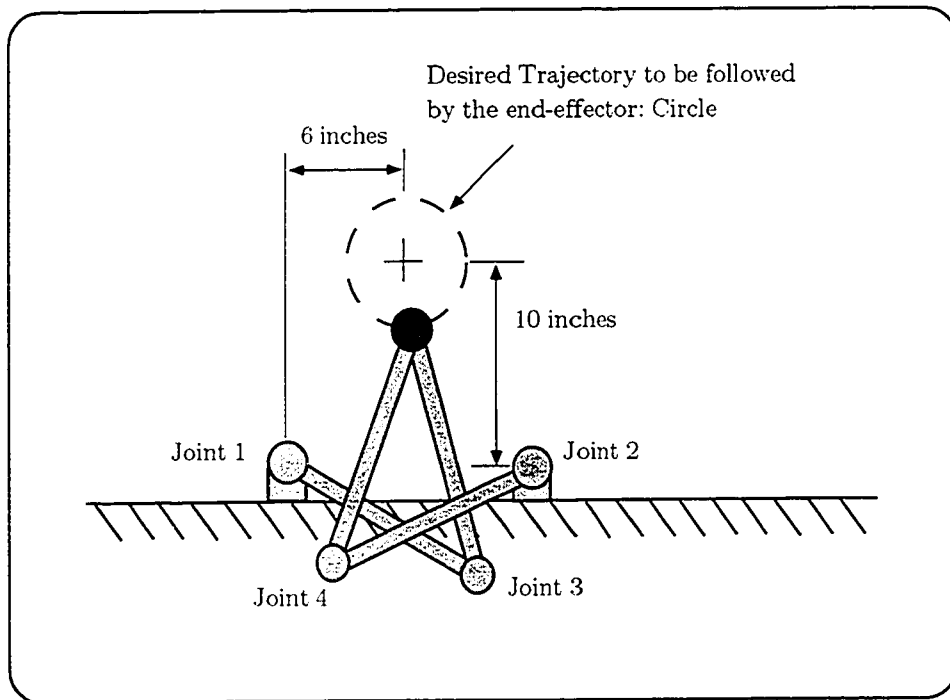


Figure 5.3: The Trajectory to be Followed by the RPDR

Step 1: Based on the range of motion of the joints we selected \mathcal{D}' as follows:

$$\mathcal{D}' \triangleq \left\{ \mathbf{q}' \in \mathbb{R}^{n'} : -30^\circ \leq q'_1 \leq 30^\circ, 150^\circ \leq q'_2 \leq 210^\circ, 95^\circ \leq q'_3 \leq 151^\circ, \right. \\ \left. -151^\circ \leq q'_4 \leq -95^\circ \right\}.$$

Step 2: The bound for $\|[\bar{\psi}_{\mathbf{q}'}(\mathbf{q}_\star')]^{-1}\|$ corresponding to \mathcal{D}' was computed to be $c_1 = 5.7$.

Step 3: For the R.P.D.R. we have the following expression for $\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')$:

$$\bar{\psi}_{\mathbf{q}'}(\mathbf{q}') = \begin{bmatrix} \bar{\psi}_{q'}(1,1) & \bar{\psi}_{q'}(1,2) & -a_3 \sin(q_1 + q_3) & a_4 \sin(q_2 + q_4) \\ \bar{\psi}_{q'}(2,1) & \bar{\psi}_{q'}(2,2) & a_3 \cos(q_1 + q_3) & -a_4 \cos(q_2 + q_4) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

where, $\bar{\psi}_{q'}(1,1) = -a_1 \sin(q_1) - a_3 \sin(q_1 + q_3)$, $\bar{\psi}_{q'}(1,2) = a_2 \sin(q_2) + a_4 \sin(q_2 + q_4)$, $\bar{\psi}_{q'}(2,1) = a_1 \cos(q_1) + a_3 \cos(q_1 + q_3)$, and $\bar{\psi}_{q'}(2,2) = -a_2 \cos(q_2) - a_4 \cos(q_2 + q_4)$. While it's possible to obtain bounds $b_{i,j}$ satisfying (5.18) globally by inspection, in order to obtain less conservative bounds we searched for the maximum magnitudes of each of the derivatives the elements of $\bar{\psi}_{\mathbf{q}'}(\mathbf{q}')$ in \mathcal{D}' and set each $b_{i,j}$ to be that value. Hence we obtained less conservative bounds $b_{i,j}$ that are valid within \mathcal{D}' . Based on these, we computed c_2 as follows:

$$c_2 = \|B\| = \begin{bmatrix} 0.39 & 0.39 & 0.43 & 0.43 \\ 0.40 & 0.40 & 0.43 & 0.43 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = 1.17.$$

Step 4: For $\zeta_1 = 0.5$, we obtain $r' = 0.0154$ rad. and $r = 0.0066$ rad.

Step 5: In order to compute initiation period, T_i , such that (5.23) is satisfied, we need to obtain the maximum value of $\|\dot{\mathbf{q}}\|$. Since the velocity is dependent on the time taken to traverse the trajectory, we chose to complete the trajectory in 0.5 seconds. Parts (a) and (b) of Figure 5.4 shows the joint velocities \dot{q}_1 and \dot{q}_2 corresponding to

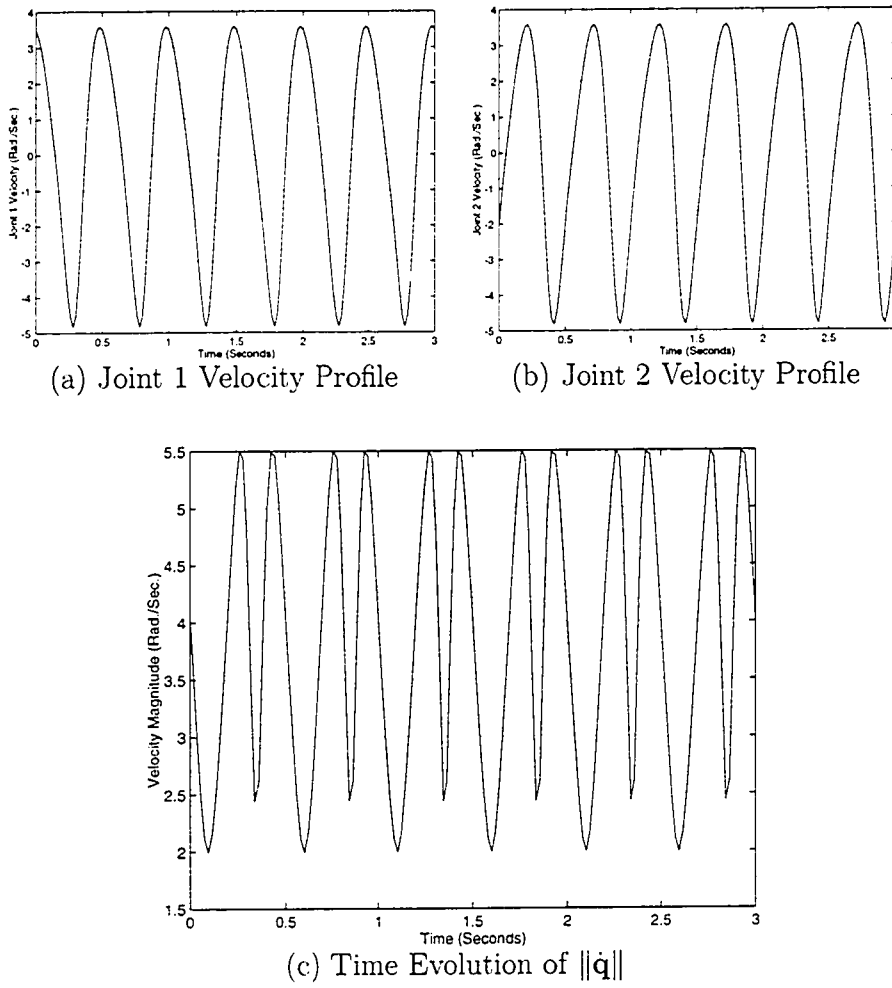


Figure 5.4: The Joint Velocities and the Corresponding Magnitude of the Velocity Vector for the Selected Motion.

this motion. Part (c) shows a plot of $\|\dot{\mathbf{q}}\| = \sqrt{\dot{q}_1^2 + \dot{q}_2^2}$. Based on this plot we assume $\|\dot{\mathbf{q}}\| < 6.0$ rad./sec. Hence, by selecting

$$T_i \leq \frac{r}{6.0} = 1.1 \text{ milliseconds,}$$

that is, if we select the initiation frequency, f_i , to be greater than or equal to 910 Hz. we can satisfy the requirements of Theorem 5.2.

Based on the resolution of the encoders of joints 1 and 2, we selected the desired tolerance level, e , for the numerical scheme to be 0.0005 radians. From (5.24) we obtain the minimum number of iterations m required to guarantee convergence as,

$$m \geq \frac{\ln\left(\frac{e(1-\zeta_1)}{c_1 r}\right)}{\ln(\zeta_1)} = 7.23.$$

That is, in order to satisfy the conditions of Theorem 5.2, we must select the initiation frequency, f_i to be greater than 910 Hz. and set the number of iterations to be 8.

Step 6: The Inverse Dynamics control law was implemented in the R.P.D.R. with the algorithm (5.3) implemented such that the above requirements were satisfied. In the next section we will present the experimental results.

5.8 Experimental Results

We began by computing $\ddot{\mathbf{q}}^d$, $\dot{\mathbf{q}}^d$, and \mathbf{q}^d corresponding to the desired motion. An initialization routine was written to ensure that the robot reached the first point in the trajectory with the desired configuration (that is the correct \mathbf{q}' corresponding to \mathbf{q}^d). The control law implemented was the well known inverse dynamics control law [66],

$$\mathbf{u} = D(\mathbf{q})\mathbf{v} + C(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}),$$

where

$$\mathbf{v} = \ddot{\mathbf{q}}^d + K_d(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + K_p(\mathbf{q}^d - \mathbf{q}),$$

with

$$K_p = \begin{bmatrix} 900 & 0 \\ 0 & 900 \end{bmatrix}, K_d = \begin{bmatrix} 60 & 0 \\ 0 & 60 \end{bmatrix}.$$

There was no compensation for friction. The algorithm (5.3) was set up to compute $\mathbf{q}' = [q'_1 \ q'_2 \ q'_3 \ q'_4]$ corresponding to the measured value of $\mathbf{q} = [q_1 \ q_2]$. We

set the number of iterations for the numerical computation (5.3) to be 8 based on Step 5 above. We conducted the first experiment with initiation period $T_i = 1$ millisecond (corresponding to 1 KHz.). Hence, in this experiment all the requirements of Theorem 5.1 were satisfied. Figure 5.5 shows the tracking results. Part (a) shows a plot of the desired trajectory of joint 1, q_1^d , (solid) and the actual trajectory of joint 1, q_1 , (dashed). Part (b) shows the corresponding plots for Joint 2. Much of the tracking error that can be observed can be attributed to frictional effects since the error present becomes large whenever the corresponding joint changes direction, the point at which the effect of coulomb friction is dominant. Other errors can be due to modeling errors and unmodeled actuator dynamics. In general we see the system recovering from the errors introduced at points of direction change.

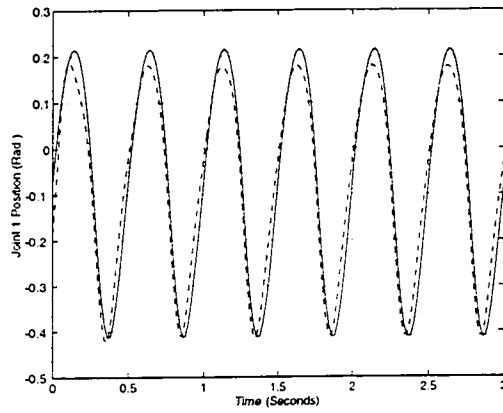
For the R.P.D.R., it's possible to compute $\mathbf{q}' = [q'_1 \ q'_2 \ q'_3 \ q'_4]$ analytically. This presented an ideal opportunity to track the actual error in the numerical computation (5.3). Since $q'_1 = q_1$ and $q'_2 = q_2$ for the R.P.D.R., there was no error associated with computing them. Parts (c) and (d) of Figure 5.5 shows the initial error before (5.3) is implemented, corresponding to the computation of q'_3 and q'_4 respectively. Parts (e) and (f) of Figure 5.5 shows the final error after 8 iterations of (5.3). It can be seen that the final error is in the range of 0.000001 rad., well within the selected tolerance $e = 0.0005$ rad. Hence, we see that the results agree with the conclusions of Theorem 5.2. In the next experiments we observed the effect of increasing T_i , violating Theorem 5.2. Due to space limitations we present only two additional experiments corresponding to $T_i = 10$ milliseconds (Figure 5.6) and $T_i = 100$ milliseconds (Figure 5.7). The results have the same format. It can be seen from parts (c) and (d) that the initial error increases dramatically as expected. It can be seen however, that even at $T_i = 10$ milliseconds the final error is still well within the desired tolerance level e (Parts (e) and (f) of Figure 5.6). Therefore, the

trajectory tracking was unaffected. At $T_i = 100$ milliseconds, we see the numerical iteration unable to cope with the large error and at times diverging. This finally causes the tracking to fail (See Part (b) of Figure 5.7). Hence, the experimental results confirmed the conclusions of Theorem 5.2.

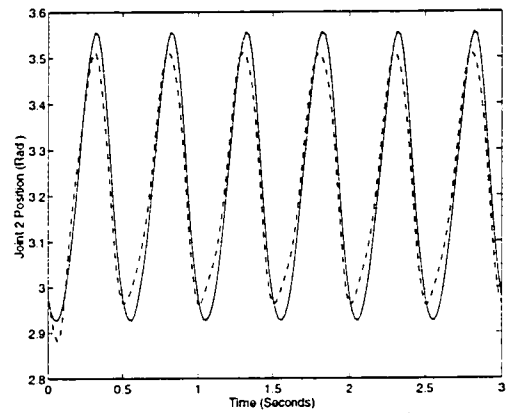
5.9 Summary

In this chapter we addressed two important issues associated with the implementation of the dynamics formulation for parallel robots presented in [24]. The first issue is the explicit characterization of a set where the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ is valid. Since the equations of motion are valid only in a set where the parameterization is valid, in control the domain of attraction for a given control law will be a subset of such a set. Hence, it's useful to characterize the full region surrounding a point in the workspace where $\mathbf{q}' = \sigma(\mathbf{q})$ is valid. In this chapter we partially answered this question by characterizing a ball surrounding a point \mathbf{q} , where $\mathbf{q}' = \sigma(\mathbf{q})$ is valid. The full set surrounding \mathbf{q} where $\mathbf{q}' = \sigma(\mathbf{q})$ is valid may be larger and hence, further research needs to be done in order to fully resolve this issue. The more important contribution of this chapter in this regard is the characterization of a region, \mathcal{D}' , of the robot workspace such that the parameterization $\mathbf{q}' = \sigma(\mathbf{q})$ is valid along any given trajectory contained within \mathcal{D}' . The set \mathcal{D}' is sufficiently large to allow a trajectory followed by the robot in a typical application to remain entirely within \mathcal{D}' . This result will be useful in control synthesis for selecting the region of the workspace of a parallel robot where control is to be implemented.

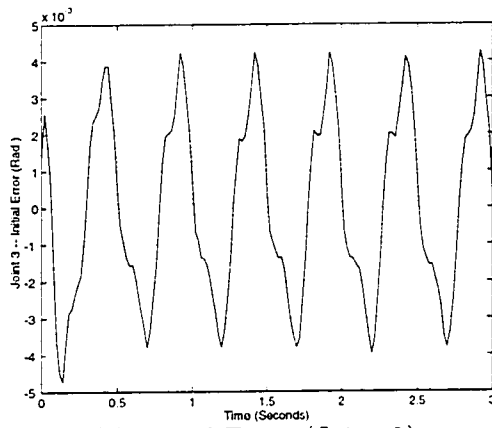
The second issue that was addressed in this chapter is the real time computation of $\mathbf{q}' = \sigma(\mathbf{q})$ for model based control. We proposed a numerical algorithm that is easier to compute when compared to the popular Newton algorithm, and derived conditions that ensure convergence. Based on this result we derived a procedure for



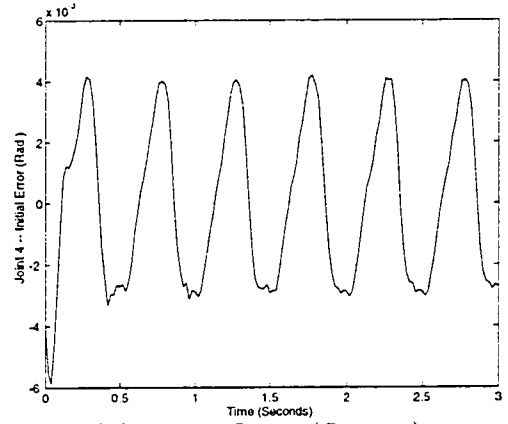
(a) Desired (solid) and Actual (Dashed) Trajectory Profile of Joint 1



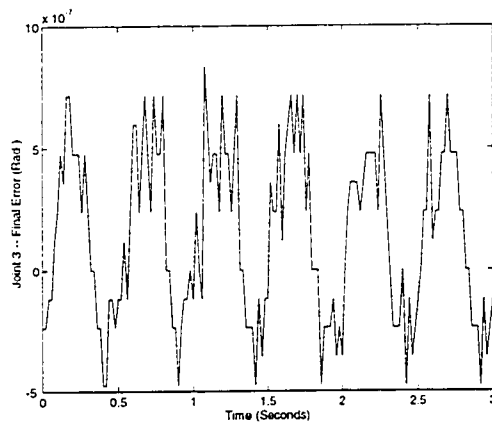
(b) Desired (solid) and Actual (Dashed) Trajectory Profile of Joint 2



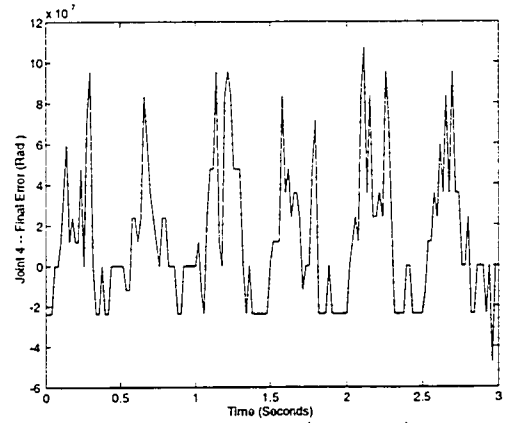
(c) Initial Error (Joint 3)



(d) Initial Error (Joint 4)

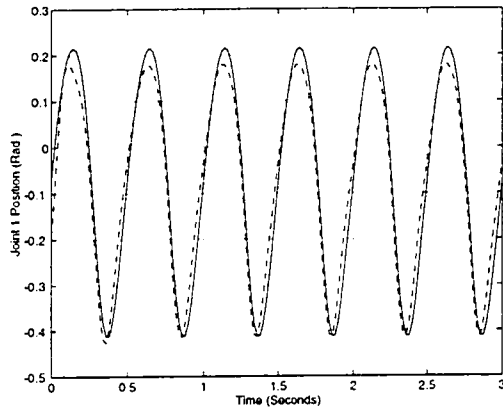


(e) Final Error (Joint 3)

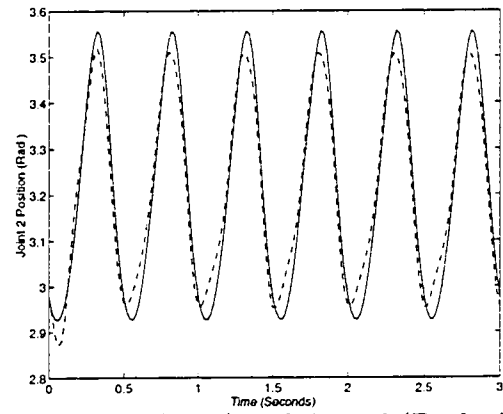


(f) Final Error (Joint 4)

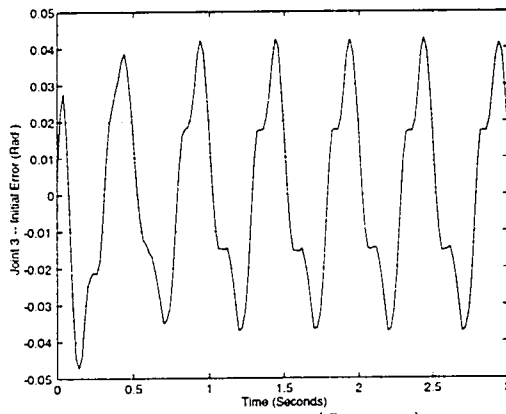
Figure 5.5: The Response of the R.P.D.R. When The Initiation Frequency was Set at 1 KHz. ($T_i = 1$ millisecond).



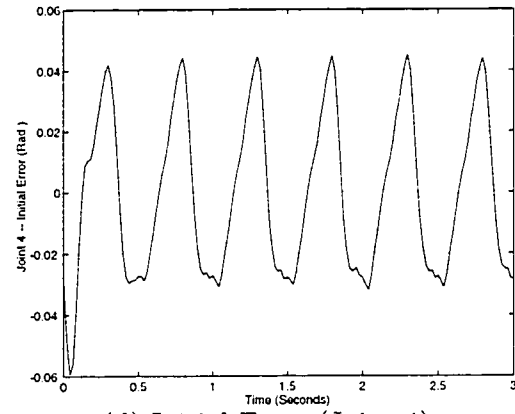
(a) Desired (solid) and Actual (Dashed) Trajectory Profile of Joint 1



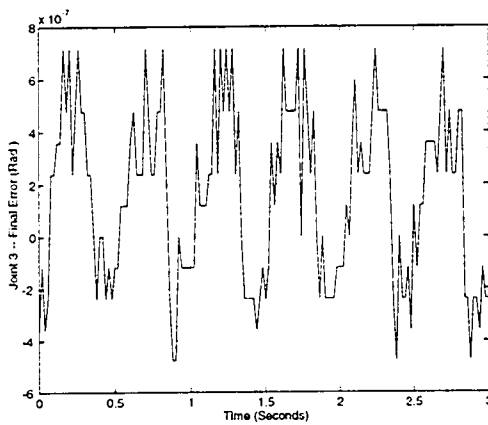
(b) Desired (solid) and Actual (Dashed) Trajectory Profile of Joint 2



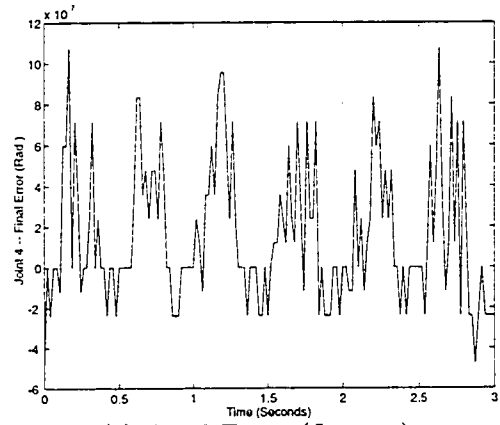
(c) Initial Error (Joint 3)



(d) Initial Error (Joint 4)

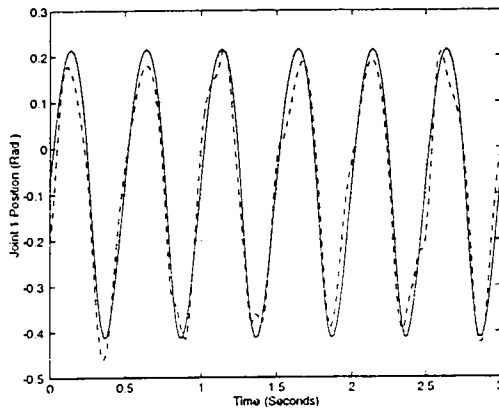


(e) Final Error (Joint 3)

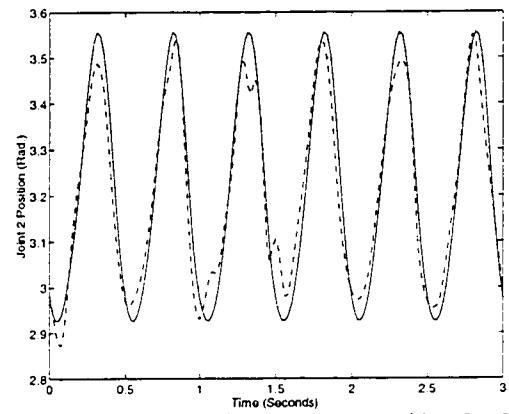


(f) Final Error (Joint 4)

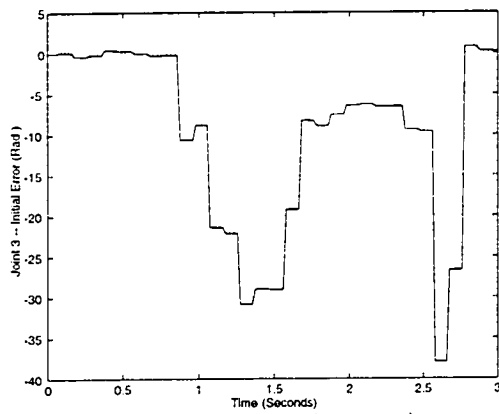
Figure 5.6: The Response of the R.P.D.R. When the Initiation Frequency was Set at 100 Hz. ($T_i = 10$ millisecond).



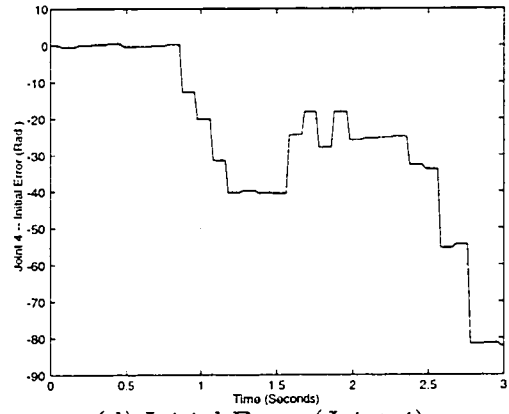
(a) Desired (solid) and Actual (Dashed) Trajectory Profile of Joint 1



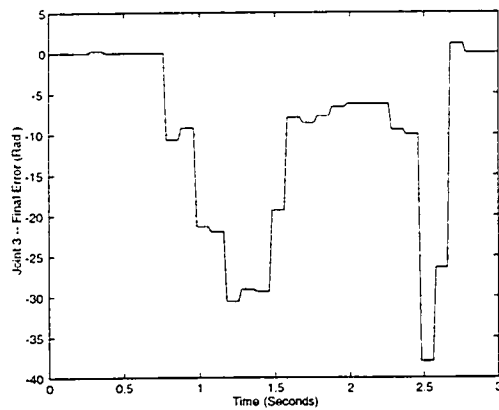
(b) Desired (solid) and Actual (Dashed) Trajectory Profile of Joint 2



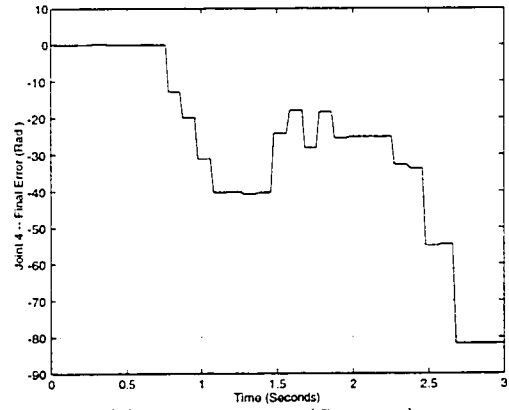
(c) Initial Error (Joint 3)



(d) Initial Error (Joint 4)



(e) Final Error (Joint 3)



(f) Final Error (Joint 4)

Figure 5.7: The Response of the R.P.D.R. When the Initiation Frequency was Set at 10 Hz. ($T_i = 100$ millisecond).

implementing control in parallel robots and implemented it in a trajectory tracking experiment with a planar delta robot. We found the experimental results to be consistent with the results of this chapter.

Chapter 6

Conclusions and Future Recommendations

In this chapter we summarize the contributions of this thesis and discuss future extensions to this research. The work done in this project can be broadly categorized into two areas. The contributions of Chapter 2 and Chapter 3 fall in to the category of serial robot control. The issues addressed in Chapter 4 and Chapter 5 are in the area of parallel robot control.

6.1 Control of Serial Robots

In the area of serial robots we addressed the uniform boundedness issue. In past research the class of robots for which the inertia matrix, $D(\mathbf{q})$ appearing in (1.1) is uniformly bounded was characterized and named Class \mathcal{BD} . Our objective was to characterize corresponding classes of robots for which the other terms arising in (1.1) are uniformly bounded. We concentrated on the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix in Chapter 2 and found that the class of robots for which the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is uniformly bounded is the same as Class \mathcal{BD} . The class of robots for which the Hessian is uniformly bounded was characterized in Chapter 3 and named Class \mathcal{BGJ} . We found that Class \mathcal{BD} is a subset of Class \mathcal{BGJ} and hence, all of the terms in question are uniformly bounded for Class \mathcal{BD} robots. This is an important discovery since almost all of the known commercial serial robots belong to Class \mathcal{BD} . Since the uniform boundedness has important implications in control and the conditions for belonging to Class \mathcal{BD} are not hard to satisfy, this issue should be taken into consideration when designing robots.

The next contribution is the derivation of explicit expressions for the uniform bounds β and γ satisfying (1.2)-(1.3). Since the uniform bounds frequently appear in the expressions of control laws, in the implementation of control the explicit values of the uniform bounds are often necessary to compute the control law expression. In Chapter 2 we derived an explicit expression for the uniform bound γ in terms of kinematic link parameters. In Chapter 3 we first derived an explicit expressions for the uniform bound β and then derived an expression for a local bound c_4 satisfying (1.4) for robots outside Class \mathcal{BGJ} . If a given robot is outside Class \mathcal{BGJ} , this expression can still be used to implement control although the stability properties will then become local.

The expressions for the uniform bounds β and γ derived in chapters 2 and 3 lends themselves well to algorithmic computation. A possible extension of this research is the development of a software package that will automatically compute these uniform bounds. Such a software package can be a stand alone program written specifically to compute the uniform bounds or it can be part of a larger software package written for control of serial robots. Such a software package can be designed to perform many useful functions in control synthesis such as computing all relevant parameters, suggesting an appropriate control law based on the application, computing trajectories, and performing simulations.

6.2 Control of Parallel Robots

In the area of parallel robots we addressed several issues related to the implementation of the dynamics formulation for parallel robots derived in [24]. Our first objective was to characterize an explicit domain where the parameterization (1.9) exists. This issue is important since the equations of motion for parallel robots presented in [24] is valid only if the parameterization (1.9) exists. We addressed this issue in Chapter 5

and characterized an explicit set where (1.9) exists. In addition we also characterized a region of the workspace, namely the set \mathcal{D}' , where the parameterization (1.9) exists along any trajectory contained within \mathcal{D}' . This region is sufficiently large to allow the trajectory of the robot in a typical application to remain entirely within it. This result is important in control synthesis.

The next contribution was the design and construction of the Rice Planar Delta Robot (R.P.D.R.). This robot was specially designed to act as a test bed for results on control of parallel robot. The versatile design allows for modifying the system to change the dynamics parameters of the robot. Due to the simplicity of the R.P.D.R. it was possible to compute the equations of motion quite accurately. The close agreement between the simulation and the experimental results reported in Chapter 4 indicates the accuracy of the equations of motion. The R.P.D.R. was used extensively in this project to experimentally verify the results. It will be useful in future experiments on parallel robots also.

Another important contribution of this research is the experimental validation of the PD plus simple gravity control law proposed for parallel robots in [22]. This control law is ideal for set point tracking problems since the compensation term in the control law is constant and hence, it can be computed off line which leaves very little real time computation. A common application of parallel robots is in fast pick-and-place operations. In Chapter 4 we implemented a fast pick-and-place motion in the R.P.D.R. and compared experimental results with simulations. We also discussed issues related to implementing the PD plus simple gravity control law for pick-and-place applications.

For trajectory tracking applications, most of the available control laws require the computation of the dynamics in real time. In order implement such control laws, the parameterization (1.9) must be computed in real time. Since analytical expressions

for (1.9) doesn't exist in general, (1.9) must be computed numerically. A numerical scheme to compute (1.9) in real time can only be implemented if convergence can be guaranteed. In Chapter 5 we addressed this issue and derived conditions that ensure the convergence of a numerical scheme proposed to compute (1.9). Based on this result we proposed a procedure for implementing control laws that require the computation of (1.9) in real time. Based on this result we implemented a trajectory tracking experiment in the R.P.D.R. The experimental results validated the theoretical predictions.

In this area an issue that remains unresolved is the characterization of the domain of attraction of control laws. Note that since the equations of motion themselves are not global for parallel robots stability properties of control laws are never valid globally. The characterization of the domain of attraction of control laws that are valid locally is a difficult problem even for serial robots. For parallel robots this problem is even more complicated since the valid domain will be a subset of a set where the parameterization (1.9) exists. The closed ball of radius r , characterized in Chapter 5, where the parameterization (1.9) is valid is physically too small to implement control. Hence, a useful extension to this work will be the characterization of a larger set where (1.9) exists. Other useful extensions of this research will be the implementation of the results reported in this thesis in more complicated parallel robots such as the six degree of freedom hexa robot.

6.3 Summary

In this chapter we briefly summarized the contributions of this thesis and proposed extensions. The following are the main contributions of this thesis:

- 6.1. Characterization of the class of robots for which the $C(\mathbf{q}, \dot{\mathbf{q}})$ matrix is uniformly bounded and the derivation of an explicit expression for a uniform bound γ satisfying (1.3).
- 6.2. Characterization of the class of robots for which the Hessian is uniformly bounded and the derivation of an explicit expression for a uniform bound β satisfying (1.2).
- 6.3. Design and Construction of the Rice Planar Delta Robot which will serve as a test bed for experiments on parallel robot control.
- 6.4. Experimental verification of the PD plus simple gravity control law for parallel robots and the evaluation of this control law for fast pick-and-place applications.
- 6.5. Characterization of an explicit set where the parameterization (1.9) exists.
- 6.6. The proposal of a numerical scheme for computing (1.9) in real time for model based control laws for trajectory tracking and the derivation of conditions for convergence. Proposal of a procedure for implementing control based on this result and the experimental verification.

Bibliography

- [1] Ailon, A., and Ortega, R., "An observer-based set-point controller for robot manipulators with flexible joints", *Systems & Control Letters*, Vol. 21, pp. 329-335, 1993.
- [2] Amirat, Y., Francois, C., Fried, G., Pontnau, J., and Dafaoui, M., "Design and Control of a New Six DOF Parallel Robot: Application to Equestrian Gait Simulation," *Mechatronics*, Vol. 6, No. 2, pp. 227-239, 1996.
- [3] Arimoto, S., and Miyazaki, F., "Stability and Robustness of PID Feedback Control of Robot Manipulators of Sensory Capability," *1st Int. Symp. on Robotics Research*, M. Brady and R.P. Paul (Eds.), pp. 783-799, MIT Press, Boston, MA., 1984.
- [4] Armstrong, B., Khatib, O., and Burdick, J., "The Explicit Dynamic Model and Inertial Parameters of the PUMA560 Arm" in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 1, (San Francisco, USA), pp. 510-518, 1986.
- [5] Babaci, S., Amirat, Y., Pontnau, J., and Francois, C., "Fuzzy Adaptation Impedance of a 6 D.O.F. Parallel Robot. Application To Peg In Hole Insertion" *Proc. of the fifth IEEE International Conference on Fuzzy Systems*, pp. 1770-1776, New Orleans, Louisiana, USA, September, 1996.
- [6] Bartle, Robert G., *The Elements of Real Analysis*, John Wiley & Sons, Inc., 1976.

- [7] Bégon, P., Pierrot, F., and Dauchez, P., "Fuzzy Sliding Mode Control of a Fast Parallel Robot", *Proc. IEEE International Conference on Robotics and Automation*, pp. 1178-1183, 1995.
- [8] Berghuis, H., Ortega, R., and Nijmeijer, H., "A Robust Adaptive Robot Controller", *IEEE Transaction on Robotics and Automation*, Vol. 9, No. 6, pp. 825-830, December 1993.
- [9] Berghuis, H., and Nijmeijer, H., "A Passivity Approach to Controller-Observer Design for Robots", *IEEE Transaction on Robotics and Automation*, Vol. 9, No. 6, pp. 740-754, December 1993.
- [10] Chételat, O., Merlet, J.-P., Myszkorowski, and Longchamp, R., "Globally Convergent Iterative Algorithm for Coordinate Transformations in Articulated Mechanisms", *Proceedings of the Fifth IFAC Symposium on Robot Control SY-ROCO'97*, Nantes, France, pp.743-749, Sept. 3-5, 1997.
- [11] Chételat, O., Myszkorowski, P., and Longchamp, R., "Diffeomorphisme global pour la cinématique directe", Technical Report IA-94-4, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
- [12] Chiacchio, P., Pierrot, F., Sciavicco, L., and Siciliano, B., "Robust Design of Independent Joint Controllers with Experimentation on a High-Speed Parallel Robot", *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 4, pp. 393-403, August, 1993.
- [13] Craig, J. J., *Adaptive Control of Mechanical Manipulators*, Addison-Wesley Publishing Company, 1988.
- [14] Craig, J. J., *Introduction to Robotics*, Addison-Wesley Publishing Company, 1989.

- [15] De Luca, A. and Panzieri, S., "Learning Gravity Compensation in Robots: Rigid Arms, Elastic Joints, Flexible Links", *International Journal of Adaptive Control and Signal Processing*, Vol. 7, pp. 417-433, 1993.
- [16] dSPACE digital signal processing and control engineering GmbH, Technologiepark 25, D-33100 Paderborn, Germany, 1993.
- [17] Fattah, A., Angeles, J., and Misra, A. K., "Dynamics of a 3-DOF Spatial Parallel Manipulator with Flexible Links", *Proc. IEEE International Conference on Robotics and Automation*, pp. 627-632, 1995.
- [18] Fu, L., "Robust Adaptive Decentralized Control of Robot Manipulators", *IEEE Transaction on Automatic Control*, Vol. 37, No. 1, pp. 106-110, 1992.
- [19] Gautier, M., Khalil, P., and Restrepo, P., "Identification of the Dynamic Parameters of a Closed Loop Robot" *Proc. IEEE International Conference on Robotics and Automation*, pp. 3045-3050, 1995.
- [20] Geng, Z., and Haynes, L.S., "Dynamic Control of a Parallel Link Manipulator Using a CMAC Neural Network", *Computers Elect. Engng*, Vol 19, No. 4, pp. 265-276, 1993.
- [21] Geng, Z., Haynes, L.S., Lee, J.D., and Carrol, R.L., "On the Dynamic Model and Kinematic Analysis of a Class of Stewart Platforms", *Robotics and Autonomous Systems*, Vol. 9, pp. 237-254, 1992.
- [22] Ghorbel, F., "Modeling and PD Control of Closed-Chain Mechanical Systems", *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, Louisiana, December 13-15, 1995.

- [23] Ghorbel, F., Chételet, O., Gunawardana, R., and Longchamp, R. "Modeling and Set Point Control of Closed-Chain Mechanisms: Theory and Experiment," *IEEE Transactions on Control Systems Technology*, (Submitted)
- [24] Ghorbel, F., Chételet, O., and Longchamp, R., "A Reduced Model of Constrained Rigid Bodies with Application to Parallel Robots", *Proceedings of the 4th IFAC Symposium on Robot Control*, Capri, Italy, September 19-21, 1994.
- [25] Ghorbel, F., and Gunawardana, R., "On the Uniform Boundedness of the Derivative of the Gravitational Force of Serial Link Robot Manipulators", *Proc. of the ASME International Mechanical Engineering Congress and Exposition (IMECE)*, San Francisco, California, November 12-17, 1995.
- [26] Ghorbel, F., and Gunawardana, R., "A Uniform Bound for the Jacobian of the Gravitational Force Vector for a Class of Robot Manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 119, No. 1, pp. 110-114, 1997.
- [27] Ghorbel F., and Spong M.W., "Integral Manifold Corrective Control of Flexible Joint Robot Manipulators: The Known Parameter Case," *DSC-Vol. 31, Modeling and Control of Compliant and Rigid Motion Systems*, pp. 81-93, ASME Winter Annual Meeting, Atlanta, Georgia, December 1-6, 1991.
- [28] Ghorbel F., and Spong M.W., "Adaptive Integral Manifold Control of Flexible Joint Robots with Configuration Invariant Inertia" *Proc. of the American Control Conference*, pp. 3314-3318, Chicago, Illinois, June 24-26, 1992.
- [29] Ghorbel F., and Spong M.W., "Adaptive Integral Manifold Corrective Control of Flexible Joint Robot manipulators" *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 707-714, Nice, France, May 10-15, 1992.

- [30] Ghorbel F., and Spong M.W., "Robustness of Adaptive Control of Robots," *Journal of Intelligent Robotic Systems*, Vol. 6, pp. 3-15, 1992.
- [31] Ghorbel F., Srinivasan B., and Spong M.W., "On the Positive Definiteness and Uniform Boundedness of the Inertia Matrix of Robot Manipulators", *Proc. of the 32nd IEEE Conf. on Decision and Control*, pp. 1103-1108, San Antonio, Texas, December, 1993.
- [32] Gosselin, C. M., "Parallel Computational Algorithms for the Kinematics and Dynamics of Parallel Manipulators", *Proc. of IEEE International Conference on Robotics and Automation*, pp. 883-888, 1993.
- [33] Greenwood, D.T., *Classical Dynamics*, Prentice-Hall, Englewood Cliffs, N.J., 1977.
- [34] Gunawardana, R., and Ghorbel, F., "Control of Serial and Parallel Robots: Analysis and Implementation", *Internal Report*, Department of Mechanical Engineering, Rice University, 1998.
- [35] Gunawardana, R., and Ghorbel, F., "On the Boundedness of the Hessian of the Potential Energy of Robot Manipulators", *Submitted for publication in the IEEE Transactions on Robotics and Automation*.
- [36] Gunawardana, R., and Ghorbel, F., "The Class of Robot Manipulators with Bounded Jacobian of the Gravity Vector", *Proc. International Conference on Robotics and Automation (ICRA)*, pp. 3677-3682, April 1996.
- [37] Gunawardana, R., and Ghorbel, F., "On the Uniform Boundedness of the Coriolis/Centrifugal Terms in the Robot Equations of Motion", *Submitted for publication in the International Journal of Robotics & Automation*.

- [38] Gunawardana, R., and Ghorbel, F., "PD Control of Closed-Chain Mechanical Systems: An Experimental Study", *Proceedings of the Fifth IFAC Symposium on Robot Control SYROCO'97*, Nantes, France, Sept. 3-5, 1997 (to appear).
- [39] Hudgens, J.C., and Tesar, D., "A Fully Parallel Six Degree-of-Freedom Micromanipulator: Kinematic Analysis and Dynamic Model" *Proc. of ASME Des Eng. Div. Publ. DE Trends and Developments in Mechanisms Machines, and Robots*, Kissimmee, FL, USA, 1988 V 15-3, pp. 29-37, 1988.
- [40] Hutson, V., and Pym, J.S., *Applications of Functional Analysis and Operator Theory*, Academic Press, New York, 1980.
- [41] Ji, Z., "Study of the effect of Leg Inertia in Stewart Platforms", *Proc. of IEEE International Conference on Robotics and Automation*, pp. 121-126, 1993.
- [42] Kang, H. J., and Freeman, R. A., "An Interactive Software Package (MAP) for the Dynamic Modeling and Simulation of Parallel Robotic Systems Including Redundancy", *Proc. of ASME International Computers in Engineering Conference and Exposition*, Boston, MA, pp. 117-122, 1990
- [43] Kelly, R., Ortega, R., Ailon, A., and Loria, A., "Global Regulation of Flexible Joint Robots Using Approximate Differentiation", *IEEE Transaction on Automatic Control*, Vol. 39, No. 6, pp. 1222-1224, 1994.
- [44] Kelly, R., and Salgado, R., "PD Control with Computed Feedforward of Robot Manipulators: A Design Procedure", *IEEE Trans. on Robotics and Automation*, Vol. 10, No. 4, pp. 566-571, August 1994.
- [45] Kleinfinger, J.F., and Khalil, W., "Dynamic Modeling of Closed-Loop Robots" *Proceedings of the 16th International Symposium on Industrial Robots*, pp. 401-412, Brussels, Belgium, 1986.

- [46] Konishi, Y., Aoyama, T., and Inasaki, I., "Learning Control of a Parallel-Link Direct-Drive Robot Manipulator" *Robotics and Autonomous Systems*, pp. 127-134, 1989.
- [47] Kreyszig, E., *Advanced Engineering Mathematics*, Wiley Eastern Limited, New Delhi, 1972.
- [48] Lee, J.D., "Optimal Control of a Flexible Parallel Link Robotic Manipulator" *Computers and Structures*, Vol. 48, No. 3, pp. 375-385, 1993.
- [49] Lee, J. D.. and Geng, Z., "A Dynamic Model of a Flexible Stewart Platform" *Computers & Structures*, Vol. 48, No.3, pp. 367-374, 1993.
- [50] Lee, K., and Dharman. K. S., "Dynamic Analysis of a Three-Degrees-of-Freedom In-Parallel Actuated Manipulator", *Journal of Robotics and Automation*, Vol. 4, No. 3, pp. 361-367, 1988.
- [51] Lewis, F.L., Abdallah, C.T., and Dawson, D.M., *Control of Robot Manipulators*, Macmillan Publishing Company, New York, 1993
- [52] Lin, S., " Dynamics of the Manipulator with Closed Chains" , *IEEE Transactions on Robotics and Automation*. Vol. 6, No. 4, pp. 496-501, 1990.
- [53] Luh, J. Y. S., and Zeng, Y., "Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms", *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 2, pp. 95-103, 1985.
- [54] The MathWorks Inc., "Matlab", 24 Prime Park Way, Natwick, Mass 01760, September, 1994.

- [55] Murray, J. J., and Lovell, H., "Dynamic Modeling of Closed-Chain Robotic Manipulators and Implications for Trajectory Control", *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 4, pp. 522-528, 1989.
- [56] Nakamura, Y., and Ghodoussi, M., "Dynamics Computation of Closed-Link Robot Mechanisms with Nonredundant and Redundant Actuators", *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, pp. 294-302, 1989.
- [57] Nicosia, S., and Tomei, P., "A Tracking Controller for Flexible Joint Robots Using Only Link Position Feedback", *IEEE Transaction on Automatic Control*, Vol. 40, No. 5, pp. 885-890, May 1995.
- [58] Nikravesh, P.E., and Gim, G., "Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops", *Proc. of the ASME Design Technical Conferences-15th Design Automation Conference*, pp 27-33, Motreal, Quebec, Canada, September 17-21, 1989
- [59] Ortega, R, Loria, A., and Kelly, R., "A Semiglobally Stable Output Feedback PI²D Regulator for Robot Manipulators", *IEEE Transactions on Automatic Control*, Vol. 40, No. 8 pp. 1432-1436, 1995.
- [60] Ortega, R, and Spong, M.W., "Adaptive Motion Control of Rigid Robot Manipulators: a Tutorial", *Proceedings 27th IEEE Conference on Decision and Control*, Austin, Texas, pp. 1575-1584, December, 1988.
- [61] Pooran, F.J., "Dynamics and Control of Robot Manipulators with Closed-Kinematic Chain Mechanism", Ph.D. Dessertation, The Catholic University of America, 1989.

- [62] Qu, Z., "Input-Output Robust Tracking Control Design for Flexible Joint Robots", *IEEE Transaction on Automatic Control*, Vol. 40, No. 1, pp. 78–83, January 1995.
- [63] Qu, Z., and Dorsey, J., "Robust Tracking Control of Robots by a Linear Feedback Law", *IEEE Transaction on Automatic Control*, Vol. 36, No. 9, pp. 1081–1084, September 1991.
- [64] Reboulet, C., and Pigeyre, R., "Hybrid Control of a 6-DOF In-Parallel Actuated Micro-Manipulator Mounted on a Scara Robot", *International Journal of Robotics and Automation*, Vol. 7, No. 1, pp. 10-14, 1992.
- [65] Sadegh, N., and Horowitz, R., "Stability and Robustness of a Class of Adaptive Controllers for Robotic Manipulators", *The International Journal of Robotics Research*, Vol. 9, No. 3, pp. 74–92, 1990.
- [66] Spong, M.W., and Vidyasagar, M., *Robot Dynamics and Control*, John Wiley & Sons, Inc., 1989.
- [67] Su, C., and Stepanenko, Y. "Adaptive Control for Constrained Robots without Using Regressor", *Proc. International Conference on Robotics and Automation (ICRA)*, pp. 264-269, April 1996.
- [68] Takegaki, M., and Arimoto, S., "A New Feedback Method for Dynamic Control of Manipulators", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, pp. 119–125, 1981.
- [69] Taylor, A.E., Mann, W.R., *Advanced Calculus*, John Wiley & Sons, Inc., 1983.
- [70] Tomei, P., "Adaptive PD Controller for Robot Manipulators", *IEEE Transaction on Robotics and Automation*, Vol. 7, No. 4, pp. 565–570, August 1991.

- [71] Tomei, P., "Tracking Control of Robots with Uncertain Parameters and Disturbances", *IEEE Transaction on Automatic Control*, Vol. 39, No. 5, pp. 1067-1072, May 1994.
- [72] Unda, J., de Jalón, J.G., Losantos, F., and Euparantza, R., "A Comparative Study on Some Different Formulations of the Dynamic Equations of Constrained Mechanical Systems", *Transaction of the ASME, J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 109, pp. 466-474, December 1987.
- [73] Vidyasagar, M., *Nonlinear Systems Analysis*, Prentice Hall, 1993
- [74] Walker, M.W., "Adaptive Control of Manipulators Containing Closed Kinematic Loops" *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 1, pp.10-19, 1990.
- [75] Wang, L. T., and Chen, C. C., "On the Dynamic Analysis of General Parallel Robotic Manipulators", *International Journal of Robotics and Automation*, Vol. 9, No. 2, pp. 81-87, 1994.
- [76] Wang, L. T., and Chen, C. C., "On the Numerical Kinematic Analysis of General Parallel Robotic Manipulators", *IEEE Transactions on Robotica and Automation*, Vol. 9, No. 3., pp. 272-85 1993.
- [77] Wehage, R.A., and Haug, E.J., "Generalized coordinate Partitioning for Dimensional Reduction in Analysis of Constrained Dynamic Systems" , *Journal of Mechanical Design*, Vol 104, pp.247-255, 1982
- [78] Wen, J. T., "An Unified Perspective on Robot Control: The Energy Function Approach", *International Journal of Adaptive Control and Signal Processing*, Vol. 4, No. 6, pp. 487-500, 1990.

Appendix A

The Design of the Rice Planar Delta Robot

A.1 Introduction

In this appendix we describe the design process of the Rice Planar Delta Robot (R. P. D. R.). We will begin with a general discussion of our design objectives. The design process was based on these objectives. We will then describe the design of each of the components of the robot in the remaining sections.

Our primary objective was to design and build a parallel robot which can be used to implement control and test results in the area of parallel robots. In a typical application a parallel robot is made to follow some desired motion. In order to achieve the desired motion the actuators should be able to develop the necessary torque/force profile based on the control law being used. Therefore, in order to conduct experiments we needed a robot in which the actuator forces can be programmed. Furthermore, since parallel robots are capable of very fast motion we needed actuators and a programming system with a fast dynamic response. We chose a computer with a DSP board to achieve this objective. The DSP board allows us to have a processor dedicated to compute the control law. This enables the implementation of complicated control laws at a very high sampling rate.

The fact that our primary objective was academic research imposed requirements that were different from typical objectives of designing a robot. These include ease of modeling, easy access to model parameters, and the ability to vary the model parameters. Even though in a typical parallel robot it's desired to keep singular configura-

tions outside the workspace, in our application we wanted the robot to be capable of moving through singular configurations in order to study them. Furthermore, since we wanted to isolate and study the features of parallel robots, it was desirable to reduce unmodeled effects such as friction, backlash, and link and joint flexibility. These requirements and additional safety requirements led to the following set of design objectives:

- 1.1. The actuators should be able to provide sufficient power to be able to achieve high speed.
- 1.2. The control system should be capable of computing a complicated control law at a very high sampling rate (in the KHz. range).
- 1.3. The robot should have all of the sensors necessary to implement control.
- 1.4. The robot should be easy to model and we should have access to as many model parameters as possible.
- 1.5. It should be possible to vary model parameters of the robot such as link length, masses etc.
- 1.6. The robot should be able to have a complete range of motion including singular configurations.
- 1.7. Unmodeled effects such as friction, backlash, joint and link flexibilities should be minimized (flexibilities and backlash effects should be minimized sufficiently to make their effects negligible).
- 1.8. The robot should be safe to the operator as well as the surrounding environment.
- 1.9. A target end-effector error of 0.1 mm. or less.

The design process was guided by these objectives. The complete robot system (the R. P. D. R.) consists of the following three components:

- 1.1. The robot and the base.
- 1.2. The amplifier circuit unit and power supply.
- 1.3. The computer, the dSpace board and software for both.

The robot and base consists of the links, the actuators and sensors, passive joints, and the base. We will describe the design of the robot and base in Section A.2. This is followed by Section A.3 where we describe the amplifier circuit unit which consists of all of the interfacing circuitry between the computer and the robot. Finally in Section A.4 we will discuss the computer and DSP board (dSPACE) which is used to control the robot.

A.2 The Robot and the Base

The robot consists of four links connected by three passive joints terminated in the two ends by actuators (See Figure A.1). The actuators are mounted on a solid base. The robot work area is enclosed by a protective plexiglass box. We will next describe the design of each of these components.

The Base: In designing the base several factors were considered. Firstly, we needed a solid base which did not vibrate due to the motion of the robot in order to achieve Objective 1.7. This was achieved by designing the base and attaching the plexiglass cover (oriented along the direction of motion) to provide high stiffness at the location where the actuators are mounted (See Figure A.2). The legs were made out of steel tubing and the top surface out of steel channel. Secondly, in order to achieve objective 1.6 the robot had to be mounted such that each link had a completely unobstructed vertical plane. This was achieved by mounting the actuators on opposite

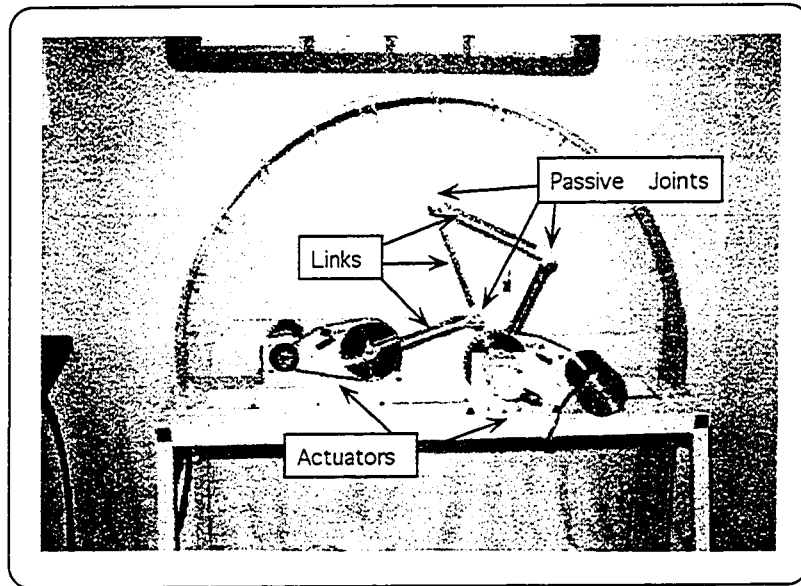


Figure A.1: Joint and Link Configuration of the Rice Planar Delta Robot

sides of the base and having the links in between the actuators (See Figure A.1). This allows a full circle range of motion for all of the joints. Thirdly, in order to achieve Objective 1.5, we needed to allow for the variation of the one model parameter that is related to the base, which is the distance between the axes of the actuated joints. This was achieved by designing a series of mounting holes for the actuator units. Finally, in order to achieve Objective 1.8, we estimated the maximum possible work space and designed a plexiglass enclosure for it.

Actuator Unit: Each actuator unit consists of a motor and transmission system, an optical encoder, and a brake (See Figure A.3). We chose DC motors because they are easy to model and simple to control. Since a DC motor delivers low torque at high speed, it needs a transmission that reduces speed and increases torque. We decided against using a gear head since it adds backlash. In order to achieve Objective 1.7 we felt that the best choice was to use a sprocket and timing chain system. The

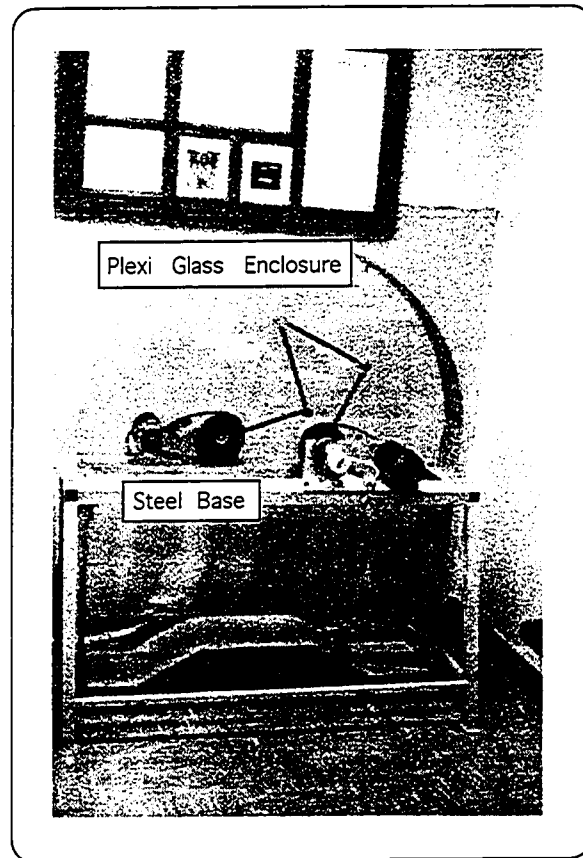


Figure A.2: The Rice Planar Delta Robot-Full View

timing chain made of steel wires is especially designed to virtually eliminate joint flexibilities and backlash (See Figure A.4). We chose a sprocket with 38 teeth for the joint end and a sprocket with 9 teeth for the motor end. This results in an increase of torque by a factor of 4.22 from the motor to the joint. We also added a belt tensioner pulley. In order to select the motor, we conducted some simulations to estimate the required maximum torque. The general approach was to estimate the torque required to achieve the desired range of velocities and accelerations in a wide range of robot configurations and then multiply this by a design factor to account for unmodeled effects such as friction. Based on these we had an estimate of the required torque

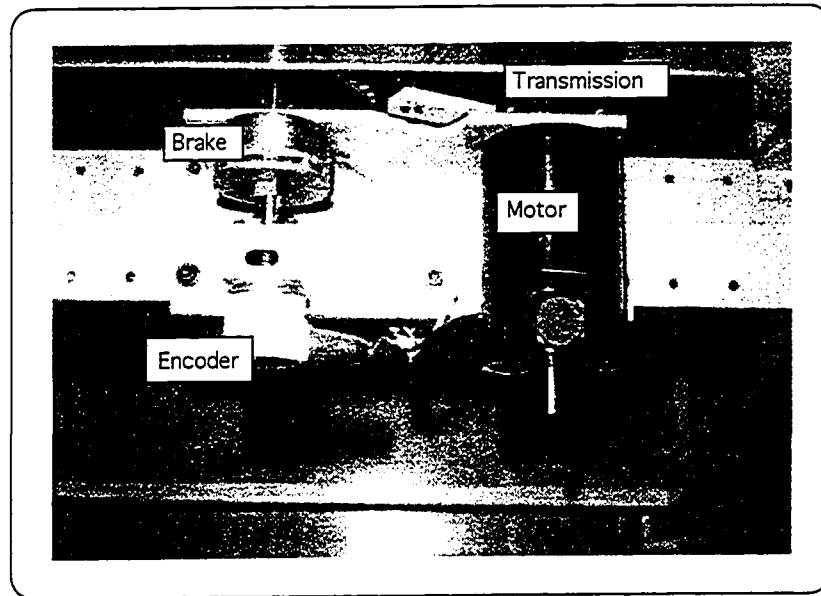


Figure A.3: The Actuator Unit

output from the motor. After considering several products we chose a 1/4 hp Bodine 4435 motor with an output torque of 100 Oz-in for continuous operation at 2500 r.p.m. The motors are capable of delivering much higher torque at lower speeds for short periods of time. When designing the actuator system special attention was paid to the alignment of centerlines and to making all rolling contacts through bearings to reduce friction.

The sensors used for control in the RPDR are optical encoders. These provide position feed back. The velocity feedback is obtained by differentiating the position and filtering out noise. The selection of the encoder was based on the error target in Objective 1.9. The worst case for error at the endeffector corresponding to the motion of a joint is when the links corresponding to that joint are completely stretched out. In this case, based on the dimensions of the links, the endeffector is 23 inches from

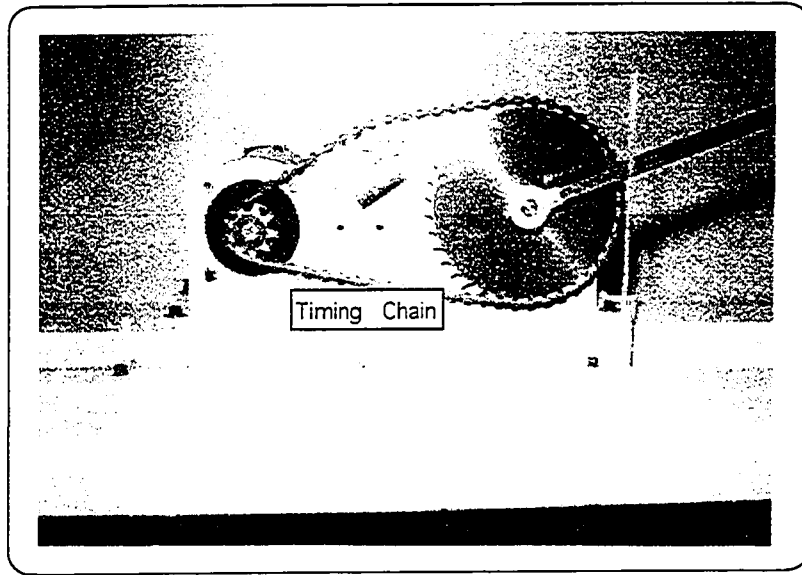


Figure A.4: The Transmission

the joint. Hence, we can only afford an error, q_{error} , given by

$$q_{error} = \tan^{-1} \left(\frac{0.1\text{mm}}{23\text{in}} \right) = \tan^{-1} (0.000171) = 0.000171 \text{ rad.}$$

This corresponds to a resolution of approximately $\frac{0.000171}{2\pi} > \frac{1}{40000}$ of a circle. Since the resolution of a relative encoder with two channels (A and B) can be electronically increased by a factor of four, an encoder with 10000 pulses per revolution will be able to achieve our error target. We chose a Hengstler Optical Encoder with the required resolution. Even though the design ensured alignment of the encoder centerline and the joint centerline to a high level of accuracy, we chose to connect the encoder through a flexible coupling to protect it.

We decided to add brakes to the RPDR to hold the links in place once it reaches the desired configuration. We chose negatively actuated brakes, that is, brakes that hold the load when the power is turned off. The brakes are controlled by the computer. Therefore, when the robot needs to be moved power is applied to the brakes first to

release the robot and then after completing the motion the brakes are turned off to hold the robot. Even though we used this method in early experiments, since the brakes add friction and there are time delays between applying power and when the brakes are actually released, we chose to have the brakes released throughout in the later experiments. In this case the torques developed in the motors hold the robot in the desired configuration.

Links: The links were chosen to be light weight and sufficiently stiff to achieve Objective 1.7. We chose 0.5 inch square, hollow steel tubing as they provide a high stiffness to weight ratio. The link flexibility was computed and checked to make sure that we achieve the stiffness required for Objective 1.7. In order to achieve Objective 1.5 we wanted to allow for varying the length and weight of links. We considered designing a length adjusting mechanism and chose against it since such a mechanism would compromise our link stiffness and light weight objectives. Instead, the links were attached to the joints by a single pin which allows for easy change of the tubing. Since steel tubing is extremely inexpensive, when there is a need to vary the link lengths and weight of RPDR, a new set of links can be made easily and attached.

Passive Joints: In designing the passive joints we paid attention to reducing friction, keeping them light weight, ensuring a rigid attachment to the links, and keeping them within size limitations imposed by the need to keep the planes of motion of the links free of obstructions. Each rolling contact was made through a pair of roller bearings which ensured very low friction. The joint housing was made of an Aluminum alloy to reduce weight.

A.3 Circuit Box and Power Supply

There is a need for several interfacing circuits between the computer/dSPACE board and the sensors and actuators. We placed all of these circuits in one unit which we refer to as the “Circuit Box”. We made this unit in an old computer housing and used its power supply for logic power. The 60 V DC power required to run the motor is provided by a Hewlett Packard power supply. The actuator unit is connected to the circuit box which in turn communicates with the computer/dSPACE board (See Figure A.5). Inside the circuit box there are two circuits. The first circuit controls the

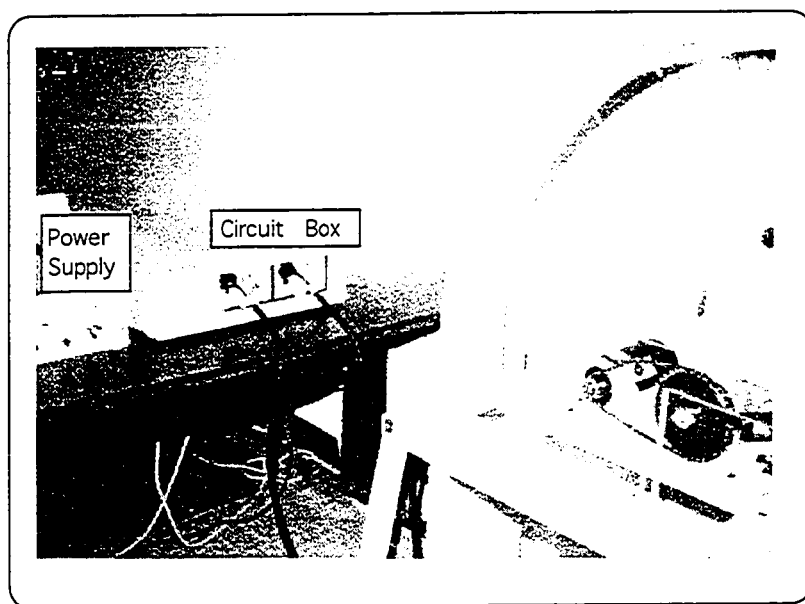


Figure A.5: The Circuit Box

motors. In order to drive the motors there is a need to have a circuit that reads the signals from the dSPACE board and develops a voltage proportional to the desired torque in the motors. While commercial amplifiers are available for this purpose, we felt that they will introduce additional unmodeled dynamics. Therefore, in order to achieve Objective 1.4 we decided to design the amplifiers ourselves. We chose H-

Bridge drivers to control the motors. These circuits allow the control of amplitude by using pulse width modulation and in addition allows direction control by switching the power to the motor in either direction (See Figure A.6). We also incorporated a safety switch that shuts power to the motors (See Figure A.7). The second circuit controls the brakes which allows the brakes to be operated by either the computer or

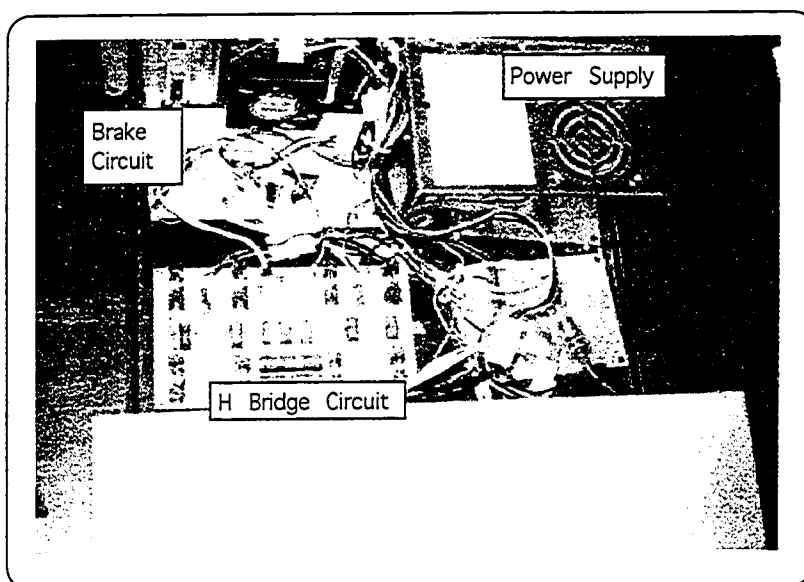


Figure A.6: The Interior View of the Circuit Box

the manual switches located in front of the circuit box. The encoders only require a 5 volt DC power which was provided by the power supply in the circuit box.

A.4 The Computer, dSPACE Board and Software

In selecting the computer and the dSPACE board, the primary objective was to compile a system that has the capability to compute complicated control laws very fast. The dSPACE boards equipped with a fast DSP is ideal for this purpose. We had further requirements such as optical encoder channels, pulse width modulated

signal outputs, and digital input/output lines to control brakes and direction control of motors. We chose the dSPACE 1102 board since it met all these requirements. This board has been able to compute all of the control laws we have implemented plus other additional calculations and data transfers for research purposes in under 500 micro seconds. This allows for a sampling rate in the KHz. range which has been sufficient for all our purposes.

We needed a computer that is capable of real time plotting and saving data efficiently. It also needs to communicate efficiently with the dSPACE board. For this reason we chose one of the fastest personal computers available at that time, a 100 MHZ Pentium computer (See Figure A.7). Real time plotting and data acquisition

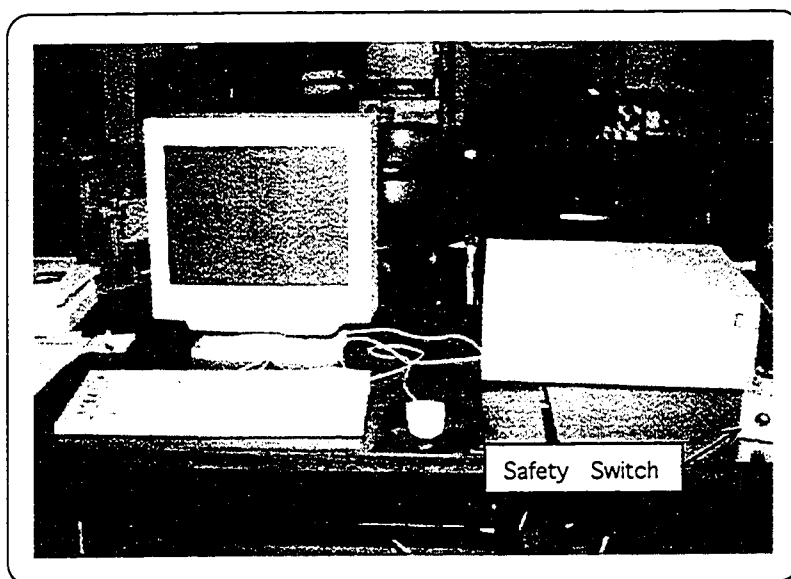
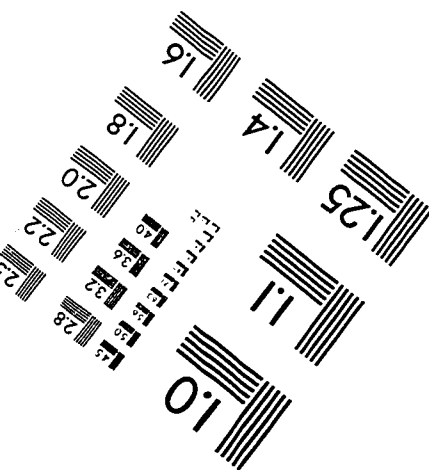
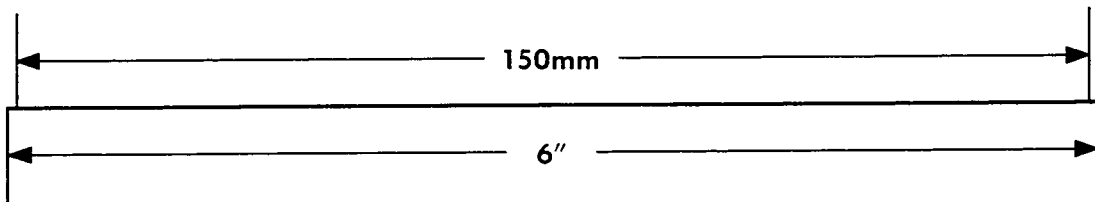
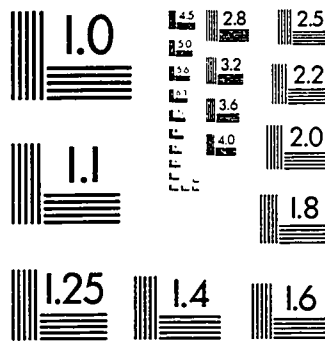
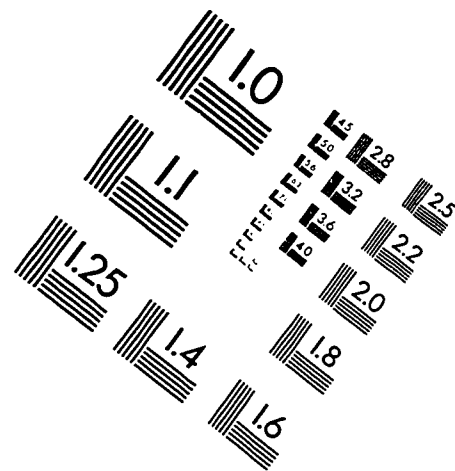
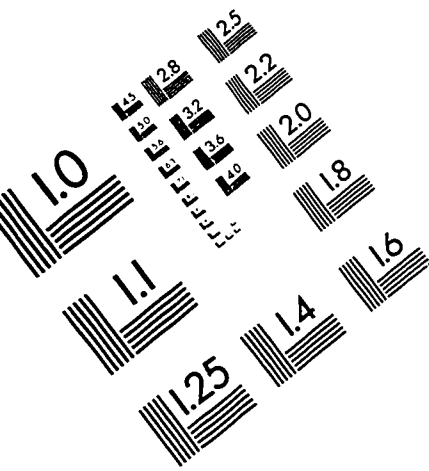


Figure A.7: The Computer

is done by software provided by the dSPACE company. In order to provide a user interface to control the robot, we developed software in C which ran in Windows. We chose a Borland C++ compiler for this purpose.

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

