

# Design and Control for a Gait Rehabilitation Robot

by

**Kyle Chisholm**

A Thesis submitted to  
the Faculty of Graduate Studies and Research  
in partial fulfilment of  
the requirements for the degree of  
**Master of Applied Science**

Ottawa-Carleton Institute for  
Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering  
Carleton University  
Ottawa, Ontario, Canada  
September 2010

Copyright ©  
2010 - Kyle Chisholm



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-71524-6  
*Our file* *Notre référence*  
ISBN: 978-0-494-71524-6

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

---

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# Abstract

For those suffering from neurological disorders or injuries, regaining lost motor function is an important component of the rehabilitation process. Instead of using compensatory methods such as walking aids or wheelchairs, modern approaches to gait rehabilitation focus on training the patient to relearn how to walk and robotics are being introduced to enhance this therapy. The potential of new approaches to gait rehabilitation motivated the design of a novel robot called the Virtual Gait Rehabilitation Robot (ViGRR) to be developed at the Advanced Biomechanics Laboratory (ABL) at Carleton University. This thesis is a presentation of the design and implementation of a scaled prototype robot as a test platform for ViGRR.

The design process of the ViGRR prototype was initiated with the formulation of the robot configuration and controller concept. The kinematic and dynamic models of the robot and a human leg were then derived, where the leg model and average gait trajectories were used to determine the requirements for the device. Through an iterative detailed design procedure, the robot design parameters were optimized and the drive units and control apparatus were selected and implemented. A robot simulation was created and the complete system was realized with hard real-time control software written for performing trajectory tracking and force control. Kinematic and dynamic calibrations and tracking experiments were carried out and the results demonstrated good performance using a model-based nonlinear robot controller.

A tracking task involving the user manipulating the robot in force control mode

and visual feedback demonstrated the system's ability to implement a control scheme analogous to the ViGRR rehabilitation-focused control concept for the full scale device. The system realization for the ViGRR prototype presented in this thesis provides a platform for research into new control strategies for gait rehabilitation.

To my Mom and Dad.

The best human beings I have ever met.

# Acknowledgments

I am exceptionally grateful to my supervisor, Mojtaba Ahmadi, who gave me support through his guidance, knowledge, experience, and abundant kindness. I'd like to thank Adam Mullins who was integral to the development of the ViGRR robot, my colleagues at the ABL lab for their support, and Clifton Pereira for awesome coffee breaks. I'd also like to acknowledge Ali Morbi for his friendship, technical support and love of Justice.

# Table of Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>Table of Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.1.1 Principles of Motor Learning and Gait Rehabilitation . . . . .	2
1.1.2 The Role of Robotics in Rehabilitation . . . . .	4
1.1.3 Approaches to Robotics-Assisted Gait Rehabilitation . . . . .	7
1.2 ViGRR: A Novel Rehabilitation Robot . . . . .	19
1.2.1 Controller Concept for ViGRR . . . . .	20
1.2.2 Prototype Concept . . . . .	23
1.3 Contributions . . . . .	28
1.4 Outline . . . . .	30
<b>2 Problem Formulation and Modelling</b>	<b>32</b>
2.1 Euler-Lagrange Formulation of Robot Dynamics . . . . .	32

2.2	Leg Model . . . . .	36
2.2.1	Leg Model Definitions and Dynamics . . . . .	37
2.2.2	Leg Kinematics . . . . .	42
2.2.3	Passive Joint Torques . . . . .	45
2.2.4	Anthropometry . . . . .	46
2.2.5	Average Gait Trajectories . . . . .	48
2.2.6	Leg Model Verification and Validation . . . . .	55
2.3	Robot Model . . . . .	60
2.3.1	ViGRR Dynamics . . . . .	60
2.3.2	Robot Kinematics . . . . .	64
2.3.3	Redundancy Resolution . . . . .	67
2.3.4	Drive Train Dynamics . . . . .	71
2.3.5	Tool Transformations . . . . .	73
2.3.6	Leg-Robot Interaction Transformations . . . . .	73
2.3.7	Robot Model Verification . . . . .	76
2.4	Summary . . . . .	77
<b>3</b>	<b>Robot Design and Apparatus</b>	<b>78</b>
3.1	Requirements . . . . .	78
3.1.1	Workspace . . . . .	78
3.1.2	End Effector Loading Requirements . . . . .	84
3.1.3	Visualizing the Workspace . . . . .	85
3.1.4	Safety Requirements . . . . .	88
3.1.5	Controller Hardware and Software Considerations . . . . .	88
3.1.6	Human Factors Considerations . . . . .	89
3.2	ViGRR Robot Design . . . . .	90
3.2.1	Basic Simulation Modelling . . . . .	91

3.2.2	Drive Unit Sizing Requirements . . . . .	93
3.2.3	Optimization . . . . .	93
3.2.4	Drive Train Components . . . . .	96
3.3	Mechanical Design . . . . .	97
3.4	Hardware Overview . . . . .	97
3.4.1	Danaher Motion S200 Base Units . . . . .	98
3.4.2	ATI Force/Torque Sensor . . . . .	102
3.4.3	Safety Mechanisms . . . . .	104
3.4.4	Quanser Real-Time Motion Control Platform . . . . .	105
3.5	Summary . . . . .	108
<b>4</b>	<b>Controller Implementation</b>	<b>109</b>
4.1	Trajectory Generators . . . . .	110
4.1.1	Joint Space . . . . .	111
4.1.2	Cartesian Space Trajectory Generators . . . . .	112
4.2	Model-Based Nonlinear Control . . . . .	114
4.2.1	Parameter-Regressor Formulation of the Dynamics . . . . .	114
4.2.2	Inverse Dynamics . . . . .	115
4.2.3	Passivity-Based Nonlinear Control . . . . .	117
4.3	Task Space and Force Control . . . . .	118
4.3.1	Impedance Control . . . . .	119
4.3.2	Admittance Control . . . . .	120
4.4	ViGRR Controller Software Implementation . . . . .	121
4.4.1	Emergency Stops and Enables . . . . .	123
4.4.2	Sensors . . . . .	123
4.4.3	Controllers . . . . .	125
4.4.4	Safety Controls . . . . .	128

4.4.5	Redundancy Resolution Scheme . . . . .	130
4.4.6	Controller Graphical User Interface . . . . .	132
4.4.7	ViGRR Robot Control Simulink Library . . . . .	133
4.5	Simulation . . . . .	135
4.5.1	Sensor Models . . . . .	136
4.5.2	Visualization . . . . .	137
4.6	Calibration . . . . .	137
4.6.1	Basic Kinematic Calibration . . . . .	138
4.6.2	Single Linkage Dynamic Parameter Identification . . . . .	142
4.6.3	Full Robot Dynamic Parameter Identification . . . . .	153
4.7	Summary . . . . .	160
<b>5</b>	<b>Experimentation</b>	<b>161</b>
5.1	Joint Trajectory Tracking . . . . .	162
5.1.1	Results . . . . .	163
5.1.2	Interpretation of the Results . . . . .	163
5.2	End Effector Trajectory Tracking . . . . .	163
5.2.1	Results . . . . .	168
5.2.2	Interpretation of Results . . . . .	173
5.3	Demonstration of Admittance Control With Circular Visual Feedback	174
5.3.1	Results . . . . .	177
5.3.2	Interpretation of the Results . . . . .	177
5.3.3	Application to ViGRR Rehabilitation Controller Concept . . .	182
5.4	Summary . . . . .	185
<b>6</b>	<b>Conclusions and Future Work</b>	<b>186</b>
6.1	Conclusions . . . . .	187
6.2	Future Work . . . . .	189

<b>List of References</b>	<b>193</b>
<b>Appendix A Quanser Q8 and S200 Terminal Definitions</b>	<b>202</b>
<b>Appendix B Danaher Motion AKM Motor Specifications</b>	<b>206</b>
<b>Appendix C Harmonic Drive Unit Sizing Specifications</b>	<b>209</b>
<b>Appendix D Hall Effect Proximity Sensor Specifications</b>	<b>211</b>
<b>Appendix E Mini85 6-axis Force/Torque Sensor Calibration Accuracy Report</b>	<b>214</b>
<b>Appendix F ViGRR Regressor Matrices</b>	<b>217</b>

# List of Tables

2.1	Clinical joint angle definitions. . . . .	38
2.2	Joint angle conversion from anatomical convention to 3R manipulator. . . . .	40
2.3	Length of each leg segment $l_i$ as a fraction of body height $H$ . . . . .	47
2.4	Mass of each leg segment as a fraction of body mass $M$ and the centre of mass (CoM) location as a fraction of the segment length $l_i$ . . . . .	47
2.5	Radius of gyration $r_{g_i}$ for each mass segment of leg as a factor of segment length $l_i$ . . . . .	48
3.1	Weight and height of American males and females between 2003 and 2006. . . . .	79
3.2	Angle and Torque limits for leg joints. . . . .	82
3.3	Properties for generating end effector loading requirements used in drive train selection. . . . .	93
3.4	Nonlinear optimization objective function input parameter bounds (search space). . . . .	95
3.5	Nonlinear optimization results for design parameters. . . . .	96
3.6	Selected Components. Detailed specifications in [79–81]. . . . .	97
3.7	S200 base unit current controller parameters. . . . .	100
3.8	Drive unit properties. . . . .	101
3.9	Mini85 FT sensor ranges and specifications for the SI-1900-80 calibration. . . . .	102

4.1	Sensor filter coefficients for joint velocities (225 Hz) and FT sensor (3 Hz).	124
4.2	Controller Safety Limits.	129
4.3	ViGRR redundancy resolution scheme parameters.	132
4.4	Kinematic calibration results for link lengths.	140
4.5	Single DoF dynamic calibration results.	149
5.1	Joint step trajectory beginning and end values.	163
5.2	Circular end effector trajectory parameters.	168
5.3	Parameters used to generate circular end effector position $X_h^v$ for visual feedback.	176
5.4	Admittance control experiment parameters.	178
A.1	Description of S200 J4 D-sub Command IO pin assignments in use for the rehabilitation robot.	202
A.2	Q8 Data Acquisition Board Output Wiring Description	203
A.3	Q8 Data Acquisition Board Input Wiring Description	204

# List of Figures

1.1	An early version of the position-controlled Lokomat and a newer version outfitted with force/torque sensors and a treadmill capable of measuring ground-reaction forces for each foot. . . . .	9
1.2	The degrees of freedom for the LOPES robot. . . . .	12
1.3	The ALEX gait training results from a force channel controller. . . . .	14
1.4	The HapticWalker footplate safety mechanisms and device with spinal cord injury patient. . . . .	16
1.5	The MotionMaker and WalkTrainer. . . . .	17
1.6	ViGRR controller concept flow chart. . . . .	21
1.7	Configuration of ViGRR: Design considerations. . . . .	25
1.8	Overview of thesis and design process. . . . .	29
2.1	Frame, length and mass definitions for a planar n-revolute joint linkage system. . . . .	34
2.2	Limb segment angles and planar coordinate conventions for the leg. . . . .	38
2.3	Frame, length and mass definitions for a the 3R manipulator leg model. . . . .	39
2.4	Gait progression and ground reaction forces [21]. . . . .	48
2.5	Hip Joint angles vs. percent gait cycle. . . . .	50
2.6	Knee Joint angles vs. percent gait cycle. . . . .	50
2.7	Ankle Joint angles vs. percent gait cycle. . . . .	51
2.8	Normalized hip joint moments vs. percent gait cycle. . . . .	52

2.9	Normalized knee joint moments vs. percent gait cycle. . . . .	53
2.10	Normalized ankle joint moments vs. percent gait cycle. . . . .	53
2.11	Normalized horizontal (anterior/posterior) ground reaction forces vs. percent gait cycle. . . . .	54
2.12	Normalized vertical (anterior/posterior) ground reaction forces vs. percent gait cycle. . . . .	54
2.13	ViGRR Leg model calculated hip joint torques (Sim) during the swing phase compared to average gait data (Winter) with one standard deviation bounds [69]. . . . .	57
2.14	ViGRR Leg model calculated knee joint torques (Sim) during the swing phase compared to average gait data (Winter) with one standard deviation bounds [69]. . . . .	58
2.15	ViGRR Leg model calculated ankle joint torques (Sim) during the swing phase compared to average gait data (Winter) with one standard deviation bounds [69]. . . . .	59
2.16	ViGRR Leg model calculated vertical ground reaction force (Sim) during the stance phase compared to average gait data (Winter) with one standard deviation bounds [69]. . . . .	60
2.17	Frame, length and mass definitions for a the 4R ViGRR prototype robot.	61
2.18	Parameters and diagram for incompatible configuration avoidance. . .	70
2.19	Link 4 axes with FT sensor and hand coordinate frames. . . . .	74
2.20	Link 4 axes with FT sensor and foot coordinate frames. . . . .	75
3.1	Passive hip joint moments and bounded limits for workspace requirements [70]. . . . .	80
3.2	Passive knee joint moments and bounded limits for workspace requirements [70]. . . . .	81

3.3	Passive ankle joint moments and bounded limits for workspace requirements [70]. . . . .	81
3.4	5th percentile female height leg model workspace bounds. . . . .	83
3.5	95th percentile male height leg model workspace bounds. . . . .	83
3.6	Workspace Visualization GUI. . . . .	86
3.7	Overview of the design process for ViGRR. . . . .	91
3.8	Assembled joints and links of ViGRR prototype. . . . .	98
3.9	Topographical Layout of Robot Apparatus. . . . .	99
3.10	S200 drive unit torque mode controller [81]. . . . .	100
3.11	The Mini85 force/torque sensor. . . . .	103
3.12	Wiring for Encoder conversion from differential to single-ended. . . .	106
3.13	The Quanser Q8 DAQ with wiring and terminal labels in use with ViGRR. Descriptions of terminals are provided in Appendix A. . . . .	107
3.14	Wiring for Hall-effect proximity sensors. . . . .	108
4.1	Outer loop and inner loop controls architecture [63]. . . . .	110
4.2	ViGRR Controller Simulink model. . . . .	122
4.3	Force/torque sensor subsystem. . . . .	125
4.4	ViGRR end effector step input enabled subsystem. . . . .	128
4.5	Redundancy resolution scheme Simulink subsystem. . . . .	130
4.6	Inverse kinematics with redundancy resolution scheme subsystem mask. .	131
4.7	ViGRR graphical user interface for low-level control. . . . .	133
4.8	ViGRR visualization graphical user interface. . . . .	134
4.9	ViGRR simulink library hierarchy. . . . .	134
4.10	ViGRR dynamics simulation. . . . .	135
4.11	Force/torque sensor simulation. . . . .	136
4.12	Encoder sensors simulation. . . . .	137
4.13	Visualization of ViGRR X3D model. . . . .	138

4.14	Kinematic calibration error for recalculated end effector X position. . .	140
4.15	Kinematic calibration error for recalculated end effector Y position. . .	141
4.16	Kinematic calibration error for recalculated end effector orientation. . .	141
4.17	The command joint 1 torques (measured) and the computed torques (model) from the single DoF dynamic calibration. . . . .	145
4.18	The command joint 2 torques (measured) and the computed torques (model) from the single DoF dynamic calibration. . . . .	145
4.19	The command joint 3 torques (measured) and the computed torques (model) from the single DoF dynamic calibration. . . . .	146
4.20	The command joint 4 torques (measured) and the computed torques (model) from the single DoF dynamic calibration. . . . .	146
4.21	Dynamic calibration error for recalculated torques from Joint 1 single DoF calibrated model parameters. . . . .	147
4.22	Dynamic calibration error for recalculated torques from Joint 2 single DoF calibrated model parameters. . . . .	147
4.23	Dynamic calibration error for recalculated torques from Joint 3 single DoF calibrated model parameters. . . . .	148
4.24	Dynamic calibration error for recalculated torques from Joint 4 single DoF calibrated model parameters. . . . .	148
4.25	Close up of joint 3 model and measured torques for the single link dynamic calibration. . . . .	152
4.26	Close up of joint 4 model and measured torques for the single link dynamic calibration. . . . .	152
4.27	The command joint 1 torques (measured) and the computed torques (model) from the parameter dynamic calibration. . . . .	155
4.28	The command joint 2 torques (measured) and the computed torques (model) from the parameter dynamic calibration. . . . .	155

4.29	The command joint 3 torques (measured) and the computed torques (model) from the parameter dynamic calibration . . . . .	156
4.30	The command joint 4 torques (measured) and the computed torques (model) from the parameter dynamic calibration. . . . .	156
4.31	Dynamic calibration error for calculated joint 1 torques from 4DoF calibrated model parameters. . . . .	157
4.32	Dynamic calibration error for calculated joint 2 torques from 4DoF calibrated model parameters . . . . .	157
4.33	Dynamic calibration error for calculated joint 3 torques from 4DoF calibrated model parameters. . . . .	158
4.34	Dynamic calibration error for calculated joint 4 torques from 4DoF calibrated model parameters. . . . .	158
4.35	The dynamic parameter $\phi$ from the single link dynamic calibration $\phi_s$ and the full robot calibration $\hat{\phi}$ . . . . .	160
5.1	ViGRR Controls setup for experiment implementation. . . . .	162
5.2	Position trajectory tracking for joint 1. . . . .	164
5.3	Position trajectory tracking for joint 2. . . . .	164
5.4	Position trajectory tracking for joint 3. . . . .	165
5.5	Position trajectory tracking for joint 4. . . . .	165
5.6	Tracking position error for joint 1 smooth step trajectory. . . . .	166
5.7	Tracking position error for joint 2 smooth step trajectory. . . . .	166
5.8	Tracking position error for joint 3 smooth step trajectory. . . . .	167
5.9	Tracking position error for joint 4 smooth step trajectory. . . . .	167
5.10	End effector X-Y plane position trajectory tracking. . . . .	169
5.11	End effector X-Y plane position trajectory error. . . . .	169
5.12	End effector X position trajectory tracking. . . . .	170
5.13	End effector Y position trajectory tracking. . . . .	170

5.14	End effector orientation trajectory tracking. . . . .	171
5.15	Tracking X position error for end effector smooth step trajectory and circular path following. . . . .	171
5.16	Tracking Y position error for end effector smooth step trajectory and circular path following. . . . .	172
5.17	Tracking orientation error for end effector smooth step trajectory and circular path following. . . . .	172
5.18	Admittance control with visual feedback flow chart. . . . .	175
5.19	Test setup for circle trajectory following task. . . . .	176
5.20	Visualization feedback for user. . . . .	177
5.21	End effector X-Y plane position trajectory tracking. . . . .	178
5.22	End effector X-Y plane position trajectory error. . . . .	179
5.23	End effector X position trajectory tracking. . . . .	179
5.24	End effector Y position trajectory tracking. . . . .	180
5.25	Tracking X position error for end effector smooth step trajectory and circular path following. . . . .	180
5.26	Tracking Y position error for end effector smooth step trajectory and circular path following. . . . .	181
5.27	The ViGRR detailed gait rehabilitation controller concept flowchart. .	183
A.1	Quanser Q8 specification sheet [84] . . . . .	205
B.1	Motor torque-speed curve definitions [79] . . . . .	206
B.2	AKM2x motor specifications [79] . . . . .	207
B.3	AKM4x motor specifications [79] . . . . .	208
C.1	CSF Harmonic Drive Unit Sizing Specifications [80] . . . . .	209
C.2	CSG Harmonic Drive Unit Sizing Specifications [80] . . . . .	210
D.1	Hall effect sensor operating range [86] . . . . .	211
D.2	Hall effect sensor timing diagram [86] . . . . .	212

D.3	Hall effect sensor electrical characteristics [86]	213
D.4	Hall effect sensor magnetic characteristics [86]	213
E.1	Mini85 SI-1900-80 list of applied loads during calibration [87]	215
E.2	Mini85 SI-1900-80 list of percent errors for applied loads during calibration [87]	216

# Chapter 1

## Introduction

Impairment of motor control is a debilitating outcome of neurological disorders or injuries sustained to the central nervous system (CNS). There is an ageing population and the incidence of neurological disorders are expected to increase [1]. If we wish to meet the growing demands of neurorehabilitation, it is prudent to investigate the use of new tools, technologies, and practices that will help maintain a healthy population.

For sufferers of Parkinson's disease, cerebral palsy, stroke, spinal chord injuries, or other CNS disorders, regaining lost motor control through rehabilitation can greatly improve the quality of life of the affected individual. An important goal of rehabilitation is to allow the patient to increase their participation in social activities and interactions that define their life role by restoring functional ability [2]. Ambulation is a key functional ability for a person's autonomy, and research in gait rehabilitation is continually evolving to provide better therapy strategies that improve patient outcomes and quality of life [3, 4].

Recent approaches in gait rehabilitation involve research into use of robotics [5, 6]. Assisting gait rehabilitation using robotics has emerged as a result of moving toward a focus on regaining lost function by applying principles for motor learning as opposed to compensatory methods such as walking aids, orthotics or wheelchairs. Relearning a lost motor function is primarily facilitated by engagement in task-specific repetition,

focused attention and high intensity [4, 7, 8]. Robotics research in this field is heavily influenced by their potential to meet these learning requirements and compliment therapist driven strategies. There is a great research potential in exploring new designs for robotic gait rehabilitation and pursuing novel control strategies that can improve patient outcomes [6]. This thesis presents the design and implementation of a novel gait rehabilitation platform prototype whose ultimate purpose is to explore new therapy control strategies.

## **1.1 Motivation**

A novel robot design and controller implementation for gait rehabilitation is motivated by the potential for better assessment and analysis, flexibility of the robot platform, improved patient motivation through virtual environments, and feedback mechanisms that are useful for the patient and therapist. The motivating factors described in this chapter are the impetus for the development of a prototype scaled version of a new gait rehabilitation robot presented in this thesis.

### **1.1.1 Principles of Motor Learning and Gait Rehabilitation**

Research into rehabilitation and neuroscience revealed new insights into the phenomenon of learning [7]. The central nervous system has the ability to adapt and reorganize neurons and other cells to regain lost function or learn a new skill. This adaptive capacity, called neural plasticity, occurs spontaneously and also in the absence of rehabilitation therapy. Individuals who sustain injuries to the CNS learn new methods of coping with their disability and develop new skills with coordinated movements. However, self-initiated compensatory behavioural changes may not be ideal and can cause a reliance on the less affected part of the body such as in hemiparetic patients. Rehabilitation therapies can improve the outcome of the patient

by facilitating motor learning and neural plasticity. The primary factors causing experience-dependant neural plasticity in rehabilitation are characterized as a set of principles summarized below. By focusing on these principles, rehabilitation therapists have been able to restore lost function in CNS impaired patients [4, 7, 8].

**Intensity and specificity** Specificity refers to a specific skill being developed and intensity is a measure of the effort input to the activity. Simply repeating unskilled movements or exercises does not facilitate plasticity. The subject must be actively engaged in attempting to perform the skill with a high level of effort and attention.

**Repetition** Although repetition of an unskilled or previously learned motion may not cause plasticity, repetition of newly learned movements is essential for lasting changes to take effect. This is very important in rehabilitation for the learned function to be maintained such that the patient can perform the activity at home and make further restorative gains without having to repeatedly relearn tasks.

**Time** Plasticity also takes time, as it is not a single event but a process that may take weeks or months to have a lasting effect for motor recovery. Studies have shown that therapy initiated within days of the injury improved outcome compared to initiating several weeks afterwards, but there is no clear time frame when rehabilitation cannot be initiated.

**Salience** The relevant experience and importance, or salience, of a skill is a key factor in causing plasticity and motivating the patient. This principle is often difficult to exploit in a clinical rehabilitation setting where repetition, intensity, and long recovery periods make it onerous to motivate the patient or provide a clear connection with tasks associated with daily living.

### **1.1.2 The Role of Robotics in Rehabilitation**

New rehabilitation therapies emerged as a result of studies in how plasticity is mediated by various exercises and stimuli. As therapies shifted from a compensatory focus to a behaviour-driven learning model, robotics were introduced to perform repetitive, high-intensity activity-based exercise such as in gait training [9, 10] or to teach coordinated motor tasks with visual and/or haptic feedback for arms [11, 12]. Initial results showed promise for rehabilitation robotics and spurred research over the last decade [5, 6, 13].

The primary goal of any practice in rehabilitation is improving the quality of life of the patient. Achieving functional gains and other measurable positive outcomes of rehabilitation serve to meet this goal. Despite the continued growth and development of robotics in rehabilitation, few studies have shown significant improvements in patient outcomes compared to other practices without the use of robotics and clinical trials are in their infancy [6, 14, 15]. There are many identified potential advantages robotics can offer, but there is still a need to work toward bridging the gap between technologically advanced systems and the needs of patients and therapists in a clinical setting [1, 14, 16–18]. By identifying the general limitations, advantages, and research potential of robotics, the design and implementation of a new robotics platform for gait rehabilitation can be motivated.

#### **Complexity and Relevance to Rehabilitation**

A limiting factor of robotic systems is that they are inherently complex in their mechanical design, actuation, and controls. With this complexity, there are associated costs, power requirements, maintenance, development time, training, and technical expertise. These are largely technical limitations that can be mitigated by good design practices, economic cost/benefit analysis, a focus on research, and tight

collaboration with all parties involved in the rehabilitation process.

Another disadvantage of robotics is the lack of tactile feedback and human interaction between the therapist working and the patient. There is a relationship, compassion, and understanding developed between them. It is their experiences that shape the rehabilitation process through analysis, exercise, motivation, assessment and ultimately the patient's outcomes. Robotic devices cannot replace that experience, but they can significantly enhance it by meeting the potential advantages robotics has to offer [1, 15].

### **Repetition and Intensity**

A primary identified advantage of robotics is alleviating the exertion and physical burden on the therapist. Activity-based therapy often requires the explicit and consistent assistance of one or more therapists working with the patient. High repetition and intensity are important factors in neural plasticity, and robotics can support prolonged exercise without endangering the therapist's health [19].

### **Assessing and Monitoring the Patient**

Analysis, monitoring and assessment are components of rehabilitation that can benefit from the use of robotics. Monitoring of the patient's progression with the help of a robotic device and capable sensors can provide a means of quantitatively assessing the patient and their functional gains over the course of therapy. Detailed information about the kinematic and estimated effort of the patient can help therapists provide more effective and targeted therapies [20].

Gait analysis for example, is typically performed in a facility separate from the therapeutic exercises and practice. It is used as a tool for assessing pathology of the patient, functional assessment, and development of a treatment plan [21–24]. The process of gait analysis, however, is time consuming and requires significant technical

resources and space. Continuous assessment of the gait with a robotic device can provide insights into the patient's functional ability without the extraneous time, instrumentation and space associated with gait analysis. [20, 25, 26].

### **Flexible and Targeted Control Strategies**

A robust actuation system with feedback from the sensor data regarding the state of the user during therapy may also allow for greater versatility and intelligence in therapeutic activity regimens compared to passive exercise machines. As suggested by principles for mediating neural plasticity, active participation and appropriate intensity are important factors for rehabilitation. Robotics controllers can be implemented to exploit the sensor data for targeting specific motion, joints or behaviours and adjusting intensity as the patient progresses [6, 11, 12, 27–29]. A deeper understanding of the mechanisms of motor learning and rehabilitation can also be discovered through analysis of data collected using rehabilitation robotics [5].

### **Sensory Feedback and Virtual Environments**

With regards to salience and keeping the patient engaged, visual and tactile (haptic) feedback directly linked to the device actuation and sensing allows greater immersion and can keep the patient motivated. A number of rehabilitation robotics approaches utilize virtual environments with haptic devices or visual feedback cues [6, 26, 30, 31]. These approaches can be closely linked to novel control strategies that utilize the virtual environment for high-level decision-making and controller feedback mechanisms. Research in providing simplified but relevant quantifiable analysis data to the therapist and patient during rehabilitation exercises can also be exploited for use as visual feedback mechanisms [26, 32, 33].

## Research Potential

The lack of widespread randomized clinical trials demonstrating the advantages of robotics is a barrier to adoption of robotics in a clinical setting [6]. However, this limitation is also one of the strongest motivators for research with robotics and rehabilitation. There is strengthening interest in robotics as prevalence of neurological disorders grow and solutions will be needed to match the demands of our ageing population. This demand may be met by pursuing the identified potential advantages yet to be fully realized by research in rehabilitation robotics.

Robotics cannot be viewed as a replacement for the established and evolving theory and practice of rehabilitation as we know it today, nor can they be seen as expensive exercise machines. Rather, robotics in the clinic will likely become a tool which compliments the clinicians, neurologists, therapists and rehabilitation engineers working with the patient. More importantly, robots can be intelligent, programmable, adaptable, and robust, making them ideal candidates for research into the mechanisms and best practices by which gait rehabilitation can be most effective.

### 1.1.3 Approaches to Robotics-Assisted Gait Rehabilitation

For gait rehabilitation, improved outcomes were observed with body weight supported treadmill training (BWSTT) which is now widely accepted as a beneficial practice for gait impairments caused by CNS injuries such as in stroke patients [8,34]. Treadmill-based exercise elicits functional improvements in timing and symmetry of muscle activations during gait when controlling the walking speed and by partially supporting body weight for severely impaired individuals. However, this therapy is time consuming, repetitive and extraneous for the therapist assisting the patient. These limitations, coupled with the growing research in implementing robotics for upper arm rehabilitation, provided the motivation for the application of robotics to gait rehabilitation [5].

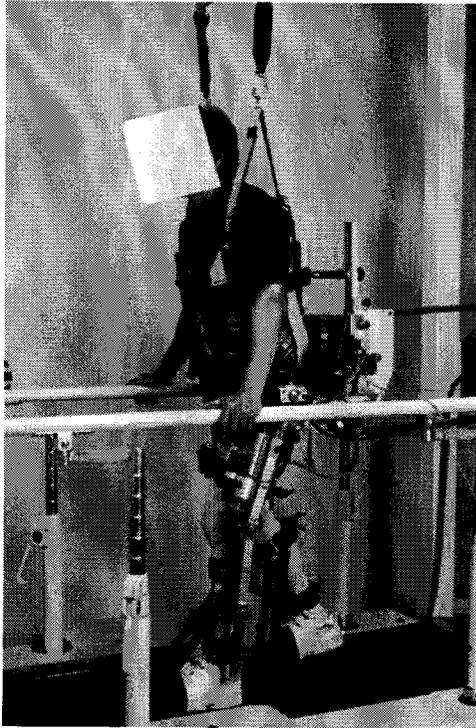
BWSTT motivated the application of a commercial robotic gait orthosis called Lokomat to assist with the high level of repetition and therapist exertion when treating patients with severe paresis [19]. Since the Lokomat's introduction, research in robotics-assisted gait rehabilitation has expanded to involve a number of robotic devices and investigations into rehabilitation-focused controllers implementing the technology.

The design of robotic devices for gait rehabilitation varies greatly in controller implementation, scale, configuration and actuation, as the complexity in the gait rehabilitation process elicits a multitude of design challenges and interpretations of the needs of patients and therapists alike. A selection of these devices are reviewed here, with each design having its own purpose, advantages and limitations. The conceptual design for a new gait rehabilitation device is motivated by the conclusions drawn and unresolved issues posed by the implementations and experimental results from the robots described in the following sections.

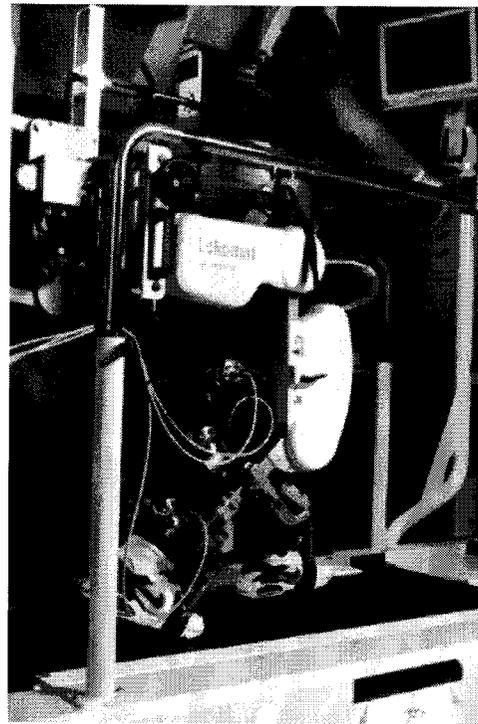
### **Lokomat**

The Lokomat shown in Figure 1.1 was originally developed as a guiding position-controlled device that moved the patient through average gait trajectories without feedback from measurements of the actual effort supplied by the user [19]. This approach was optimistic in its potential benefits for spinal cord injury patients and alleviating therapist exhaustion. However, it did not provide measurements of the patient's effort throughout the exercise (or the ability of the therapist to determine by feel how the patient was progressing), was not validated as an improved rehabilitation technique compared to BWSTT, and did not necessarily provide increased motivation for the patient.

In order to provide the capability of assessing the patient's efforts while using the Lokomat, force/torque (FT) sensors were installed [36]. New control strategies were



(a) Early Lokomat model [19]



(b) Recent Lokomat model [35]

**Figure 1.1:** An early version of the position-controlled Lokomat and a newer version outfitted with force/torque sensors and a treadmill capable of measuring ground-reaction forces for each foot.

tested that implemented force feedback with adaptive assistance with simple visual feedback to the patient [29]. With the improved instrumentation, a more patient-driven approach was applied where joint torques were measured through inverse dynamics and modelling of the human occupant and compliance was programmed into the controller. The new controller was intended to provide variable compliance or assistance, increase motivation of the patient, and act more like a therapist by responding to the patient's changes in gait measured by the sensors.

Further development of the Lokomat was initiated by investigating the powered orthosis as a haptic device with a virtual environment projected onto a display. An obstacle scenario with a toe impact model, force feedback impedance controller and even the effect of surround sound was tested [37,38]. Originally designed as a position

controlled robotic device, the Lokomat proved to be limited as a haptic display [37]. However, combining patient-driven control strategies and virtual environments may increase patient motivation and improve how patients cope with obstacles.

Studies measuring the efficacy of the Lokomat began to appear. Initial outcomes from a clinical trial with 16 participants were deemed positive and expressed the need for larger scale randomized clinical trials [39]. Detailed studies of the kinematics and joint torque patterns were investigated when using the device [35,40,41]. These studies concluded that there is significant movement of the users while using the Lokomat relative to the motion of the powered gait orthosis but with the pelvis and arm motion being constrained. This led to different muscle activations and kinematic patterns to compensate for the lacking degrees of freedom. Abnormal joint torque patterns of stroke patients were not significantly modified when using the device despite the forced symmetric walking pattern implemented by the Lokomat.

A large scale clinical trial measuring outcomes using the Lokomat demonstrated that the Lokomat did not perform as well as traditional therapy methods for subacute stroke patients but still showed improvement in nearly all of the measured outcomes [42]. The most significant discrepancies were observed for improvements in overground walking speed and endurance. A number of limitations of the Lokomat were identified as the reasons behind these findings. First, the Lokomat places restrictions on the pelvis and arm movement, limiting pelvic rotation and weight shifting which could alter gait patterns of the lower limb and restrict muscle activation for stability and propulsion. There is also the limited guidance and feedback provided to the patient during the gait training. Abnormal joint moments and muscle activation patterns may still occur despite being moved through normal kinematic gait patterns and there were no feedback mechanisms in place to measure the effort of the participant as they were moved through a predefined gait trajectory. In this study, the Lokomat models were not outfitted with the FT sensors or the adaptive controller.

The trials were also limited in that they studied only the exclusive use of the Lokomat trainer and did not investigate the potential advantages of a multidisciplinary approach with both Lokomat and over-ground training at various stages of recovery [42].

## **LOPES**

The LOPES, or lower extremity powered exoskeleton, is a device intended for gait rehabilitation with selective support of gait functions during treadmill training. The actuation [43] and design [44] of the device focus on the ability to provide impedance control and allow the user to experience walking while being unhindered by the orthosis (patient in charge), having selective assistance from the robot, or having the robot move the participant in a predefined trajectory (robot in charge) [45].

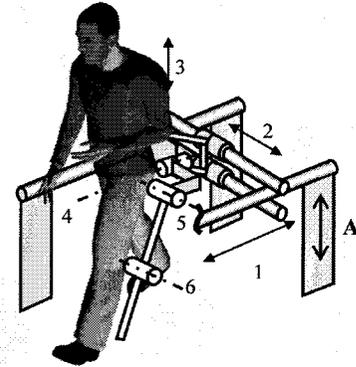
Virtual models were applied to help specific gait functions or sub-tasks. For example, spring-dampers systems can be virtually applied to various parts of the body in order to simulate assistance related to real-world actions such as body weight support, balance and stabilization support, step-length increases, or foot clearance during the swing phase of gait [45]. Initial experiments in providing toe clearance assistance were performed by demonstrating the ability to selectively modify the foot height during gait using a virtual model [46,47].

Particular attention was placed on the number of degrees of freedom (DoF) for the LOPES design so that a user can walk normally, safely and comfortably while optimally minimizing the required number of actuators. A total of 8 DoFs were included in the design and are shown in Figure 1.2 [44]. The motivation for not including ankle actuation in this design is that the only necessary ankle function was to ensure foot clearance during the swing phase which can be accomplished using a passive orthosis. Active assistance, if necessary, can be accommodated by an external powered device attached to the foot [44]. However, issues regarding compensating for the inertia and gravitational effects of such a device, especially if it needs to

be attached serially to the exoskeleton, were not addressed. The ankle is also an important aspect of healthy gait, especially for propelling the user forward with large muscle activation required by the soleus and gastrocnemius (calf) muscles [48, 49].

DoF	Human possibility	Robot design		
		Actuated	Free	Blocked
Pelvis to fixed world	3 rotations 3 translations	Left/Right Forwd/backwd	Up/down	All rotations except for play
Hip	3 rotations	Ad-/Abduction Flex-/extension	-	Exo/endo-rotation except for play
Knee	1 rotation (sagittal)	Flex-/extension	Ad-/Abduction <sup>1</sup>	-
Ankle	3 rotations <sup>2</sup>	-	All motions	-

(a) Table listing human and LOPES DoFs [44]



(b) Illustration of LOPES DoFs [44]

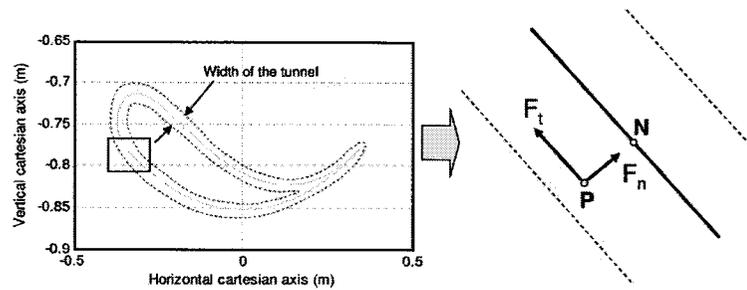
**Figure 1.2:** The degrees of freedom for the LOPES robot.

Having fewer DoFs actuated than is capable by human biomechanics can limit the ability of the robot to feel transparent for normal gait or to provide comprehensive rehabilitation therapies related to stability and pelvic motion. The effect of limiting DoFs with the LOPES robot was studied on healthy subjects to assess the differences in gait kinematics and muscle activations [44, 50]. Blocking horizontal translations of the pelvis and frontal rotations of the hip did not significantly change the behaviour of the subject when using the device. However, changes between treadmill walking and use of the LOPES exoskeleton in healthy individuals were observed. When using the device, there was a decrease in knee angle range and muscle activations on toe off and heel strike. Increases in muscle activity were observed for larger accelerations involved in the leg swing. Timing was largely unaffected and the gait appeared normal with the exoskeleton [50]. Whether or not these discrepancies will affect assist-as-needed and selective gait training control regiments have yet to be tested.

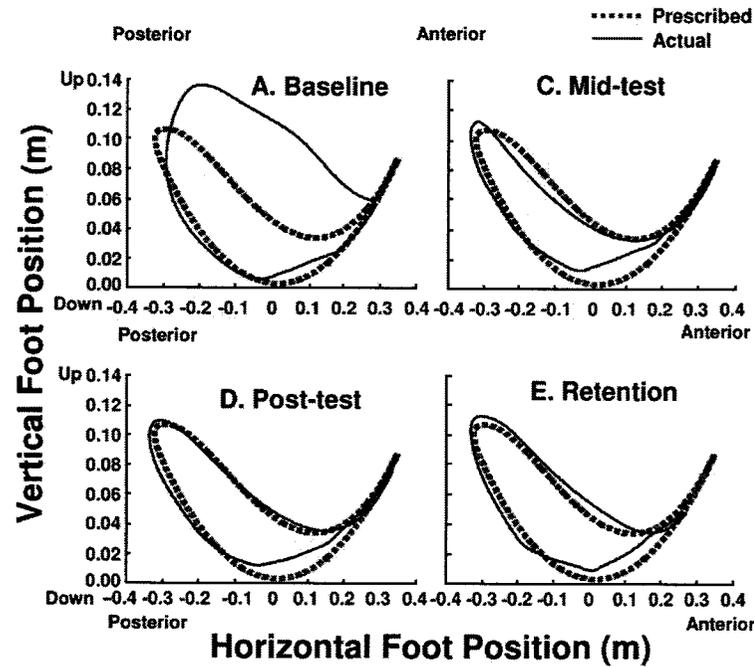
## ALEX

The Active Leg Exoskeleton (ALEX) is a treadmill-based powered orthosis which uses FT sensors and visual feedback with a novel control scheme for gait rehabilitation. This device features similar DoFs to the LOPES with the added rotation of the trunk about the vertical axis and a passive ankle joint. The hip and knee joints are powered in the sagittal plane and use friction-compensated linear actuators. A single leg is controlled with the prototype device [51, 52].

The control strategy employed by ALEX involves a desired trajectory of the foot in the sagittal plane, with an impedance controlled force channel which applies a non-linear restoring force in the direction normal to the desired trajectory if the person's foot deviates too much from the path being followed. A tangential force assists the user to move their foot along the path as shown in Figure 1.3 (a) with real-time visual feedback provided to the user. Several healthy subjects were tested with the controller. Figure 1.3 (b) shows a typical test result. A baseline reading was taken without the force channel or visual feedback (A. Baseline), the training was performed, and a final reading without the tunnel or visual feedback was taken (E. Retention). The retention plot shows short-term improvements in the trajectory following task [52]. Whether these initial findings will translate to improved patient outcomes has yet to be determined.



(a) ALEX trajectory following force channel [52]



(b) Representative result of a test subject [52]

Figure 1.3: The ALEX gait training results from a force channel controller.

## HapticWalker

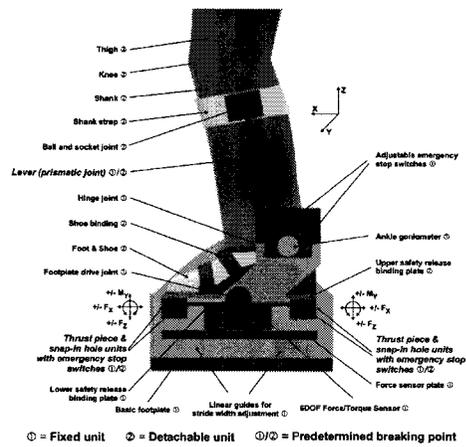
The HapticWalker shown in Figure 1.4 takes a fundamentally different approach to the design of a gait rehabilitation robot compared to those incorporating treadmills. This device uses actuated foot plates as opposed to an exoskeleton coupled with a treadmill. A body harness is used as a safety mechanism and for partial body weight support. The motivation for using such a configuration stems from the idea that multiple environments can be simulated with the device such as ramps, stairs, or

obstacles. Exoskeletons also inhibit torso motions and require modification of the linkage lengths according to each individual's comfort level and size. Alternatively, a foot plate design allows for easier donning and doffing of the device, access to the patient by the therapist, and free movement of the pelvis, torso and arms [53].

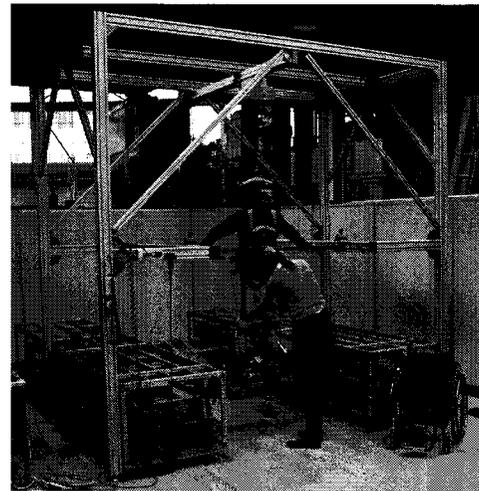
The Lokomat design was motivated by the ability to alleviate all physical effort by the therapist during treadmill training. With the HapticWalker, the therapist is still required to assist the user during walking motion as there is no active assistance over the hip, ankle and knee joints. This aspect of the design was recognized by a retrospective study of the HapticWalker and reiterated the viewpoint that robotics is still meant to be used as a tool and not a replacement for the therapist, whose personal experience and touch are invaluable to the rehabilitation process [9]. This review also highlighted the need for more clinical trials comparing the growing number of approaches to gait rehabilitation using robotics in terms of practicability, efficacy, and cost-effectiveness. Implications related to using the device as a haptic display and applying unmeasured external loads to the patient by the therapist were not addressed. The kinematics of the body are also not measured.

Greater care in the safety mechanisms must be taken into account for a foot plate since the robot is not anthropomorphic and hard stops cannot be relied upon to ensure the safety of the user. The haptic walker employs a variety of safety mechanisms built into the real-time controller and foot plate, such as mechanical releases, ankle angle sensors, and emergency stop safety switches shown in Figure 1.4 (a) [54].

A position controller was implemented where the end effector trajectories were generated using captured gait or stair climbing motion data then fixing the centre of mass (CoM) of the individual along the horizontal axis. This mode was compared to motion generated relative to a fixed CoM in both the horizontal and vertical axes (ie. very little vertical motion of the trunk) [55, 56]. In comparison with over ground (or



(a) Foot plate safety mechanisms [54]



(b) The HapticWalker [53]

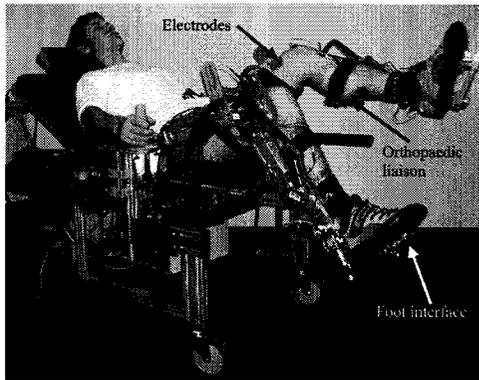
**Figure 1.4:** The HapticWalker footplate safety mechanisms and device with spinal cord injury patient.

real stair) muscle activation patterns based on electromyographic (EMG) measurements, participants using the device experienced lower amplitudes and delayed thigh muscle activations that may be attributed to the position control of the robot. No significant EMG differences were observed between the two modes [56].

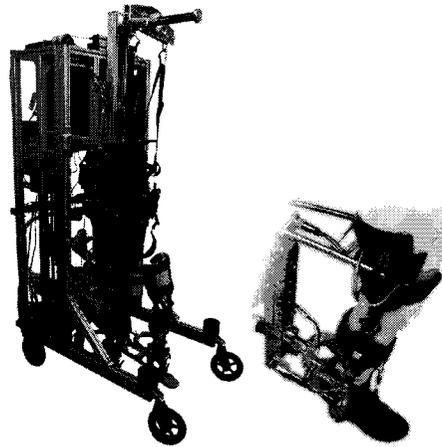
Ground reaction forces were recently studied using the footplate FT sensors for their utility in assessing the patient as well as likeness to over-ground walking and stair climbing [57]. Although the device was completely position controlled over the trajectory, results showed similar peak force values with slightly flattened curves due to premature toe off and heel strike when using the device. The ground reaction force data is expected to be useful for visual feedback cues to the patient and therapist. Future iterations of the device are intended to have admittance control to provide haptic feedback and variable assistance while still providing a stiff virtual ground [57].

## MotionMaker and WalkTrainer

The MotionMaker shown in Figure 1.5 (a) is an exercise device for the legs that combines synchronized leg motion with functional electrical stimulation (FES). Paraplegic individuals require movement of the lower limbs in order to prevent complications associated with immobilization. Accordingly, FES can be applied as a mobilization regiment. Incomplete SCI patients can also benefit from FES to facilitate recovery of motor control. The device controls sagittal plane motion of each leg with linear actuators driving revolute joints at the hip, knee and ankle. The device is position controlled for leg press and cycling trajectories. A preliminary trial demonstrated the applicability of the device, showing some improved muscle contraction strength with a leg press motion for three of the four SCI patients after a two month training program [58].



(a) The MotionMaker [58]



(b) The WalkTrainer [59]

**Figure 1.5:** The MotionMaker and WalkTrainer.

A second device called the WalkTrainer shown in Figure 1.5 (b) built upon the MotionMaker and the fundamental concept of combining FES with position controlled motion. The WalkTrainer is a mobile system where the patient is oriented vertically with a powered gait orthosis, body weight support harness and

electrostimulator [59–61]. The pelvic support device is a parallel structure with 6 degrees of freedom using 6 linear actuators. The legs are actuated with another parallel structure that interfaces with the foot and thigh. The mobile structure was driven over flat ground as the gait motions are synchronized with FES. Selective compliance of the actuation was programmed into the device with virtual stiffness and damping for each DoF. A basic clinical trial demonstrated the feasibility of the WalkTrainer for SCI patients by having a patient use the WalkTrainer but did not draw significant conclusions regarding the efficacy of the device [59].

### **Upper-Extremity Rehabilitation Robots**

What separates ALEX from other gait training devices is its controller strategy which is unique to most gait rehabilitation robots. Its controller can be characterized as an impedance-based force feedback channel, which is a much more common approach employed by arm-based rehabilitation robots. In the review paper [6], the ALEX controller was categorized as only one of 3 devices that focused on impedance-based controls for the lower extremity, compared to 18 articles for the upper extremity. The control strategies provided in [6] highlight the growing library of approaches for rehabilitation based on principles of behaviour-initiated plasticity and motor learning and there is potential for gait rehabilitation robots to investigate (and potentially adopt) better control algorithms.

However, it is unclear what algorithms are most effective for a particular type of rehabilitation process. As further systematic clinical studies characterize the efficacy of various control algorithms and associated exercises for particular motor impairments, robotics can become an effective tool in providing therapy according to the patient’s specific pathology, recovery stage, and activity [6].

## 1.2 ViGRR: A Novel Rehabilitation Robot

A new lower extremity rehabilitation robot, called the Virtual Gait Rehabilitation Robot (ViGRR), is conceptualized and motivated by the need for a robust haptic platform to conduct research for improved gait rehabilitation therapies. As a part of Carleton University's Advanced Biomechatronics Laboratory (ABL), the design and implementation of ViGRR is part of ABL's mandate to perform research in human mimetic, interactive, and biomechanical robotic systems. Primary goals driving the robot design and eventual implementation are to provide virtual environments with strategic therapies and haptic feedback, capture real-time measurements for assessment of the patient's abilities and progress, and ultimately improve the functional outcome of patients with CNS disorders or injuries. Corollary goals and milestones for a full-scale implementation include:

1. Design, build and test a prototype device with reduced degrees of freedom and scope that will serve to demonstrate basic capabilities of the device.
2. Implement and test intelligent controllers on the prototype as a proof-of-concept to teach new motor skills and provide assessment of the user's performance.
3. Design, build and test a full scale device.
4. Implement a virtual environment with tactile and visual feedback for a selection of skill-development scenarios and controllers.
5. Demonstrate the feasibility of the system for learning a new motor skill with healthy individuals.
6. Demonstrate the ability of the robot to improve functional outcomes for patients with CNS injuries or disorders by conducting clinical trials.

ViGRR will incorporate a form factor similar to MotionMaker, with a supine individual for ease of use and accessibility but with the ability to rotate the torso angle to an upright configuration with the same device. This format can accommodate patients with a wide range of functional abilities. Like the HapticWalker, ViGRR will incorporate foot plates and force/torque (FT) sensors for interaction with the user, providing flexibility in the types of exercises, interactive controllers, and patient assessment capabilities. By minimizing the adjustments of the device such as those required for linkage lengths in exoskeleton type robots, ViGRR should be safer and easier to don and doff. The preliminary design of a prototype robot incorporates this basic form factor for ViGRR and forms a basis for subsequent work toward realizing the device.

The primary purpose of a limited prototype device (Step 1) is to test controller and functional concepts to be implemented by the full scale robot. Completing this initial development stage for ViGRR is the primary goal for this thesis. The following sections provide an overview of a generic full scale rehabilitation controller concept for ViGRR, motivate the prototype configuration, and outline the contributions of the thesis related to ViGRR's prototype development.

### **1.2.1 Controller Concept for ViGRR**

The functionality of ViGRR should include the ability to develop a multitude of rehabilitation-focused controllers. This section further motivates this requirement and provides examples of potential applications that could make use of a more generalized haptic device as a gait rehabilitation tool.

The principles of behaviour-initiated neural plasticity and motor learning is applicable to both the lower extremity and the upper extremity [8], yet there is a deficiency in the number of approaches and control algorithms for gait rehabilitation compared to reaching tasks with the arm [6]. This discrepancy may be in part due to costs and

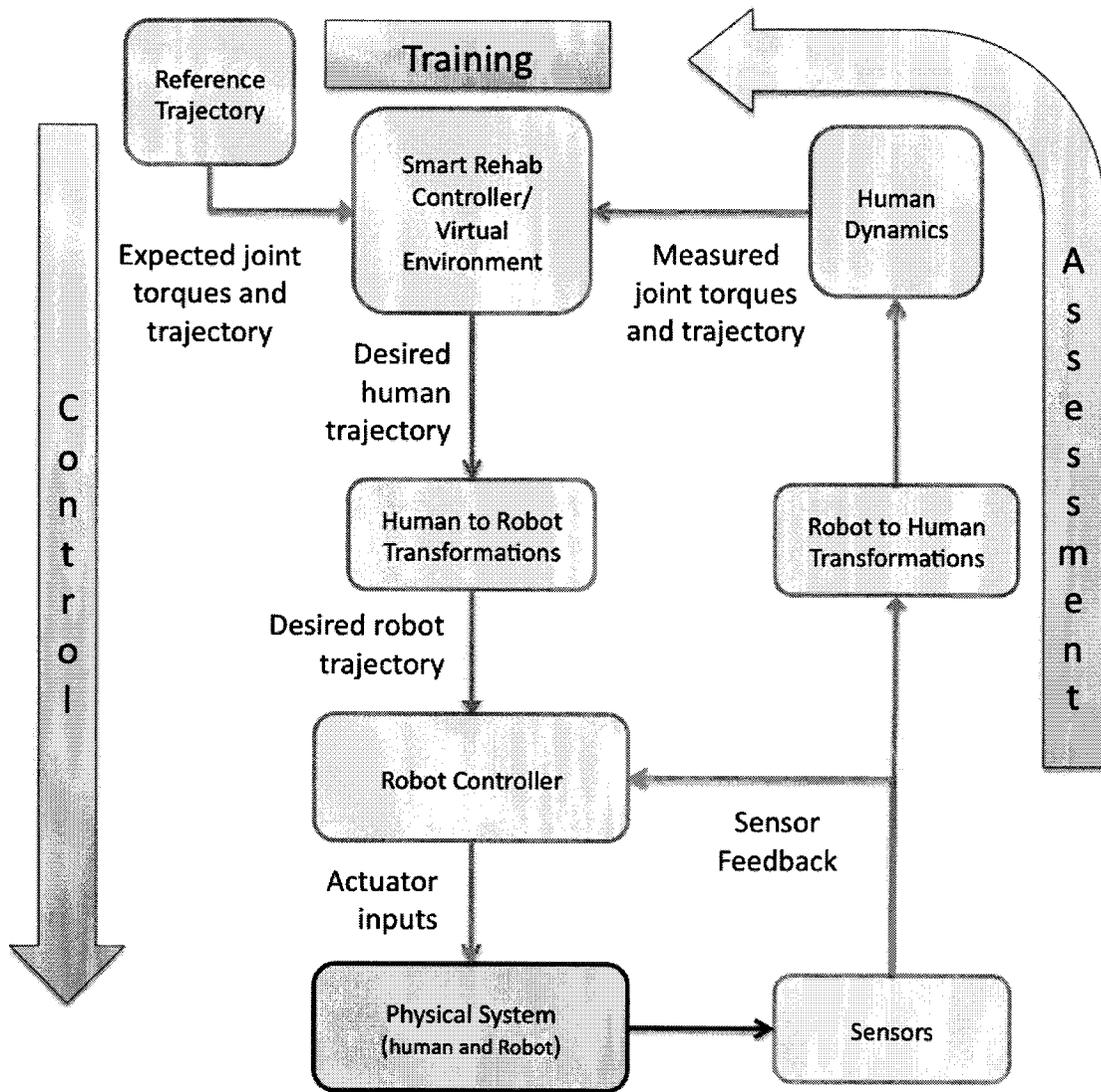


Figure 1.6: ViGRR controller concept flow chart.

technical limitations associated with more powerful actuators required for robotics assisted gait rehabilitation and the focus of robotics on primarily the single repetitive task of ambulation with average gait patterns.

Of the algorithms presented in the paper, for example, controllers that focus on haptic interfaces with virtual environments pose an attractive approach to motivating the patient while initiating highly versatile assistance mechanisms [6]. Haptics for gait

rehabilitation is part of the design focus for LOPES, HapticWalker and new instrumentation for Lokomat [29, 47, 57]. However, the control algorithms are still largely based on mimicking or replacing therapist assistance during body weight supported treadmill training. Rather than attempting to replace the capabilities of a therapist, one could adopt a controller focused on having the participant learn an explicit new motor task where relevant visual feedback is a key component of the training, much like upper extremity rehabilitation robots.

This concept may be extended to attempting lower extremity rehabilitation exercises that focus on improving subtask functions by teaching appropriately targeted, skilled motions. These subtasks can be administered in the context of a virtual gaming environment as opposed to only focusing on performing a normal walking trajectory on a treadmill. Subtasks may include balance recovery in the sagittal or frontal plane, strengthening hip flexors, improving symmetrical muscle activations, improving dorsiflexion of the foot during the swing phase, etc. Skills taught through a virtual environment to challenge and motivate the patient may include balancing on a (forgiving) plank and ball with virtual gravity, shifting weight from foot to foot to steer a race car, timing altering leg press movements according to musical and visual cues, etc. The intensity, repetition and difficulty of the skills may be dynamically changed through adaptive control schemes, or set by the therapist.

These skills and subtasks may be performed using a plethora of rehabilitation devices and equipment. However, the lower extremity rehabilitation robot ViGRR can have the added benefits of improved safety, fewer space requirements, detailed measurements of the patient's effort, real-time feedback, and quantifiable assessment. The underpinning foundation of these advantages is the versatility a robot can offer. Thus, a robust platform with the ability to implement a wide range of control strategies is a primary motivator for the ViGRR robot.

A high-level conceptual framework for the ViGRR controller is provided by the

flowchart in Figure 1.6. This controller concept involves three processes: Assessment, training, and control. In the assessment portion of the control loop, the controller must have the ability to measure the user's kinematics and joint torques by interpreting the sensors used in the device. A rehabilitation-focused controller algorithm and/or virtual environment will use the patient assessment measurements and a reference trajectory to generate a desired trajectory or interaction force command to the robot. This is the training or motor learning portion of the control scheme. The robot inner loop controller then matches the desired force or trajectory. This is the control portion of the flowchart that completes the ViGRR control concept.

Each area of this control scheme is dependant on a complete systems integration involving the modelling, design, construction and controller implementation of the robot. The investigation into this control algorithm and system development with respect to restoring gait and leg function is initiated by the development of a ViGRR prototype device presented in this thesis.

## 1.2.2 Prototype Concept

It was decided to build a limited prototype for ViGRR as a more practical approach to the development of the rehabilitation robot. By having fewer capital costs and a quicker initial development cycle, control strategies can be tested and design issues can be better understood on the prototype before moving to a full scale implementation. The prototype is envisioned as a device with fewer degrees of freedom but incorporating the desired controller capabilities of the full scale concept and the ability to be integrated on some level to a full scale implementation. The following high-level conceptual design requirements are based on the motivating factors in Section 1.1:

- The robot must be able to perform as a haptic display with visual and tactile feedback capable of delivering ground contact forces at various locations on the

foot;

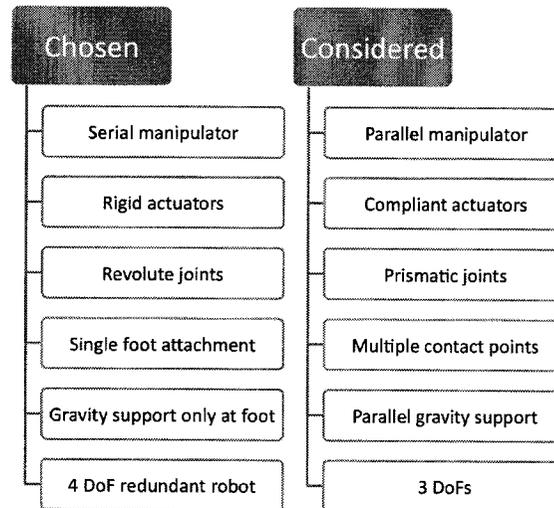
- A footplate with full leg support (counterbalancing the effect of gravity) is required for simulation of a variety of skills associated with the lower limbs (not simply treadmill-based gait training);
- A range of torso configurations are required, from supine to fully upright such that a wide range of therapies can be administered from simple exercises to full walking motion;
- The ability to measure joint kinematics and torques through sensors and an inverse dynamics model of the person.

The initial prototype design is limited to include only sagittal planar motion with a fixed pelvis location and actuation of one leg. A reclined posture is a simple initial configuration of the torso to evaluate controller strategies using the device. In the years ahead, additional DoFs may be added to the robot as funding and development time permits. A full scale design will include pelvic DoFs and actuation of both legs with a focus on human factors and the needs of therapists and patients in a clinical setting. A number of high-level preliminary design decisions regarding the configuration and format of the prototype robot are provided in the following section.

### **Preliminary Configuration Design**

The planar configuration of the simplified robot prototype is limited in complexity and degrees of freedom in order to have the design process run faster and provide time for further research and experimentation with control schemes. The focus of the configuration design is on requiring minimal actuator power, matching the required workspace, having few degrees of freedom, and the ability to control human-robot

interaction as a haptic device. Pertaining to the high-level design of the planar robot configuration, the following decisions and considerations are motivated in their accompanying paragraphs and refer to the chosen and considered design factors listed in Figure 1.7.



**Figure 1.7:** Configuration of ViGRR: Design considerations.

**Serial manipulator** A serial configuration with heavier linkages and more powerful motors generally costs more and is difficult to apply to a haptic device, since fast accelerations are required and the inertia of the robot should not be felt by the user. However, there is typically an increased possible workspace for a serial manipulator over parallel robotic devices. In application to the leg and walking, the accelerations from ground impact are not as significant compared to those experienced by hand in typical haptic devices, and a serial robot with powerful actuators is required to reflect the ground reaction forces to meet the needs of the ViGRR prototype concept.

**All revolute joint robot** A revolute joint planar robot configuration is an attractive design primarily because it is analogous to the leg. During walking, relatively little applied joint torques are required to support the body since the legs

are in an almost-singular configuration during the stance phase of gait. The joints also allow higher forward foot velocities with little applied torque during swing phase. A similar configuration of the robot would have these same advantages of the leg with the torso oriented vertically for upright walking. At different torso angles, it would be a simple matter to rotate the base link of a serial robot to suit the different orientations of the torso, in effect increasing the workspace of the robot. If a prismatic joint is included in the configuration, significant forces would be required to match the ground reaction forces from typical walking motion and work against gravity. A disadvantage of a revolute serial planar device is the increased gravitational loading at the end effector when the user is lying down. Both the weight of the robot and the person's leg must be supported against the effect of gravity in addition to supplying the desired load at the end effector. Parallel gravity support mechanisms may also be designed to counter this effect.

**Robot is offset to the side of leg** The planar robot is offset to the side of the user, with attachments to the leg intersecting the space where the leg can move in plane. This prevents collisions between the user and the robot and allows for a large workspace. Safety mechanisms can also be employed to mechanically release the leg from the robot in case of an emergency or instability in the controller without the robot injuring the user. The larger freedom of motion and workspace does, however, pose a real danger of moving the leg beyond its safe range of motion in the joints, potentially causing hyper extension or flexion and injury. These dangers are mitigated by robust and redundant safety mechanisms, such as adjustable impact bars, and microswitches. Further discussion of the safety mechanisms is given in Chapter 3.

**Single attachment at foot** The interface between the robot and the human is applied only at the foot with the serial linkage end effector. This is the approach taken by the HapticWalker discussed in Chapter 1 and deviates from the anthropomorphic design where the robot is attached to the leg at each limb segment with adjustable linkages. The end effector design allows for easier modelling of the human-robot interaction, requires fewer force sensors, and provides more freedom with the robot configuration. If the leg is attached to another part of the serial manipulator, either more degrees of freedom are required or the robot joints must line up exactly with the human leg joints. Aligning the robot and human leg joints is time consuming, requires adjustable link lengths, and misalignment may lead to injury. However, supporting only the foot means that the leg joints may experience larger forces when applying gravity compensation.

**No parallel gravity support devices** The primary issue with gravity compensation is that the leg model (discussed in Chapter 2) has a singularity when the knee joint is at zero degrees. When the leg approaches this configuration, the inverse Jacobian becomes singular and the calculated end effector forces and torques required to support the leg in that configuration approach infinity. In simulation, the knee joint angle is limited to ensure the singular configuration is not met and the Jacobian inverse is better conditioned numerically. This is not an ideal workaround as gravity will not be cancelled properly at small knee angles, and the forces in the leg joints may be too high with only the end effector supporting the leg at the foot. Gravity compensation may also be provided by a parallel active manipulator or passive support device. Some design challenges associated with such a device are the large range of motion required by the person's leg, the variability in the leg properties, and variable support at different torso angles. The interaction between a parallel device and the leg should also

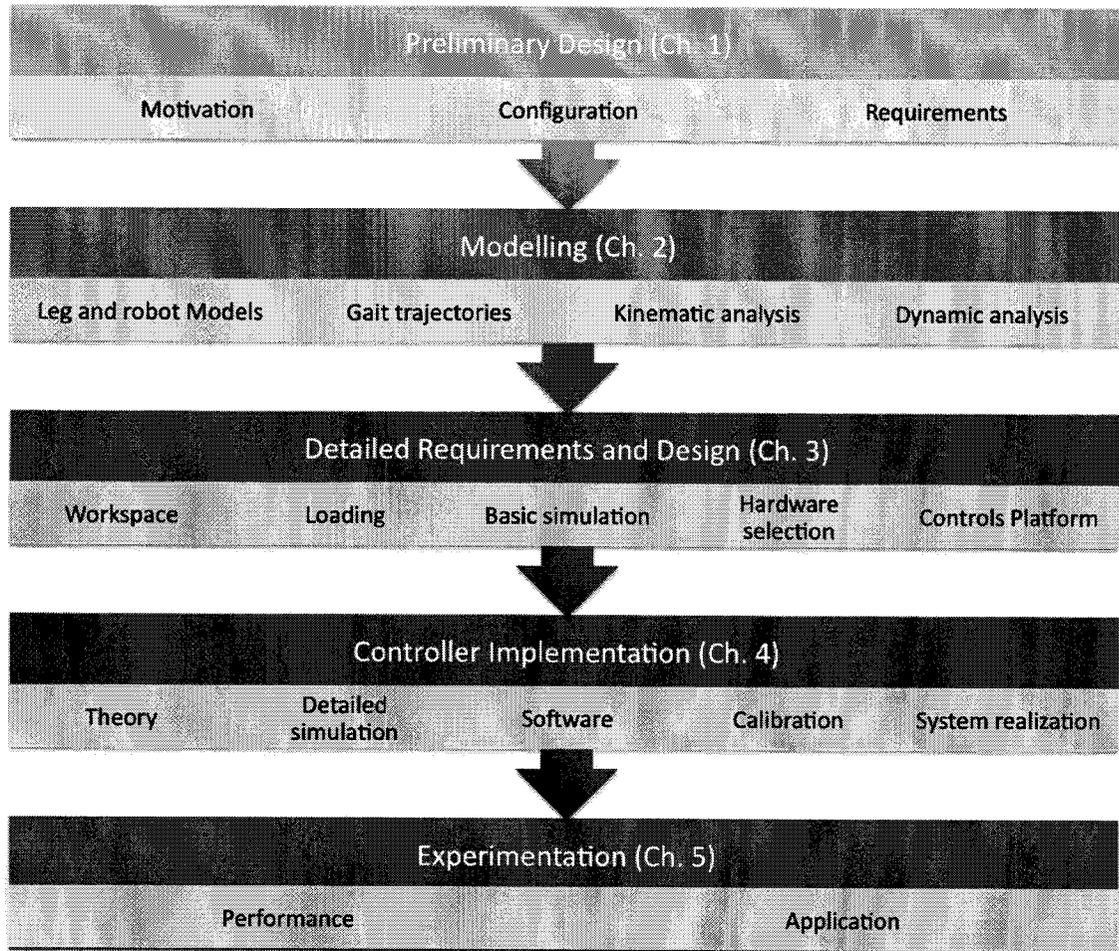
be modelled to properly assess the patient and provide a haptic control scheme or virtual environment, which potentially increases the complexity and hardware requirements of the device. A gravity compensation device is not included due to the associated cost increases, complexity, and development time.

**Four degrees of freedom** A fourth degree of freedom (4R manipulator) adds robustness to the system, despite drawbacks of added weight, cost and complexity. An advantage of redundancy is the potential for minimizing the joint velocities required to match the end effector velocity while avoiding joint limits, singular configurations or collisions. With a redundant robot, the extra degrees of freedom may also track a person's limb with a second contact point for support in potential future implementations of the ViGRR prototype. Despite the increased cost and complexity, the redundant 4 DoF configuration is implemented for its flexibility and research potential.

### 1.3 Contributions

In the context of the detailed design process for ViGRR, Figure 1.8 displays the flow of the thesis. The contributions range from the motivation for the ViGRR prototype concept to its realization as a functioning robotic research platform. The goal of the work presented in this thesis is to meet the the objective of developing a new rehabilitation robot outlined in Section 1.2. The following list of research contributions to the ViGRR project were made to meet this goal:

1. **Dynamic modelling and analysis of the robot and leg:** The leg and robot kinematics and dynamics were modelled. Anthropometry, average gait data, and passive joint torques were applied to the leg model. Optimal inverse kinematic schemes were applied to resolve redundancy in the robot.



**Figure 1.8:** Overview of thesis and design process.

2. **Design of the ViGRR prototype robot:** Workspace analysis, simulation and nonlinear optimization routines were used to design the robot.
3. **System Realization:** The actuation, sensor and control architecture apparatus were integrated and tested.
4. **Controller implementation and verification:**
  - (a) Performed kinematic and dynamic calibration.
  - (b) Implemented advanced nonlinear control techniques for joint and task-space control with smooth trajectory generators and redundancy resolution

schemes. An admittance force controller was also tested.

- (c) Demonstrated capabilities of ViGRR by testing a trajectory tracking application with an interactive force control task and visual feedback.

## 1.4 Outline

Figure 1.8 outlines the contributions provided in each Chapter. A brief description of each chapter in this thesis is presented as follows:

### **Chapter 2: Problem Formulation and Modelling**

This chapter derives the equations of motion for serial robots that are composed of only in-plane revolute joints. Dynamics of 3 DoF and 4 DoF manipulators are explicitly derived which serve to model the leg and ViGRR. The 3 DoF planar leg model is based on the dynamics of a serial manipulator, anthropomorphic data, and estimated passive joint torques. The robot model builds upon the dynamics of a 4 DoF planar robot and takes into account the joint transmission dynamics, friction model and end effector transformations.

### **Chapter 3: Robot Design and Apparatus**

The requirements and detailed design of the robot are described in this chapter. The requirements are based on the range of motion of the persons expected to use the device and their average gait trajectories and joint torques during gait. The associated safety, hardware and human factors considerations are also described. The detailed design of the ViGRR prototype involves the drive units selection, sensors, control platform, safety mechanisms and mechanical design.

## **Chapter 4: Controller Implementation**

In this chapter, the control mechanisms are provided for task space and joint level control, as well as generic impedance and admittance force controllers. Several joint-space and task space trajectory generators are also presented. The ViGRR controller features are presented, including a graphical user interface, Simulink block library and 3D visualization. A calibration procedure was performed to determine experimentally the kinematic and dynamic parameters of the robot.

## **Chapter 5: Experimentation**

Demonstration of the robot prototype capabilities is provided in this chapter. Joint-level and Cartesian space tracking errors are given in trajectory tracking mode. An admittance controller is also implemented for force feedback with user interaction. A final experiment testing the robot's ability to provide a task for the user with visual feedback is demonstrated with a circular trajectory-following challenge.

## **Chapter 6: Conclusions and Future Work**

A summary of the completed work and future directions toward implementing ViGRR as a full scale robot.

## Chapter 2

# Problem Formulation and Modelling

Derivation of the equations of motion of the robot and the leg model is a necessary step in determining the requirements of the robot, assessing the user's performance when using the device and control of the robot. Design of the ViGRR prototype is based on dynamic simulations involving the robot and leg with the specified requirements derived from average gait trajectories.

## 2.1 Euler-Lagrange Formulation of Dynamics

Since the leg model and robot are analogous in their properties and planar configuration, the derivation of the dynamics for a generic planar  $n$ -link revolute serial manipulator is applicable to both the leg and ViGRR model. The method of deriving the equations of motion was chosen to be the Euler-Lagrange formulation. An alternative recursive method of deriving these equations is the iterative Newton-Euler formulation [62, 63], however both formulations yield the same result. The Euler-Lagrange method was chosen for its relative simplicity and easier implementation with symbolic math software. The results from the following derivation will be applied to the leg and ViGRR dynamic models in subsequent sections of this chapter.

The Euler-Lagrange equations of motion are

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \phi_i, \quad (2.1)$$

where  $q_i$  is the generalized coordinate of the  $i$ th degree of freedom and  $\phi_i$  is an externally applied generalized force such that  $\phi_i q_i$  have dimensions of work.  $L$  is the Lagrangian

$$L = T - V \quad (2.2)$$

where  $L$  is the difference between the kinetic energy  $T$  and gravitational potential energy  $V$  of the system. The Euler-Lagrange formulation is based on assumptions for rigid body motion with holonomic constraints and its derivation from the principle of virtual work and d'Alembert's principle is detailed in [64].

The next step in deriving the equations of motion involves finding the Lagrangian for a generic serial robot. For an  $n$ -link robot, the kinetic energy and gravitational potential energy functions are described by

$$\begin{aligned} T &= \frac{1}{2} \dot{q}^T \left[ \sum_{i=1}^n m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) \mathcal{I}_i R_i(q)^T J_{\omega_i}(q) \right] \dot{q} \\ &= \frac{1}{2} \dot{q}^T D(q) \dot{q} \end{aligned} \quad (2.3)$$

and

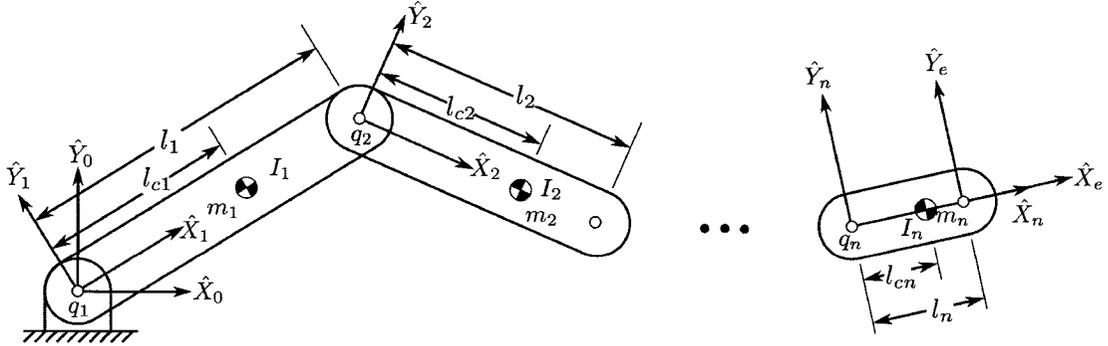
$$V = \sum_{i=1}^n m_i g_a^T r_{ci}. \quad (2.4)$$

In the above, the jacobian  $J_{v_i}(q)$  relates the generalized coordinate velocities to the translational velocities  $v_i$  of the centre of masses for each link.  $J_{\omega_i}(q)$  relates the generalized coordinate velocities to each link angular velocity  $\omega_i$  as in

$$v_i = J_{v_i}(q) \dot{q} \quad \omega_i = J_{\omega_i}(q) \dot{q}. \quad (2.5)$$

The rotation matrix  $R_i(q)$  is the orientation from the coordinate frame of link  $i$  to

the fixed global frame and  $\mathcal{I}_i$  is the constant inertial tensor of link  $i$  relative to its coordinate frame at its centre of mass. In (2.4),  $r_{ci}$  is the position of the centre of mass for link  $i$  relative to the global frame and  $g_a$  is the gravity vector.



**Figure 2.1:** Frame, length and mass definitions for a planar n-revolute joint linkage system.

The treatment of the equations of motion thus far can apply to any serial robot manipulator. For the application of modelling the planar leg or the ViGRR robot, we can define the Jacobians and simplify equations 2.3 and 2.4 for any planar revolute serial manipulator with  $n$  links. The planar robot parameters are defined by Figure 2.1, where:

- $\hat{X}_i, \hat{Y}_i$ , and  $\hat{Z}_i$  denote the  $i$ th link local coordinate frame basis vectors fixed to each link and follows the right-hand coordinate system convention.  $i = 0$  is the fixed global coordinate frame and  $e$  is the end effector frame fixed to link  $n$ .
- $q_i$  is the angle of joint  $i$  which is the generalized coordinate from (2.1) and follows Denavit-Hartenberg conventions for revolute joints [62, 63]. This is defined as the angle from  $\hat{X}_{i-1}$  to  $\hat{X}_i$  measured about  $\hat{Z}_i$ .
- $l_i$  is the length of link  $i$  along the  $\hat{X}_i$  axis
- $m_i$  is the link's mass
- $l_{ci}$  is the centre of mass location for link  $i$  along the  $\hat{X}_i$  axis

- $I_i$  is the mass moment of inertia of link  $i$  through the centre of mass about the  $\hat{Z}_i$  axis.
- $\tau_i$  are the external torques applied at each joint, which are the generalized forces  $\phi_i$  in (2.1).

The Jacobians  $J_{v_i}$  and  $J_{\omega_i}$  used to calculate the kinetic energy term (2.3) can be generated using

$$J_{v_i}(q) = \begin{bmatrix} j_{11} & j_{12} & \cdots & j_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ j_{31} & j_{32} & \cdots & j_{3n} \end{bmatrix}, \quad (2.6)$$

where

$$j_{1k} = \begin{cases} -\sum_{m=k}^{i-1} \{l_m \sin(\sum_{j=1}^m q_j)\} - l_{ci} \sin(\sum_{j=1}^i q_j) & \text{if } k < i \\ -l_{ci} \sin(\sum_{j=1}^i q_j) & \text{if } k = i \\ 0 & \text{if } k > i \end{cases}$$

$$j_{2k} = \begin{cases} \sum_{m=k}^{i-1} \{l_m \cos(\sum_{j=1}^m q_j)\} + l_{ci} \cos(\sum_{j=1}^i q_j) & \text{if } k < i \\ l_{ci} \cos(\sum_{j=1}^i q_j) & \text{if } k = i \\ 0 & \text{if } k > i \end{cases}$$

$$j_{3k} = 0$$

and

$$J_{\omega_i}(q) = \begin{bmatrix} j_{11} & j_{12} & \cdots & j_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ j_{31} & j_{32} & \cdots & j_{3n} \end{bmatrix}, \quad (2.7)$$

where

$$j_{1k} = 0$$

$$j_{2k} = 0$$

$$j_{3k} = \begin{cases} 1 & \text{if } k \leq i \\ 0 & \text{if } k > i \end{cases},$$

respectively. For a planar revolute joint manipulator, there is no angular velocity out of the  $x$ - $y$  plane so only the inertia at the centre of mass about the  $z$  axis ( $I_i$ ) for each

link needs to be considered. In this case, the angular kinetic energy term reduces to

$$J_{\omega_i}(q)^T R_i(q) \mathcal{I}_i R_i(q)^T J_{\omega_i}(q) = I_i J_{\omega_i}(q)^T J_{\omega_i}(q). \quad (2.8)$$

If we set  $g_a = [0 \ g \ 0]^T$ , where  $g$  is the acceleration due to gravity, the potential energy function reduces to

$$V = g \sum_{k=1}^n m_k \left\{ \left( \sum_{m=1}^{k-1} l_m \sin \left( \sum_{j=1}^m q_j \right) \right) + l_{ck} \sin \left( \sum_{j=1}^k q_j \right) \right\}. \quad (2.9)$$

The above derivation can now be applied to the leg model and the ViGRR robot in the following sections.

## 2.2 Leg Model

Modelling of the human leg is an essential component of gait analysis and assessment. There are many techniques for modelling the human musculoskeletal system that vary greatly in complexity. The modelling and analysis of a person's gait depends on its purpose and application. In a clinical setting, simplified dynamic and inverted pendulum models can glean insights into gait disorder pathologies [65]. For a basic gait assessment and analysis, a 2-dimensional model may be sufficient with goniometers and force plates to measure joint trajectories and ground reaction forces [21] or a vision-based gait analysis with inverse dynamics, such as [66]. In a research environment, a complex 3D model with multiple redundant muscles, bone structures, dynamic muscle models, many degrees of freedom and an accurate motion capture system may provide insights into the nature gait biomechanics and impairments [49, 67, 68]. It was chosen to introduce a simple planar model of the leg for the prototype implementation of ViGRR and the following discussion presents its motivation, formulation and equations of the model's kinematics and dynamics.

The ability to determine the joint trajectories and applied torques during operation of the gait rehabilitation robot can be beneficial for the assessment of the patient’s performance. The position and forces acting on the leg can also be useful with a haptic device in a virtual environment, where the desired trajectory of the robot depends on the forces applied to the leg and the configuration and dynamics of the leg itself. The ViGRR gait rehabilitation robot is a planar device and, accordingly, the leg is modeled as a planar linkage system with three revolute joints, similar to simple models used for 2-dimensional gait analysis [66, 69]. The leg model is analogous to a serial 3R robotic arm and the following process of deriving the equations of motion for the model are well-established methods applied in robotics.

The limitations and simplifications of this model warrant caution in assuming the validity of the model. A discussion of the model limitations, and the verification and validation of the leg model are provided in Section 2.2.6 of this chapter.

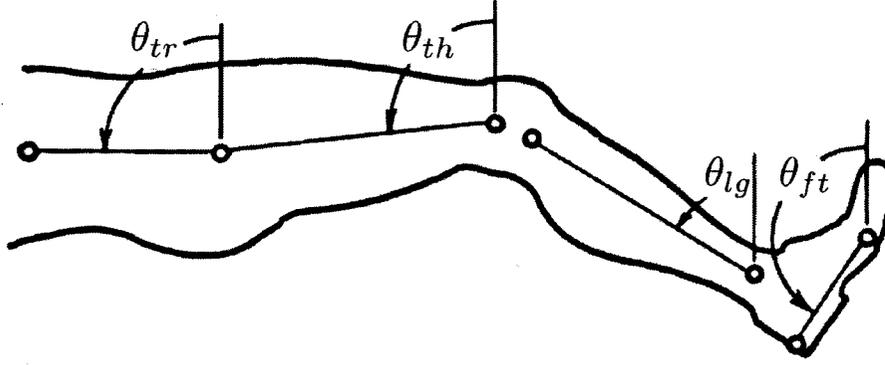
### 2.2.1 Leg Model Definitions and Dynamics

For the leg model presented in this thesis, biomechanical angle conventions of the knee and hip joints define flexion as positive, extension as negative and zero when the two limbs about a joint are aligned as given in [69]. For the ankle joint, plantarflexion is negative and dorsiflexion is positive. Table 2.1 lists the equations defining the angle of the hip, knee, and ankle joints. These equations are a function of the angles of the trunk, thigh, leg, and foot limbs relative to the ground, as shown in Figure 2.2 [69]. For the applied moments about the leg joints, biomechanical conventions define positive torque in flexion and negative in extension. For the ankle joint, torque in plantarflexion is positive and dorsiflexion is negative [69].

The leg model for ViGRR was chosen to follow a different set of conventions that correspond to robotics theory instead of biomechanical conventions. This allows for easier simulation and design and it maintains consistency with the ViGRR robot

**Table 2.1:** Clinical joint angle definitions.

Joint	Angle Convention Equations
Hip	$\theta_{hip} = \theta_{th} - \theta_{tr}$
Knee	$\theta_{knee} = \theta_{th} - \theta_{lg}$
Ankle	$\theta_{ankle} = \theta_{ft} - \theta_{lg} - 90^\circ$

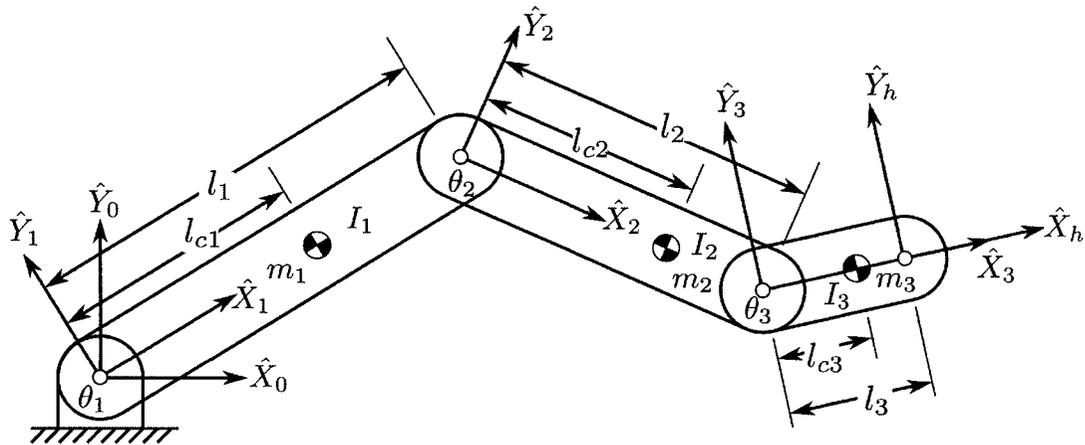
**Figure 2.2:** Limb segment angles and planar coordinate conventions for the leg.

model and the leg model. The user is also seated in a reclined position, with the trunk having different fixed angles relative to the fixed global frame.

In the ViGRR leg model, each leg segment (thigh, lower leg, and foot) is assumed to be a rigid body with the same properties as those defined by a planar revolute manipulator with  $n = 3$  degrees of freedom and follows the conventions defined in Section 2.1. Figure 2.3 shows the 3R manipulator leg model parameters. The revolute joint angles correspond to the hip, knee and ankle joints where  $q_i = \theta_i$  and the coordinate frame at the end of the human foot is denoted with subscript  $h$ .

A relationship between the leg geometry, biomechanical joint angle conventions and the 3R model can now be expressly defined. The following list of relationships between the two conventions are summarized in Table 2.2:

- The rehabilitation device concept involves a user with a fixed trunk orientation



**Figure 2.3:** Frame, length and mass definitions for a the 3R manipulator leg model.

and hip position. The orientation of the fixed trunk is given as the angle  $\theta_{tr}$  which is positive counter clockwise from the  $\hat{Y}_0$  axis and can have a range from 0 degrees (where the user is completely vertical) to 90 degrees (where the user is lying flat).

- The thigh limb segment corresponds to the 1st link length which is the distance from the greater trochanter (hip joint) to the femoral condyles (knee joint) and is assumed to pass through the limb segment's centre of mass. The first joint angle of the ViGRR leg model  $\theta_1$  is related to the hip angle  $\theta_{hip}$  offset by the trunk angle  $\theta_{tr}$ .
- The lower leg corresponds to the 2nd link length which is the distance from the femoral condyles (knee joint) to medial malleolus (ankle joint) and is assumed to pass through the its centre of mass. The knee angle  $\theta_{knee}$  is positive in flexion, which corresponds to a negative second joint angle  $\theta_2$ .
- The foot corresponds to the 3rd link length which is the distance between the ankle joint and the head of the second metatarsal and is assumed to pass through the foot's centre of mass. The ankle angle  $\theta_{ankle}$  is related to the third joint

angle  $\theta_3$  by geometry from anthropometric data of the foot (Section 2.2.4).

- In robotics conventions, the torque is positive about the  $z$ -axis of the joint following the right hand rule. The ViGRR leg model hip torque  $\tau_{h_1}$  is equal to the biomechanical joint moment  $\tau_{hip}$ . The ViGRR model knee torque  $\tau_{h_2}$  and ankle torque  $\tau_{h_3}$  have a opposite signs to the biomechanical joint moment conventions  $\tau_{knee}$  and  $\tau_{ankle}$ .

**Table 2.2:** Joint angle conversion from anatomical convention to 3R manipulator.

Joint	Angle and torque conversion equations.
1	$\theta_1 = \theta_{hip} + \theta_{tr} - \pi/2$ (rad)
2	$\theta_2 = -\theta_{knee}$ (rad)
3	$\theta_3 = \theta_{ankle} - 1.0158$ (rad)
1	$\tau_{h_1} = \tau_{hip}$ (Nm)
2	$\tau_{h_2} = -\tau_{knee}$ (Nm)
3	$\tau_{h_3} = -\tau_{ankle}$ (Nm)

Building upon the equations of motion derived in Section 2.1, the dynamics of the leg model are presented in the following section.

### 3R Dynamics of the Leg

The equations of motion for the leg model are derived by the Euler-Lagrange formulation of dynamics for a 3 revolute joint (3R) planar linkage system. Evaluating equations (2.1) to (2.9) for the 3R leg model and simplifying yields

$$\tau_h = M(\theta)\ddot{\theta} + B(\theta)\dot{\theta}\dot{\theta} + C(\theta)\dot{\theta}^2 + g(\theta), \quad (2.10)$$

where the net joint torques of the human leg model are  $\tau_h = [\tau_{h_1}, \tau_{h_2}, \tau_{h_3}]^T$  and the angular vectors are  $\theta = [\theta_1, \theta_2, \theta_3]^T$ ,  $\dot{\theta}^2 = [\dot{\theta}_1^2, \dot{\theta}_2^2, \dot{\theta}_3^2]^T$ , and  $\dot{\theta}\dot{\theta} = [\dot{\theta}_1\dot{\theta}_2, \dot{\theta}_1\dot{\theta}_3, \dot{\theta}_2\dot{\theta}_3]^T$ .

The matrices  $M(\theta)$ ,  $B(\theta)$ ,  $C(\theta)$ , and vector  $g(\theta)$  are:

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix}, \quad (2.11)$$

where

$$\begin{aligned} m_{11} &= m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) \\ &\quad + m_3 (l_1^2 + l_2^2 + l_{c3}^2 + 2l_1 l_2 \cos \theta_2 + 2l_1 l_{c3} \cos(\theta_2 + \theta_3) \\ &\quad + 2l_2 l_{c3} \cos \theta_3) + I_1 + I_2 + I_3 \\ m_{22} &= m_2 l_{c2}^2 + m_3 (l_2^2 + l_{c3}^2 + 2l_2 l_{c3} \cos \theta_3) + I_2 + I_3 \\ m_{33} &= m_3 l_{c3}^2 + I_3 \\ m_{12} &= m_2 (l_1 l_{c2} \cos \theta_2 + l_{c2}^2) + m_3 (l_2^2 + l_{c3}^2 + l_1 l_2 \cos \theta_2 + l_1 l_{c3} \cos(\theta_2 + \theta_3) \\ &\quad + 2l_2 l_{c3} \cos \theta_3) + I_2 + I_3 \\ m_{13} &= m_3 (l_1 l_{c3} \cos(\theta_2 + \theta_3) + l_2 l_{c3} \cos \theta_3 + l_{c3}^2) + I_3 \\ m_{23} &= m_3 (l_2 l_{c3} \cos \theta_3 + l_{c3}^2) + I_3; \end{aligned}$$

$$B(\theta) = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}, \quad (2.12)$$

where

$$\begin{aligned} b_{11} &= -2(m_2 l_1 l_{c2} \sin \theta_2 + m_3 (l_1 l_2 \sin \theta_2 + l_1 l_{c3} \sin(\theta_2 + \theta_3))) \\ b_{21} &= 0 \\ b_{31} &= 2m_3 l_2 l_{c3} \sin \theta_3 \\ b_{12} &= -2m_3 (l_1 l_{c3} \sin(\theta_2 + \theta_3) + l_2 l_{c3} \sin \theta_3) \\ b_{22} &= -2m_3 l_2 l_{c3} \sin \theta_3 \\ b_{32} &= 0 \\ b_{13} &= -2m_3 (l_1 l_{c3} \sin(\theta_2 + \theta_3) + l_2 l_{c3} \sin \theta_3) \\ b_{23} &= -2m_3 l_2 l_{c3} \sin \theta_3 \\ b_{33} &= 0; \end{aligned}$$

$$C(\theta) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \quad (2.13)$$

where

$$\begin{aligned}
c_{11} &= 0 \\
c_{21} &= m_2 l_1 l_{c2} \sin \theta_2 + m_3 (l_1 l_2 \sin \theta_2 + l_1 l_{c3} \sin(\theta_2 + \theta_3)) \\
c_{31} &= m_3 (l_1 l_{c3} \sin(\theta_2 + \theta_3) + l_2 l_{c3} \sin \theta_3) \\
c_{12} &= -(m_2 l_1 l_{c2} \sin \theta_2 + m_3 (l_1 l_2 \sin \theta_2 + l_1 l_{c3} \sin(\theta_2 + \theta_3))) \\
c_{22} &= 0 \\
c_{32} &= m_3 l_2 l_{c3} \sin \theta_3 \\
c_{13} &= -(m_3 (l_1 l_{c3} \sin(\theta_2 + \theta_3) + l_2 l_{c3} \sin \theta_3)) \\
c_{23} &= -m_3 l_2 l_{c3} \sin \theta_3 \\
c_{33} &= 0;
\end{aligned}$$

$$g(\theta) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}, \quad (2.14)$$

where

$$\begin{aligned}
g_1 &= m_1 g l_{C1} \cos \theta_1 + m_2 g (l_1 \cos \theta_1 + l_{C2} \cos(\theta_1 + \theta_2)) \\
&\quad + m_3 g (l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_{C3} \cos(\theta_1 + \theta_2 + \theta_3)) \\
g_2 &= m_2 g (l_{C2} \cos(\theta_1 + \theta_2)) \\
&\quad + m_3 g (l_2 \cos(\theta_1 + \theta_2) + l_{C3} \cos(\theta_1 + \theta_2 + \theta_3)) \\
g_3 &= m_3 g l_{C3} \cos(\theta_1 + \theta_2 + \theta_3).
\end{aligned}$$

### 2.2.2 Leg Kinematics

The leg kinematics provide relationships between the task space (Cartesian space) and the leg joint angles. When using the ViGRR prototype robot, a person's foot will be attached to the robot end effector. Knowing how the leg kinematics and joint angles are related to the pose of the robot is essential in determining how to generate desired trajectories relative to the leg as well as avoiding configurations that may injure the user. Inverse kinematics may also be applied to get an estimate of the leg joint angles. These kinematic transformations are used for simulation, generating workspace requirements, a Cartesian space trajectory generator based on the leg joint motion such as normal gait trajectories, or assessing the user's leg kinematics during

operation of the robot.

### Forward and Inverse Kinematics

The forward kinematics describe the leg model foot (end effector) pose  $X_h$  based on the leg joint angles  $\theta$ . The pose is defined as the planar position coordinates  $(x_h, y_h)$  and the foot segment orientation with respect to the global coordinate frame by angle  $\psi_h$ . The angle  $\psi_h$  is the angle from  $\hat{X}_0$  to  $\hat{X}_h$  about the  $\hat{Z}_0$  axis. The transformation from joint angles to end effector position and orientation is given by

$$X_h = \begin{Bmatrix} x_h \\ y_h \\ \psi_h \end{Bmatrix} = \begin{Bmatrix} l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \theta_1 + \theta_2 + \theta_3 \end{Bmatrix}. \quad (2.15)$$

The inverse transformation from the foot end effector position and orientation to leg joint angles is

$$\theta = \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} \text{atan2}(y_3, x_3) \pm \cos^{-1} \left( \frac{x_3^2 + y_3^2 + l_1^2 - l_2^2}{2l_1 \sqrt{x_3^2 + y_3^2}} \right) \\ \pm \cos^{-1} \left( \frac{x_3^2 + y_3^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \\ \psi_f - \theta_2 - \theta_1 \end{Bmatrix}, \quad (2.16)$$

where

$$\begin{aligned} x_3 &= x_h - l_3 \cos(\psi_h) \\ y_3 &= y_h - l_3 \sin(\psi_h). \end{aligned}$$

### Differential Kinematics and Statics

The velocity of the foot end effector can be calculated based on the joint velocities by differentiating (2.15) with respect to time, in order to obtain

$$\dot{X}_h = J_h \dot{\theta}, \quad (2.17)$$

where  $J_h$  is the Jacobian matrix defined by

$$J_h = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix}, \quad (2.18)$$

where

$$\begin{aligned} j_{11} &= -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ j_{21} &= l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ j_{31} &= 1 \\ j_{12} &= -l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ j_{22} &= l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ j_{32} &= 1 \\ j_{13} &= -l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ j_{23} &= l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ j_{33} &= 1. \end{aligned}$$

A generic treatment of the Jacobian derivation for manipulator robotics and a study of differential kinematics and statics can be found in [62, 63]. The relationship between end effector and joint accelerations is derived by differentiating (2.17) with respect to time to obtain

$$\ddot{X}_h = \dot{J}_h \dot{\theta} + J_h \ddot{\theta}. \quad (2.19)$$

For the ViGRR leg model, the Jacobian is a square matrix and can be inverted to calculate the joint velocities and accelerations based on the end effector motion, except for the singular configuration when the second joint angle is equal to zero. The following equations describe the inverse differential kinematics calculations:

$$\dot{\theta} = J_h^{-1}(\theta) \dot{X}_h; \quad (2.20)$$

$$\ddot{\theta} = J_h^{-1}(\ddot{X}_h - \dot{J}_h \dot{\theta}). \quad (2.21)$$

The Jacobian transpose relates the planar static forces and torques at the foot

end-effector  $f_h = [f_{h_x}, f_{h_y}, \tau_{h_z}]^T$  in (N, N, Nm) to the torques applied at the joints  $\tau_h$  in (Nm) by

$$\tau_h = J_h^T f_h. \quad (2.22)$$

The forces  $f_{h_x}$  and  $f_{h_y}$  denote the forces relative to the base frame  $\hat{X}_0$  and  $\hat{Y}_0$  directions.  $\tau_{h_z}$  is the torque about the  $\hat{Z}_0$  axis.

### 2.2.3 Passive Joint Torques

When approaching the joint limits of the leg, an involuntary passive restoring torque is applied by soft tissues stretching over the joint [70, 71]. The restoring torque for each joint is not just a function of the joint angle itself, but of all three leg joint angles. A simple nonlinear elastic model for calculating these passive torques for the hip, knee and ankle [70] is given by

$$\begin{aligned} \tau_{p_1} &= \exp(1.4655 - 0.0034\theta_{knee} - 0.0750\theta_{hip}) \\ &\quad - \exp(1.3403 - 0.0226\theta_{knee} + 0.0305\theta_{hip}) + 8.072 \\ \tau_{p_2} &= -(\exp(1.800 - 0.0460\theta_{ankle} - 0.0352\theta_{knee} + 0.0217\theta_{hip}) \\ &\quad - \exp(-3.971 - 0.0004\theta_{ankle} + 0.0495\theta_{knee} - 0.0128\theta_{hip}) \\ &\quad - 4.820 + \exp(2.220 - 0.150\theta_{knee})) \\ \tau_{p_3} &= -(\exp(2.1016 - 0.0843\theta_{ankle} - 0.0176\theta_{knee}) \\ &\quad - \exp(-7.9763 + 0.1949\theta_{ankle} + 0.0008\theta_{knee}) - 1.792). \end{aligned} \quad (2.23)$$

These resulting torques are in Nm and follow the ViGRR leg model conventions. The passive joint torques are highly variable from person to person and average model parameters cannot be relied upon for accuracy [71], but can be useful for design, simulation and rough estimates of a person's applied joint moments. The passive leg model is discussed further in Section 3.1.1 where plots are shown and it is used to calculate the workspace bounds for the robot.

In summary, the dynamics of the leg model can be modified to include these

passive joint torques  $\tau_{pass} = [\tau_{p1}, \tau_{p2}, \tau_{p3}]^T$  and the torques supporting loading at the foot  $f_h$  to generate the final equations of motion for the leg model:

$$\tau_h = M(\theta)\ddot{\theta} + B(\theta)\dot{\theta}\dot{\theta} + C(\theta)\dot{\theta}^2 + g(\theta) + J_h^T f_h - \tau_{pass}. \quad (2.24)$$

## 2.2.4 Anthropometry

We now have a working model of the leg for use with ViGRR. However, the parameters of the leg segments such as lengths, and mass properties must be assumed or measured to perform the design and simulation of the user with the rehabilitation robot. When an actual subject is using the robot, some of these parameters are easily measured such as the limb segment lengths, but the inertial and mass properties are often difficult to identify and there is expected error in the model based on the simplified definitions in Section 2.2.1. The dynamic parameters can be estimated based on anthropometric studies of average human masses, heights and proportions. A significant number of anthropometric studies were performed in the 1950s and 1960s using cadavers and tests on living subjects between the ages of 20 and 70 years [72–74]. Anthropometric data is used to estimate the mass properties of an individual in the absence of measurements specific to the subject and determine their dynamics based on motion analysis and ground force sensors [66]. For the gait rehabilitation robot ViGRR, anthropometric data is used to model lower extremities for determining workspace and loading requirements.

### Limb Segment Lengths

The lengths of each body segment are provided in Table 2.3 as a fraction of the total body height (stature) of an individual [72]. These segment proportions provide approximations when measured segment lengths of the individual are not available [66].

**Table 2.3:** Length of each leg segment  $l_i$  as a fraction of body height  $H$ .

Body Part	Segment Length Definition	Length $l_i$
Thigh	Greater trochanter to femoral condyles	$l_1 = 0.245H$
Lower leg	Femoral condyles to medial malleolus	$l_2 = 0.246H$
	Sole to medial malleolus	0.039H
Foot	Heel to toe	0.152H
	Medial malleolus to 2nd metatarsal	$l_3 = 0.074H$

### Masses and Mass Moments of Inertia

The center of mass locations and masses of the of human body segments are summarized in Table 2.4. These values were determined by measurements made on human cadavers and living subjects and are summarized by [66] for the purposes of gait analysis.

**Table 2.4:** Mass of each leg segment as a fraction of body mass  $M$  and the centre of mass (CoM) location as a fraction of the segment length  $l_i$ .

Body Part	Distance to CoM Definition	CoM	Mass
Thigh	Greater trochanter to CoM	$l_{C1} = 0.433l_1$	$m_1 = 0.100M$
Lower leg	Femoral condyles to CoM	$l_{C2} = 0.433l_2$	$m_2 = 0.0465M$
	Medial malleolus to CoM	$l_{C3} = 0.249l_3$	
Foot	CoM to heel	$0.438l_3$	$m_3 = 0.0145M$
	CoM to toe	$0.594l_3$	

Mass moment of inertia of each leg segment  $i$  at its centre of gravity can be determined using

$$I_i = m_i r_{g_i}^2, \quad (2.25)$$

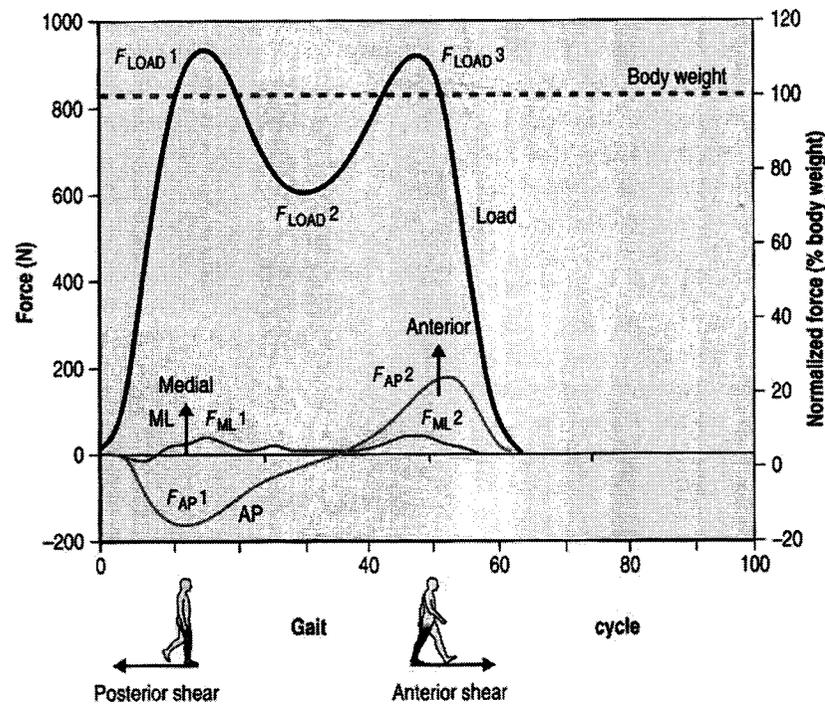
where  $m_i$  is the mass of the limb segment and  $r_{g_i}$  is the radius of gyration of link segment  $i$ . The values of  $r_{g_i}$  can be expressed as a fraction of the limb segment length  $l_i$  which are provided in Table 2.5.

**Table 2.5:** Radius of gyration  $r_{g_i}$  for each mass segment of leg as a factor of segment length  $l_i$ .

Body Part	Segment Length Definition	Radius of Gyration $r_{g_i}$
Thigh	Greater trochanter to femoral condyles	$r_{g_1} = 0.323l_1$
Lower leg	Femoral condyles to medial malleolus	$r_{g_2} = 0.302l_2$
Foot	Medial malleolus to CoM	$r_{g_3} = 0.477l_3$

## 2.2.5 Average Gait Trajectories

As a gait and lower extremity rehabilitation device, the ability of ViGRR to perform a gait trajectory with a human occupant is essential. Average measurements of gait kinematics and kinetics are available from literature and can be useful for deriving the workspace trajectory and loading requirements for the robot design process. This section introduces 2D gait trajectories used with the ViGRR human model.



**Figure 2.4:** Gait progression and ground reaction forces [21].

Human gait is characterized by a periodic motion where a single gait cycle begins

and ends at the heel strike of one leg. Figure 2.4 shows the gait progression by inspecting one leg throughout the motion and the corresponding ground reaction forces. The first 60 percent of the gait cycle is the stance phase, where the foot is planted on the ground from heel strike to toe off. From toe off to heel strike, the gait is in its swing phase and the opposite leg begins its cycle with a heel strike at about 30 percent of the gait cycle.

A person's walking speed, or cadence  $N_{cad}$ , is defined by the number of steps per minute. Since two steps are taken every gait cycle, the time during the gait cycle is related to the cadence and percent of gait cycle by

$$t_g = 1.2 \frac{P_{cycle}}{N_{cad}}, \quad (2.26)$$

where  $t_g$  is the time during the gait cycle in seconds,  $N_{cad}$  is the walking cadence in steps per minute and  $P_{cycle}$  is the percent of the gait cycle from 0 to 100.

Quantifying the gait kinematics and kinetics over the gait cycle can be accomplished with a gait analysis clinic. However, average gait analysis data for healthy individuals has been gathered by many researchers over the years and further testing and analysis would be beyond the scope of the thesis, time consuming, and redundant given the gait trajectories published in [66, 69]. For use with the ViGRR human leg model, average experimental sagittal plane gait trajectories of healthy adults were obtained from [69].

Figures 2.5, 2.6, and 2.7 show the raw data as circle markers and the fitted spline fit, which is continuous at the end values. The broken line represents the bounds of one standard deviation from the mean for the collected data. The following plots also follow the biomechanical conventions for joint angles and torques.

For the joint kinematics, the angles are given as a function of percent gait cycle  $P_{cycle}$ . The angular positions, velocities and accelerations of the leg joints are calculated from a smoothed cubic spline fit of the data. The gait cubic splines have 3rd

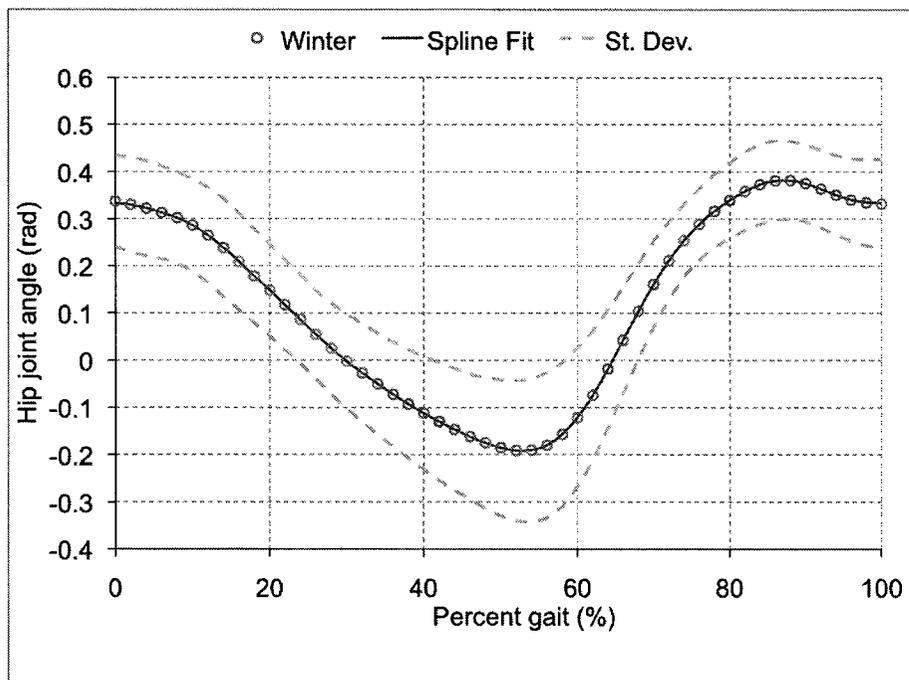


Figure 2.5: Hip Joint angles vs. percent gait cycle.

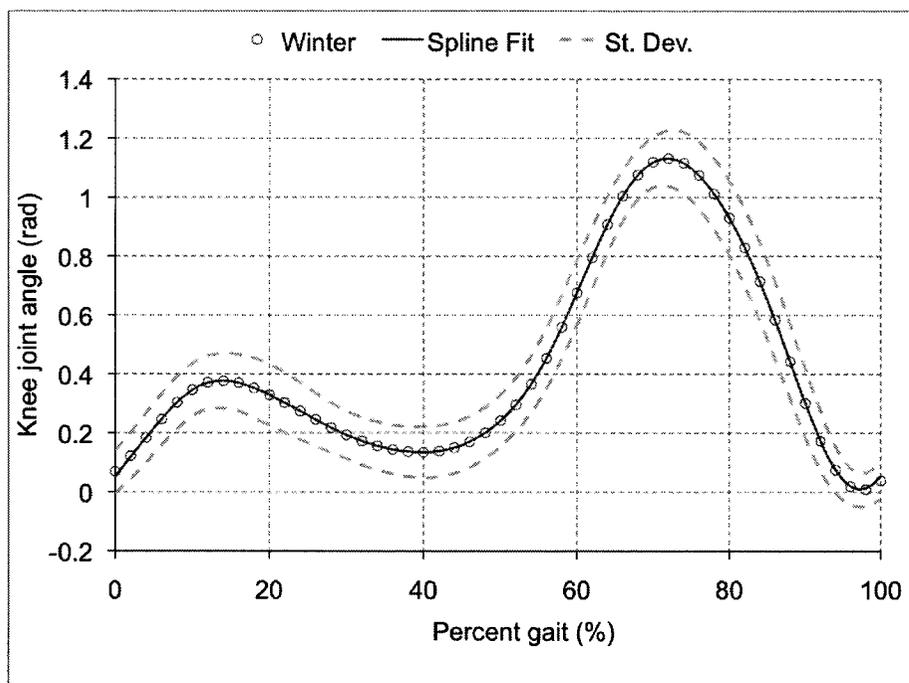
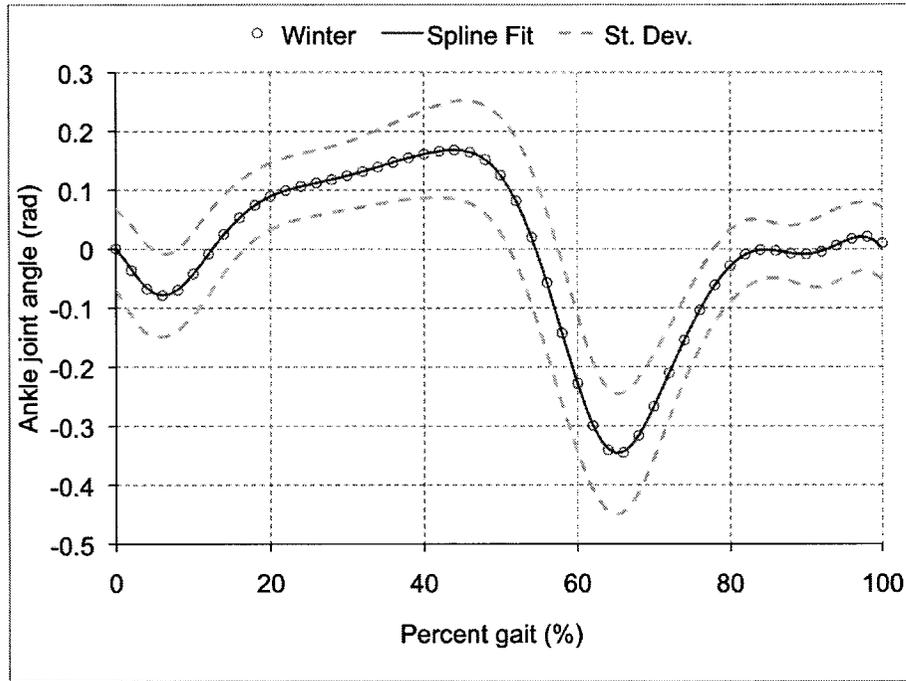


Figure 2.6: Knee Joint angles vs. percent gait cycle.



**Figure 2.7:** Ankle Joint angles vs. percent gait cycle.

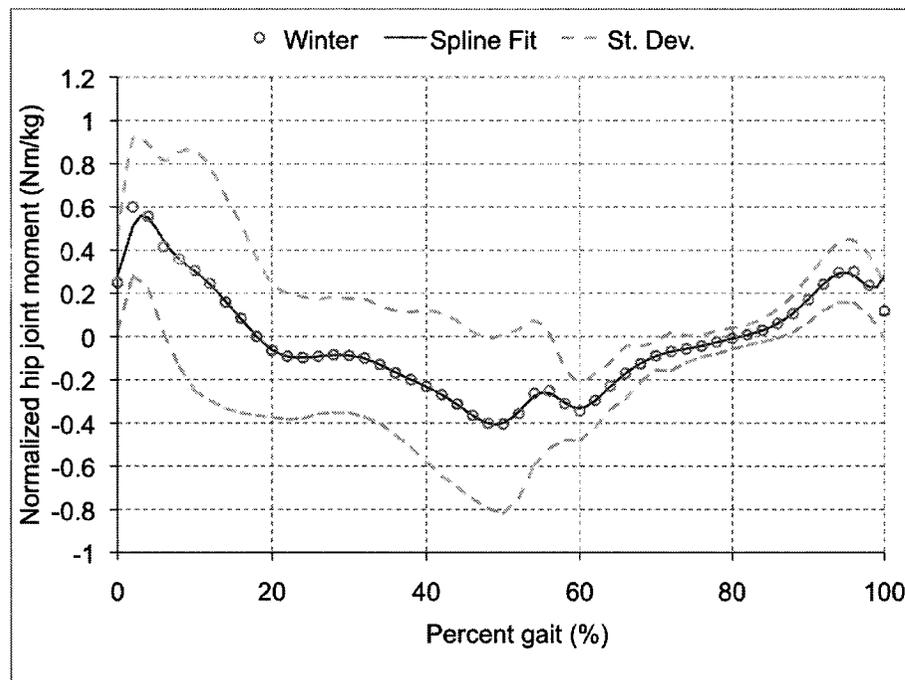
order polynomials with different coefficients specified every 2 percent gait. At each spline polynomial, the output joint angle positions, velocities and accelerations are calculated from

$$\begin{aligned}
 \theta(P_{cycle}) &= s_1(P_{cycle} - P_{break_i})^3 + s_2(P_{cycle} - P_{break_i})^2 + s_3(P_{cycle} - P_{break_i}) + s_4 \\
 \dot{\theta}(P_{cycle}, N_{cad}) &= \frac{N_{cad}}{1.2} (3s_1(P_{cycle} - P_{break_i})^2 + 2s_2(P_{cycle} - P_{break_i}) + s_3) \\
 \ddot{\theta}(P_{cycle}, N_{cad}) &= \left(\frac{N_{cad}}{1.2}\right)^2 (6s_1(P_{cycle} - P_{break_i}) + 2s_2) ,
 \end{aligned} \tag{2.27}$$

where  $P_{break_i}$  is the  $i$ th element of the percent cycle spline break vector  $P_{break} = [0, 2, \dots, 98, 100]$ , and  $P_{break_i} < P_{cycle} < P_{break_{i+1}}$ . The spline coefficients  $s_k$  for  $k = 1$  to 4 are specified between each break value. The spline fit provides continuous position, velocity and acceleration trajectories and allows for use of the gait trajectory in a controller or simulation with any specified walking speed and resolution.

The ground reactions forces (GRF) over were also obtained from [69], which are normalized to the mass of the individual. The forces in the horizontal (positive forward) and vertical (positive upward) directions are plotted against percent gait in Figures 2.11 and 2.12 with the corresponding spline fit. The GRF can be calculated as a function of percent gait cycle with its corresponding spline fit.

The applied joint torques of the individual normalized to their body mass [69] and corresponding spline fit are plotted in Figures 2.8, 2.9, and 2.10 for the hip, knee, and ankle joints, respectively. The joint moments can be calculated as a function of percent gait cycle with its corresponding spline fit.



**Figure 2.8:** Normalized hip joint moments vs. percent gait cycle.

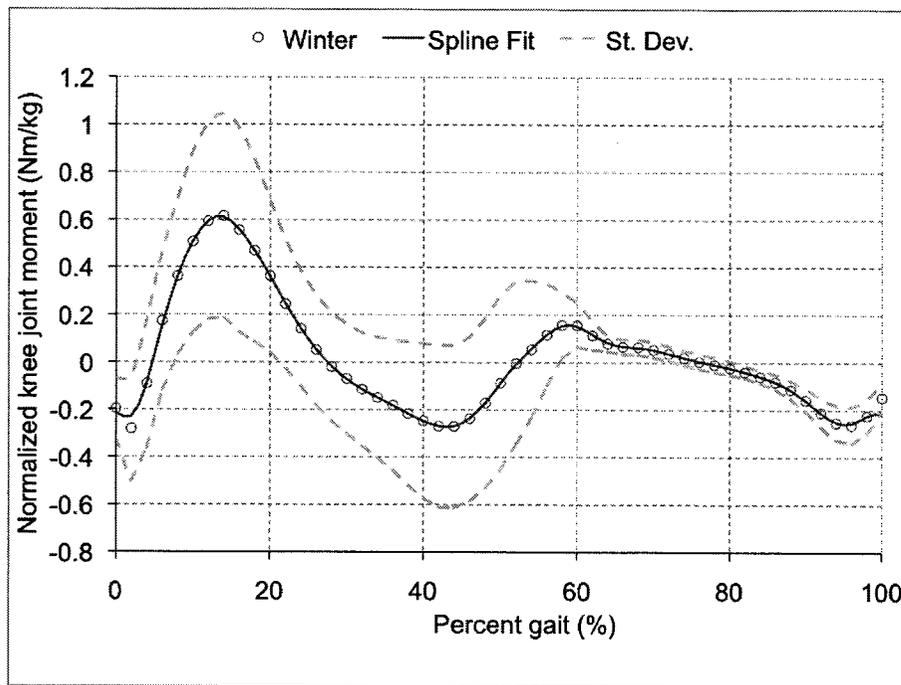


Figure 2.9: Normalized knee joint moments vs. percent gait cycle.

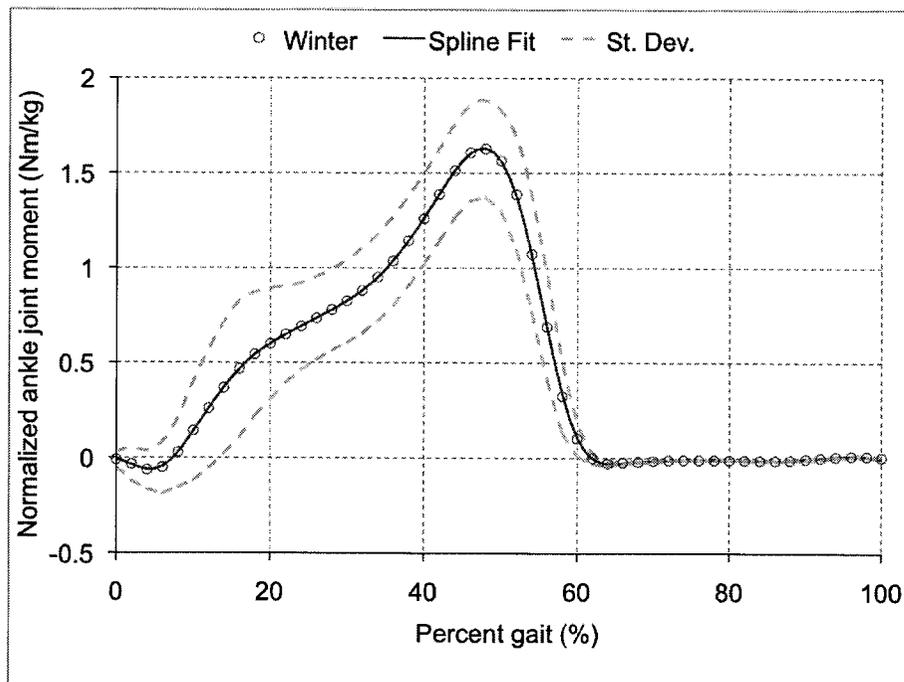
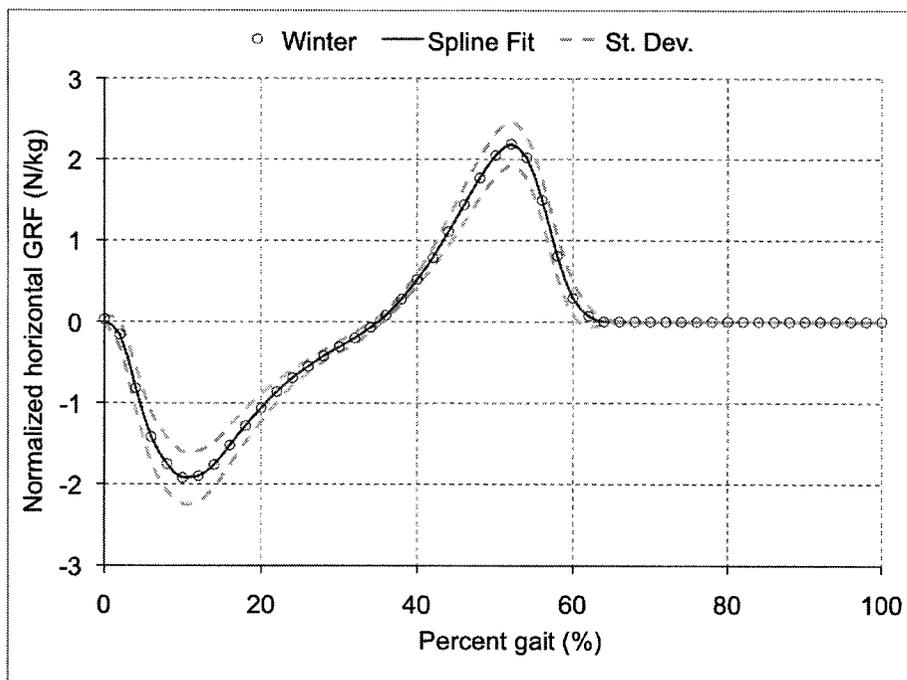
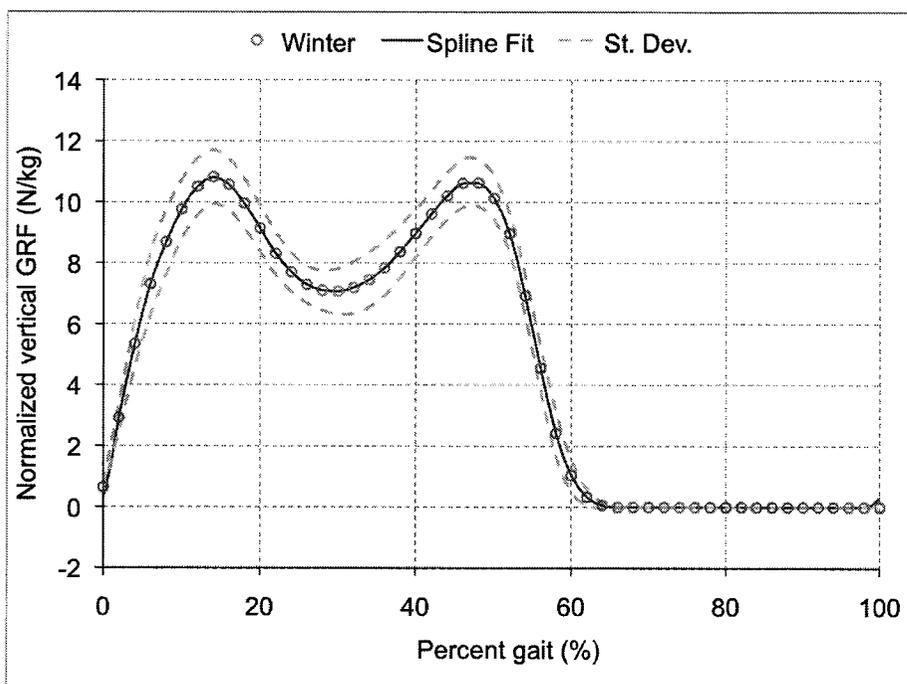


Figure 2.10: Normalized ankle joint moments vs. percent gait cycle.



**Figure 2.11:** Normalized horizontal (anterior/posterior) ground reaction forces vs. percent gait cycle.



**Figure 2.12:** Normalized vertical (anterior/posterior) ground reaction forces vs. percent gait cycle.

These average gait trajectories will be referred to in the validation of the leg model, design process, and for a controller trajectory generator.

## 2.2.6 Leg Model Verification and Validation

### Verification

The iterative Newton-Euler formulation, explored in detail in [62, 63], was used to derive the dynamics independently from the Euler-Lagrange method in order to verify the equations of motion. Arbitrary sets of leg parameters, forces at the foot, and joint trajectories were input to both equations and the output was found to be exactly the same, except for computational and truncation errors.

The forward and inverse kinematics were verified by providing an input end effector (foot) position, velocity, and acceleration, applying the inverse and forward kinematics, then comparing the result to the input values. The results showed that the equations were consistent, except for the expected computational and truncation errors. A backward difference numerical differentiation was also applied independently to the position output of the kinematics equations and were compared to the analytical velocity and acceleration calculations. The results were the same except for the numerical errors in the differentiation process.

### Model Limitations

There are significant limitations and simplifications provided in this leg model formulation. A major limitation is that the entire model is planar with only revolute joints. In reality, there are many more degrees of freedom in each of the human leg joints. The hip joint can be better approximated as a spherical joint at the head of the femur. The knee, however, does not exhibit the properties of a revolute joint, since there is a sliding motion as well as rotating motion between the surfaces of the articulated

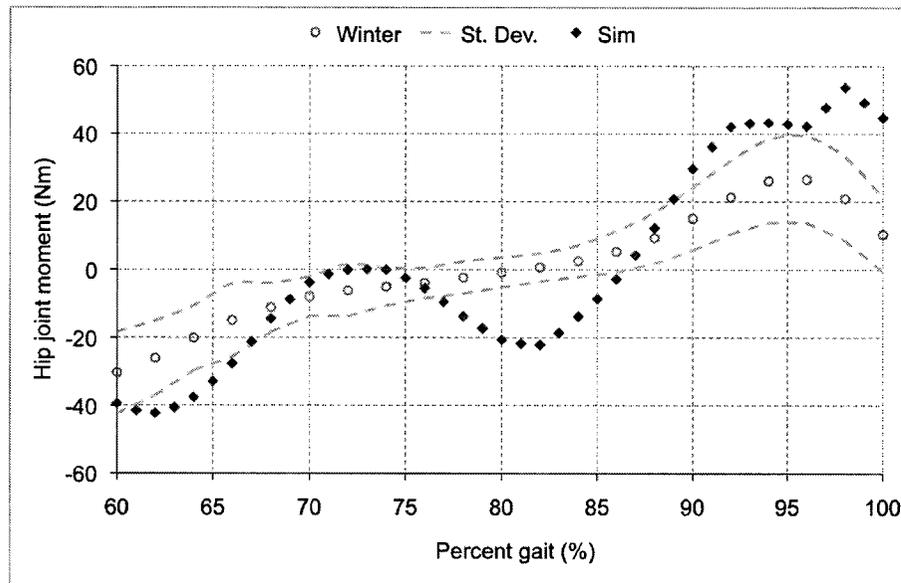
cartilage in the knee. There is also some limited motion in other degrees of freedom in the knee. The ankle is a very complex system of bones, tendons and cartilage that is better approximated, yet still simplified, by a universal joint.

The net torque applied over the leg joint is also a significant simplification. In reality, there are many complex muscle groups and tendons, variable bone geometry, muscle activation dynamics, neural motor controls, articulate surfaces, etc. that contribute to the biomechanics of the musculoskeletal system. More complicated dynamic musculoskeletal models are typically applied in a research context for analyzing healthy gait and pathologies for various gait disorders [49]. The net torque calculation from inverse dynamics allows for simple biomechanics analysis that can provide insight into the analyzed motion of the individual without the need for invasive instrumentation, detailed calibrations, complex optimizations, or individual biometrics that are difficult or impossible to measure [66].

In a clinical setting, the patient's pathology is primarily derived from medical history, a physical examination of the patient and observation of their gait. The joint trajectories and applied bulk joint moments from a gait analysis clinic are useful for determining more specific aspects of the person's gait deficiencies and what rehabilitation therapies are best suited for the patient [21]. A more detailed dynamic musculoskeletal model and analysis may yield further discoveries into various gait disorders in a research context, but is not likely to be used as a tool by the clinician for applying patient-specific rehabilitation therapies [49]. It is important to validate the ViGRR leg model with human subjects to determine whether or not a rigid body model with simple joints and bulk moments similar to the model presented in [66] is sufficient for use with ViGRR in a clinical application.

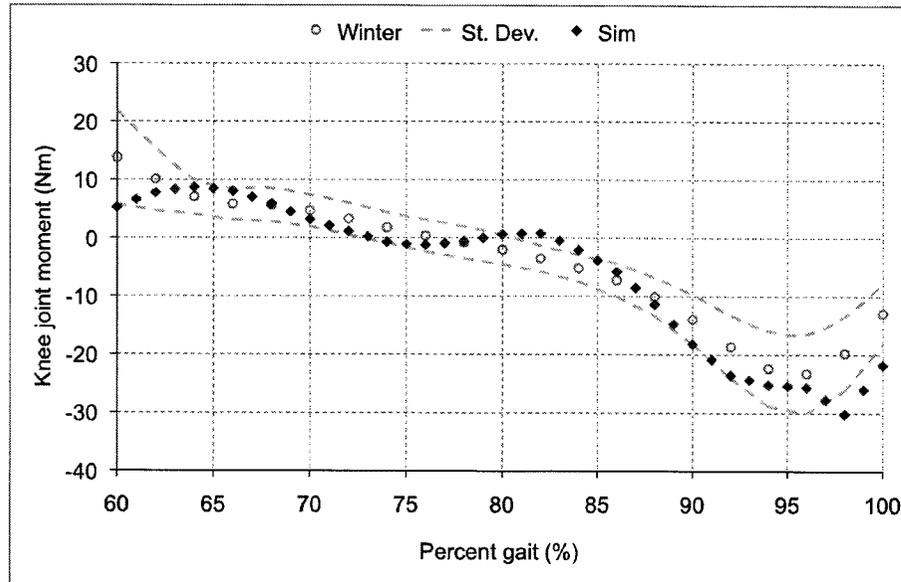
## Validation

In order to partially validate the equations of motion in application to gait analysis, the forward dynamics from the ViGRR leg model were compared to average gait data from [69]. The leg joint torques were determined using the ViGRR leg model over the swing phase, based on average joint trajectories at 100 steps per minute without any loading at the end effector. The torso was fixed at  $\theta_{tr} = 0$  degrees (vertical) and the leg properties were based on anthropometric data for an average American male which corresponds to an 88.3 kg individual with a height of 1.73 metres [75]. The calculated joint torques from the model are compared to the moments from gait analysis published by Winter [69]. Results for this validation experiment are presented for the hip, knee and ankle joints in Figures 2.13, 2.14, and 2.15, respectively.



**Figure 2.13:** ViGRR Leg model calculated hip joint torques (Sim) during the swing phase compared to average gait data (Winter) with one standard deviation bounds [69].

It can be seen that there are discrepancies between the model output and published gait analysis results. A major source of error is the fact that the torso is constrained with the ViGRR human model. During the swing phase of gait, the pelvis and torso

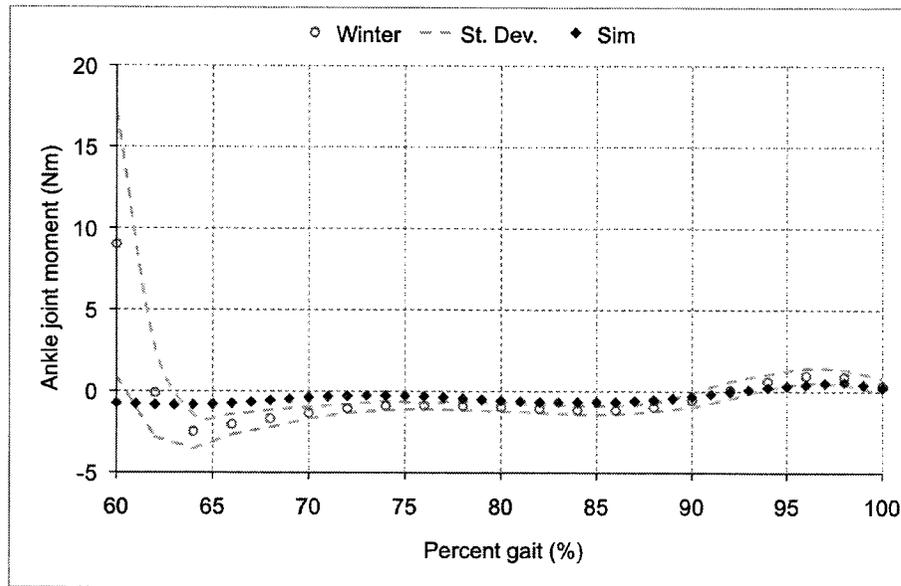


**Figure 2.14:** ViGRR Leg model calculated knee joint torques (Sim) during the swing phase compared to average gait data (Winter) with one standard deviation bounds [69].

are experiencing translation and rotation in the sagittal plane. There is also rotation of the pelvis out of plane. The largest discrepancies appear to be in the hip moments which show fewer oscillations in the Winter data set and may be attributed to the free movement of the pelvis countering the effect of the hip joint accelerating the thigh and entire lower extremities. The spike at the beginning of the ankle moment data set is attributed to the toe-off from the stance to swing portion of the gait. The simulation did not involve ground reaction forces and, accordingly, did not display the spike.

The ViGRR gait model was also used to calculate the forces at the end effector during the stance phase of gait for input average gait trajectories and joint moments from [69]. The resulting vertical ground reaction force from an 88.3 kg individual is compared to the average data in Figure 2.16.

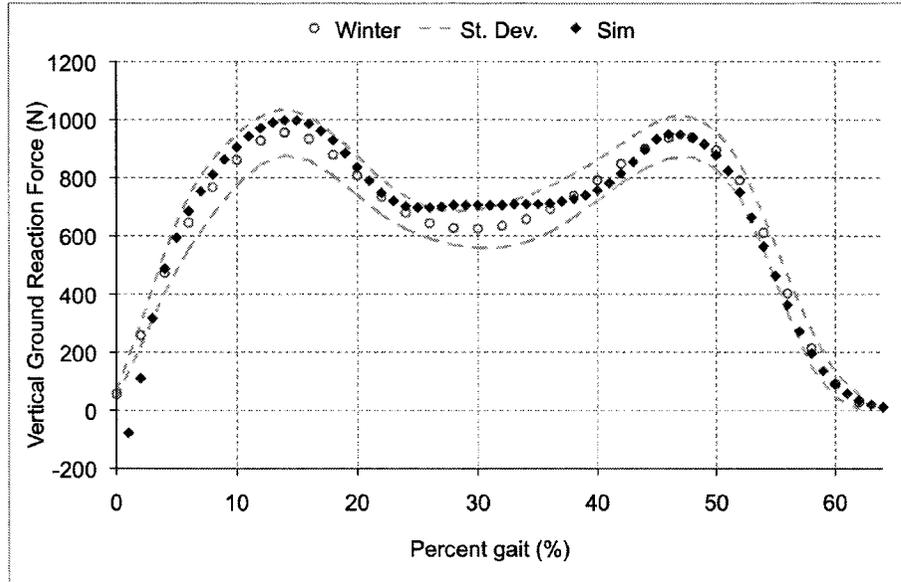
The ground reaction force normal to the floor demonstrated very similar reactions between the model and average data. The ViGRR model ground reaction force at the



**Figure 2.15:** ViGRR Leg model calculated ankle joint torques (Sim) during the swing phase compared to average gait data (Winter) with one standard deviation bounds [69].

middle of the stance phase is attenuated compared to Winter. This again is likely due to the torso and pelvis constraint which does not allow the downward acceleration of the person's centre of mass to cause a decrease in the ground reaction force. These preliminary validation results further motivate the requirement to actuate (or allow free motion of) the pelvis in a full-scale implementation of ViGRR if close to normal gait is to be experienced by the user.

The initial simulation results presented in this section provide confidence for using the presented leg model for design and simulation. However, the ViGRR prototype leg model is still optimistically constrained to the sagittal plane and in a rehabilitation setting with real individuals using ViGRR, the model has not been validated. The intention of the simplified model is to assess high-level bulk torques applied at the leg joints with a greater interest in improvement of the individual over the course of therapy and real-time information used for a training regiment. In this context, it is likely not necessary to have the precision and complexity required for more detailed



**Figure 2.16:** ViGRR Leg model calculated vertical ground reaction force (Sim) during the stance phase compared to average gait data (Winter) with one standard deviation bounds [69].

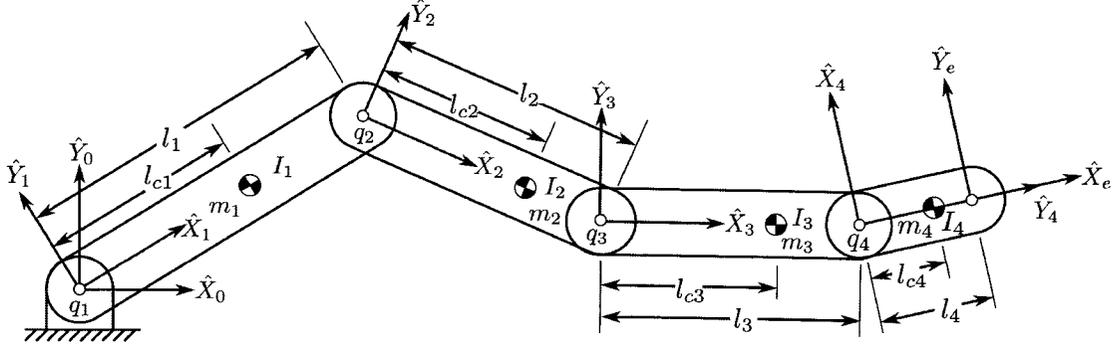
gait analysis or gait simulations. Validation of this leg model in use with ViGRR is beyond the scope of this thesis and it is discussed as potential future work in Chapter 6.

## 2.3 Robot Model

### 2.3.1 ViGRR Dynamics

The final configuration of the ViGRR prototype is a 4 DoF revolute joint serial robot. This section derives the equations of motion for this robot model. The model will be used in design, simulation, and controls. An image with the robot configuration and link definitions following the robotic conventions described in Section 2.1 are shown in Figure 2.17.

Similarly to the leg model, the dynamics for an  $n = 4$  link configuration are derived from the equations (2.1) to (2.9) yielding the following equations of motion



**Figure 2.17:** Frame, length and mass definitions for a the 4R ViGRR prototype robot.

$$\tau_r = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q), \quad (2.28)$$

where the subscript  $r$  refers to the robot,  $\tau_r = [\tau_1, \tau_2, \tau_3, \tau_4]^T$  are the applied joint torques in Nm and  $q = [q_1, q_2, q_3, q_4]^T$  is the vector of joint angles in radians. The matrices  $C(q, \dot{q})$ ,  $M(q)$ , and vector  $g(q)$  are defined below. To conserve space, the trigonometric functions are displayed in short form as  $c_{ij\dots m} = \cos(q_i + q_j + \dots + q_m)$  and  $s_{ij\dots m} = \sin(q_i + q_j + \dots + q_m)$ . The Matlab Symbolic toolbox was used for the derivation of the following ViGRR dynamic matrices:

$$C(q, \dot{q}) = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{bmatrix}, \quad (2.29)$$

where

$$\begin{aligned}
c_{1,1} &= -s_2(l_1l_2\dot{q}_2m_3 + l_1l_2\dot{q}_2m_4 + l_1l_{C2}\dot{q}_2m_2) - s_{34}(l_2l_{C4}\dot{q}_3m_4 + l_2l_{C4}\dot{q}_4m_4) \\
&\quad - s_3(l_2l_3\dot{q}_3m_4 + l_2l_{C3}\dot{q}_3m_3) - s_{234}(l_1l_{C4}\dot{q}_2m_4 + l_1l_{C4}\dot{q}_3m_4 + l_1l_{C4}\dot{q}_4m_4) \\
&\quad - s_{23}(l_1l_3\dot{q}_2m_4 + l_1l_3\dot{q}_3m_4 + l_1l_{C3}\dot{q}_2m_3 + l_1l_{C3}\dot{q}_3m_3) - l_3l_{C4}\dot{q}_4m_4s_4 \\
c_{1,2} &= -s_{34}(l_2l_{C4}\dot{q}_3m_4 + l_2l_{C4}\dot{q}_4m_4) \\
&\quad - s_{234}(l_1l_{C4}\dot{q}_1m_4 + l_1l_{C4}\dot{q}_2m_4 + l_1l_{C4}\dot{q}_3m_4 + l_1l_{C4}\dot{q}_4m_4) \\
&\quad - s_{23}(l_1l_3\dot{q}_1m_4 + l_1l_3\dot{q}_2m_4 + l_1l_3\dot{q}_3m_4 + l_1l_{C3}\dot{q}_1m_3 + l_1l_{C3}\dot{q}_2m_3 + l_1l_{C3}\dot{q}_3m_3) \\
&\quad - s_3(l_2l_3\dot{q}_3m_4 + l_2l_{C3}\dot{q}_3m_3) - l_3l_{C4}\dot{q}_4m_4s_4 \\
&\quad - s_2(l_1l_2\dot{q}_1m_3 + l_1l_2\dot{q}_2m_3 + l_1l_2\dot{q}_1m_4 + l_1l_2\dot{q}_2m_4 + l_1l_{C2}\dot{q}_1m_2 + l_1l_{C2}\dot{q}_2m_2) \\
c_{1,3} &= -s_{234}(l_1l_{C4}\dot{q}_1m_4 + l_1l_{C4}\dot{q}_2m_4 + l_1l_{C4}\dot{q}_3m_4 + l_1l_{C4}\dot{q}_4m_4) \\
&\quad - s_{23}(l_1l_3\dot{q}_1m_4 + l_1l_3\dot{q}_2m_4 + l_1l_3\dot{q}_3m_4 + l_1l_{C3}\dot{q}_1m_3 + l_1l_{C3}\dot{q}_2m_3 + l_1l_{C3}\dot{q}_3m_3) \\
&\quad - s_3(l_2l_3\dot{q}_1m_4 + l_2l_3\dot{q}_2m_4 + l_2l_3\dot{q}_3m_4 + l_2l_{C3}\dot{q}_1m_3 + l_2l_{C3}\dot{q}_2m_3 + l_2l_{C3}\dot{q}_3m_3) \\
&\quad - s_{34}(l_2l_{C4}\dot{q}_1m_4 + l_2l_{C4}\dot{q}_2m_4 + l_2l_{C4}\dot{q}_3m_4 + l_2l_{C4}\dot{q}_4m_4) - l_3l_{C4}\dot{q}_4m_4s_4 \\
c_{1,4} &= -l_2l_{C4}m_4s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - l_3l_{C4}m_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
&\quad - l_1l_{C4}m_4s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
c_{2,1} &= s_2(l_1l_2\dot{q}_1m_3 + l_1l_2\dot{q}_1m_4 + l_1l_{C2}\dot{q}_1m_2) + s_{23}(l_1l_3\dot{q}_1m_4 + l_1l_{C3}\dot{q}_1m_3) \\
&\quad - s_{34}(l_2l_{C4}\dot{q}_3m_4 + l_2l_{C4}\dot{q}_4m_4) - s_3(l_2l_3\dot{q}_3m_4 + l_2l_{C3}\dot{q}_3m_3) \\
&\quad - l_3l_{C4}\dot{q}_4m_4s_4 + l_1l_{C4}\dot{q}_1m_4s_{234} \\
c_{2,2} &= -s_{34}(l_2l_{C4}\dot{q}_3m_4 + l_2l_{C4}\dot{q}_4m_4) - s_3(l_2l_3\dot{q}_3m_4 + l_2l_{C3}\dot{q}_3m_3) - l_3l_{C4}\dot{q}_4m_4s_4 \\
c_{2,3} &= -s_3(l_2l_3\dot{q}_1m_4 + l_2l_3\dot{q}_2m_4 + l_2l_3\dot{q}_3m_4 + l_2l_{C3}\dot{q}_1m_3 + l_2l_{C3}\dot{q}_2m_3 + l_2l_{C3}\dot{q}_3m_3) \\
&\quad - s_{34}(l_2l_{C4}\dot{q}_1m_4 + l_2l_{C4}\dot{q}_2m_4 + l_2l_{C4}\dot{q}_3m_4 + l_2l_{C4}\dot{q}_4m_4) - l_3l_{C4}\dot{q}_4m_4s_4 \\
c_{2,4} &= -l_2l_{C4}m_4s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - l_3l_{C4}m_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
c_{3,1} &= s_{23}(l_1l_3\dot{q}_1m_4 + l_1l_{C3}\dot{q}_1m_3) + s_{34}(l_2l_{C4}\dot{q}_1m_4 + l_2l_{C4}\dot{q}_2m_4) \\
&\quad + s_3(l_2l_3\dot{q}_1m_4 + l_2l_3\dot{q}_2m_4 + l_2l_{C3}\dot{q}_1m_3 + l_2l_{C3}\dot{q}_2m_3) \\
&\quad - l_3l_{C4}\dot{q}_4m_4s_4 + l_1l_{C4}\dot{q}_1m_4s_{234} \\
c_{3,2} &= s_{34}(l_2l_{C4}\dot{q}_1m_4 + l_2l_{C4}\dot{q}_2m_4) - l_3l_{C4}\dot{q}_4m_4s_4 \\
&\quad + s_3(l_2l_3\dot{q}_1m_4 + l_2l_3\dot{q}_2m_4 + l_2l_{C3}\dot{q}_1m_3 + l_2l_{C3}\dot{q}_2m_3) \\
c_{3,3} &= -l_3l_{C4}\dot{q}_4m_4s_4 \\
c_{3,4} &= -l_3l_{C4}m_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
c_{4,1} &= l_{C4}m_4s_{34}(l_2\dot{q}_1 + l_2\dot{q}_2) + l_{C4}m_4s_4(l_3\dot{q}_1 + l_3\dot{q}_2 + l_3\dot{q}_3) + l_1l_{C4}\dot{q}_1m_4s_{234} \\
c_{4,2} &= l_{C4}m_4s_{34}(l_2\dot{q}_1 + l_2\dot{q}_2) + l_{C4}m_4s_4(l_3\dot{q}_1 + l_3\dot{q}_2 + l_3\dot{q}_3) \\
c_{4,3} &= l_3l_{C4}m_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\
c_{4,4} &= 0;
\end{aligned}$$

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{12} & m_{22} & m_{23} & m_{24} \\ m_{13} & m_{23} & m_{33} & m_{34} \\ m_{14} & m_{24} & m_{34} & m_{44} \end{bmatrix}, \quad (2.30)$$

where

$$\begin{aligned} m_{11} &= m_1 l_{C_1}^2 + m_4 l_{C_4}^2 c_{1234}^2 + m_4 l_{C_4}^2 s_{1234}^2 + 2l_3 m_4 l_{C_4} c_{123} c_{1234} \\ &\quad + 2l_3 m_4 l_{C_4} s_{123} s_{1234} + 2l_2 m_4 l_{C_4} c_{1234} c_{12} + 2l_1 m_4 l_{C_4} c_{1234} c_1 \\ &\quad + 2l_2 m_4 l_{C_4} s_{1234} s_{12} + 2l_1 m_4 l_{C_4} s_{1234} s_1 + (m_4 l_3^2 + m_3 l_{C_3}^2) c_{123}^2 \\ &\quad + (2l_2 l_3 m_4 + 2l_2 l_{C_3} m_3) c_{123} c_{12} + (2l_1 l_3 m_4 + 2l_1 l_{C_3} m_3) c_{123} c_1 + (m_4 l_3^2 + m_3 l_{C_3}^2) s_{123}^2 \\ &\quad + (2l_2 l_3 m_4 + 2l_2 l_{C_3} m_3) s_{123} s_{12} + (2l_1 l_3 m_4 + 2l_1 l_{C_3} m_3) s_{123} s_1 \\ &\quad + (l_2^2 m_3 + l_2^2 m_4 + l_{C_2}^2 m_2) c_{12}^2 + (2l_1 l_2 m_3 + 2l_1 l_2 m_4 + 2l_1 l_{C_2} m_2) c_{12} c_1 \\ &\quad + (l_2^2 m_3 + l_2^2 m_4 + l_{C_2}^2 m_2) s_{12}^2 + (2l_1 l_2 m_3 + 2l_1 l_2 m_4 + 2l_1 l_{C_2} m_2) s_{12} s_1 \\ &\quad + (l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4) c_1^2 + (l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4) s_1^2 + I_1 + I_2 + I_3 + I_4 \\ m_{12} &= m_4 l_{C_4}^2 c_{1234}^2 + m_4 l_{C_4}^2 s_{1234}^2 + 2l_3 m_4 l_{C_4} c_{123} c_{1234} + 2l_3 m_4 l_{C_4} s_{123} s_{1234} \\ &\quad + 2l_2 m_4 l_{C_4} c_{1234} c_{12} + l_1 m_4 c_1 l_{C_4} c_{1234} + 2l_2 m_4 l_{C_4} s_{1234} s_{12} + l_1 m_4 s_1 l_{C_4} s_{1234} \\ &\quad + (m_4 l_3^2 + m_3 l_{C_3}^2) c_{123}^2 + (2l_2 l_3 m_4 + 2l_2 l_{C_3} m_3) c_{123} c_{12} + c_1 (l_1 l_3 m_4 + l_1 l_{C_3} m_3) c_{123} \\ &\quad + (m_4 l_3^2 + m_3 l_{C_3}^2) s_{123}^2 + (2l_2 l_3 m_4 + 2l_2 l_{C_3} m_3) s_{123} s_{12} + s_1 (l_1 l_3 m_4 + l_1 l_{C_3} m_3) s_{123} \\ &\quad + (l_2^2 m_3 + l_2^2 m_4 + l_{C_2}^2 m_2) c_{12}^2 + c_1 (l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 l_{C_2} m_2) c_{12} \\ &\quad + (l_2^2 m_3 + l_2^2 m_4 + l_{C_2}^2 m_2) s_{12}^2 + s_1 (l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 l_{C_2} m_2) s_{12} + I_2 + I_3 + I_4 \\ m_{13} &= I_3 + I_4 + c_{23} (l_1 l_3 m_4 + l_1 l_{C_3} m_3) + c_3 (l_2 l_3 m_4 + l_2 l_{C_3} m_3) + l_3^2 m_4 + l_{C_3}^2 m_3 + l_{C_4}^2 m_4 \\ &\quad + l_2 l_{C_4} m_4 c_{34} + 2l_3 l_{C_4} m_4 c_4 + l_1 l_{C_4} m_4 c_{234} \\ m_{14} &= I_4 + l_{C_4}^2 m_4 + l_2 l_{C_4} m_4 c_{34} + l_3 l_{C_4} m_4 c_4 + l_1 l_{C_4} m_4 c_{234} \\ m_{22} &= m_2 l_{C_2}^2 + m_4 l_{C_4}^2 c_{1234}^2 + m_4 l_{C_4}^2 s_{1234}^2 + 2l_3 m_4 l_{C_4} c_{123} c_{1234} + 2l_3 m_4 l_{C_4} s_{123} s_{1234} \\ &\quad + 2l_2 m_4 l_{C_4} c_{1234} c_{12} + 2l_2 m_4 l_{C_4} s_{1234} s_{12} + (m_4 l_3^2 + m_3 l_{C_3}^2) c_{123}^2 \\ &\quad + (2l_2 l_3 m_4 + 2l_2 l_{C_3} m_3) c_{123} c_{12} + (m_4 l_3^2 + m_3 l_{C_3}^2) s_{123}^2 + (2l_2 l_3 m_4 + 2l_2 l_{C_3} m_3) s_{123} s_{12} \\ &\quad + (l_2^2 m_3 + l_2^2 m_4) c_{12}^2 + (l_2^2 m_3 + l_2^2 m_4) s_{12}^2 + I_2 + I_3 + I_4 \\ m_{23} &= I_3 + I_4 + c_3 (l_2 l_3 m_4 + l_2 l_{C_3} m_3) + l_3^2 m_4 + l_{C_3}^2 m_3 + l_{C_4}^2 m_4 + l_2 l_{C_4} m_4 c_{34} + 2l_3 l_{C_4} m_4 c_4 \\ m_{24} &= I_4 + l_{C_4}^2 m_4 + l_2 l_{C_4} m_4 c_{34} + l_3 l_{C_4} m_4 c_4 \\ m_{33} &= m_4 l_3^2 c_{123}^2 + m_4 l_3^2 s_{123}^2 + 2m_4 l_3 l_{C_4} c_{123} c_{1234} + 2m_4 l_3 l_{C_4} s_{123} s_{1234} + m_3 l_{C_3}^2 \\ &\quad + m_4 l_{C_4}^2 c_{1234}^2 + m_4 l_{C_4}^2 s_{1234}^2 + I_3 + I_4 \\ m_{34} &= m_4 l_{C_4}^2 + l_3 m_4 c_4 l_{C_4} + I_4 \\ m_{44} &= m_4 l_{C_4}^2 + I_4; \end{aligned}$$

$$g(q) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}, \quad (2.31)$$

where

$$\begin{aligned} g_1 &= gm_3(l_2c_{12} + l_1c_1 + l_{C3}c_{123}) + gm_2(l_{C2}c_{12} + l_1c_1) \\ &\quad + gm_4(l_{C4}c_{1234} + l_2c_{12} + l_1c_1 + l_3c_{123}) + l_{C1}gm_1c_1 \\ g_2 &= gm_3(l_2c_{12} + l_{C3}c_{123}) + gm_4(l_{C4}c_{1234} + l_2c_{12} + l_3c_{123}) + l_{C2}gm_2c_{12} \\ g_3 &= gm_4(l_{C4}c_{1234} + l_3c_{123}) + l_{C3}gm_3c_{123} \\ g_4 &= l_{C4}gm_4c_{1234}. \end{aligned}$$

## 2.3.2 Robot Kinematics

### Forward Kinematics

The forward kinematics describes the ViGRR end effector coordinate frame position and orientation (denoted by subscript  $e$  in Figure 2.17) based on the robot joint angles  $q$ . The pose  $X_e$  is defined as the planar position coordinates  $(x_e, y_e)$  and the final link orientation with respect to the global coordinate frame by angle  $\psi_e$ . The angle  $\psi_e$  is the angle from  $\hat{X}_0$  to  $\hat{X}_e$  about the  $\hat{Z}_0$  axis. The transformation from joint angles to the end effector pose relative to the fixed global frame is

$$X_e = \begin{Bmatrix} x_e \\ y_e \\ \psi_e \end{Bmatrix} = \begin{Bmatrix} l_1c_1 + l_2c_{12} + l_3c_{123} + l_3c_{1234} \\ l_1s_1 + l_2s_{12} + l_3s_{123} + l_3s_{1234} \\ q_1 + q_2 + q_3 + q_4 \end{Bmatrix}. \quad (2.32)$$

### Inverse Kinematics With Constrained Joint Angle

The 4R robot is redundant by a single extra degree of freedom. This means that the solution to the problem of mapping from  $X_e \in \mathbf{R}^3$  to  $q \in \mathbf{R}^4$  is not unique. To solve this inverse kinematics problem, optimal solutions relating the joint positions to minimize desired objective functions can be applied. Alternatively, by constraining one of the joint angles, the remaining three angles can be determined exactly from

the end effector position and orientation. For the ViGRR problem, it was chosen to constrain  $q_4$  due to the simplicity of finding the limits of  $q_4$  that are compatible given an end effector pose  $X_e$  as demonstrated in Section 2.3.3. By constraining the 4th joint angle, the inverse kinematics are calculated from

$$q = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{Bmatrix} \text{atan2}(y_3, x_3) \pm \cos^{-1} \left( \frac{x_3^2 + y_3^2 + l_1^2 - l_2^2}{2l_1 \sqrt{x_3^2 + y_3^2}} \right) \\ \pm \cos^{-1} \left( \frac{x_3^2 + y_3^2 - l_1^2 - l_2^2}{2l_1 l_2} \right) \\ \psi_e - q_2 - q_1 - q_4 \\ q_4 \end{Bmatrix}, \quad (2.33)$$

where

$$\begin{aligned} x_3 &= x_e - l_4 \cos(\psi_e) - l_3 \cos(\psi_e - q_4) \\ y_3 &= y_e - l_4 \sin(\psi_e) - l_3 \sin(\psi_e - q_4). \end{aligned}$$

### Differential Kinematic Relations and Statics

The velocity of the end effector of the robot can be calculated based on the joint velocities by differentiating (2.32) with respect to time, in order to obtain

$$\dot{X}_e = J_r \dot{q}, \quad (2.34)$$

where the ViGRR Jacobian  $J_r$  is defined as

$$J_r = \begin{bmatrix} j_{11} & j_{12} & j_{13} & j_{14} \\ j_{21} & j_{22} & j_{23} & j_{24} \\ j_{31} & j_{32} & j_{33} & j_{34} \end{bmatrix}, \quad (2.35)$$

where

$$\begin{aligned}
j_{11} &= -l_1 s_1 - l_2 s_{12} - l_3 s_{123} - l_4 s_{1234} \\
j_{21} &= l_1 c_1 + l_2 c_{12} + l_3 c_{123} + l_4 c_{1234} \\
j_{31} &= 1 \\
j_{12} &= -l_2 s_{12} - l_3 s_{123} - l_4 s_{1234} \\
j_{22} &= l_2 c_{12} + l_3 c_{123} + l_4 c_{1234} \\
j_{32} &= 1 \\
j_{13} &= -l_3 s_{123} - l_4 s_{1234} \\
j_{23} &= l_3 c_{123} + l_4 c_{1234} \\
j_{33} &= 1 \\
j_{14} &= -l_4 s_{1234} \\
j_{24} &= l_4 c_{1234} \\
j_{34} &= 1.
\end{aligned}$$

The transpose of the Jacobian relates the forces and torques at the end-effector  $f_e = [f_{e_x}, f_{e_y}, \tau_{e_z}]^T$  in (N, N, Nm) to the torques applied at the joints  $\tau_r$  in (Nm) by

$$\tau_h = J_r^T f_e, \quad (2.36)$$

where  $f_{e_x}$  and  $f_{e_y}$  denote the forces relative to the base frame  $\hat{X}_0$  and  $\hat{Y}_0$  directions, respectively.  $\tau_{e_z}$  is the torque about the  $\hat{Z}_0$  axis. The relationship between end effector and joint accelerations is derived by differentiating (2.34) with respect to time to obtain

$$\ddot{X}_e = \dot{J}_r \dot{q} + J_r \ddot{q}. \quad (2.37)$$

The inverse differential relationships do not have unique solutions due to redundancy in the robot. As is the case for redundant robots, the ViGRR Jacobian is not a square matrix and cannot be inverted. A common solution to such a problem is to introduce the Jacobian pseudoinverse defined by

$$J^\dagger = J_r^T (J_r J_r^T)^{-1}, \quad (2.38)$$

which is used to calculate the the joint velocities from the end effector velocity in

$$\dot{q} = J^\dagger \dot{X}_e, \quad (2.39)$$

where  $\|\dot{q}\|$  is minimized by the pseudoinverse [63, 76]. From (2.37) and using the pseudoinverse (2.38) the joint accelerations can also be calculated from the end effector accelerations via

$$\ddot{q} = J^\dagger (\ddot{X}_e - \dot{J}_r \dot{q}). \quad (2.40)$$

### 2.3.3 Redundancy Resolution

The advantage of having redundant degrees of freedom is that algorithms can be employed which optimize the configuration of the robot to avoid collisions and singularities or locally maximizing any objective function of  $q$ . Another solution to the inverse velocity kinematics problem is given by

$$\dot{q} = J^\dagger \dot{X}_e + (I - J^\dagger J_r) \dot{q}^0, \quad (2.41)$$

where  $\dot{q}^0$  is any arbitrary vector [76]. By premultiplying both sides of the equation by  $J_r$ , both sides simplify to equal the end effector velocity  $\dot{X}_e$  since  $J_r J^\dagger = I$ . This shows that the term  $(I - J^\dagger J_r) \dot{q}^0$  lies in the null space of  $J_r$  and  $\dot{q}^0$  does not move the end effector, providing only self motion of the robot. This property can be exploited for the redundancy resolution scheme as we can choose  $\dot{q}^0$  to move to a more optimal configuration according to objective functions of  $q$  [76].

If we let  $\omega(q)$  be an objective function to be maximized locally, then  $\dot{q}^0$  can be specified by the scaled gradient

$$\dot{q}^0 = k^0 \frac{\partial \omega(q)}{\partial q}, \quad (2.42)$$

where  $k^0 > 0$  [76].

Three objective functions for optimizing the configuration are performed for the

planar rehabilitation robot. The first is a joint limit avoidance function  $\omega(q)_{lim}$ , the second is a singularity avoidance scheme  $\omega(q)_{sing}$  and the third is a function which avoids incompatible configurations  $\omega(q)_{comp}$ . The resulting null space velocity vector is given as a sum of the objective function gradients in

$$\dot{q}^0 = k_{lim}^0 \frac{\partial \omega(q)_{lim}}{\partial q} + k_{sing}^0 \frac{\partial \omega(q)_{sing}}{\partial q} + k_{comp}^0 \frac{\partial \omega(q)_{comp}}{\partial q}. \quad (2.43)$$

### Joint Limit Avoidance

An objective function of the form

$$\omega(q)_{lim} = -\frac{1}{8} \sum_{i=1}^n \left( \frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \quad (2.44)$$

optimizes the joint configuration to move away from the maximum  $q_{iM}$  and minimum  $q_{im}$  joint limits for each joint  $i$  toward the midpoint at  $\bar{q}_i$  [76]. It is a quadratic function for each joint with a maximum at  $\bar{q}_i$  and its partial derivative with respect to each joint angle from (2.42) is simply the linear function

$$\frac{\partial \omega(q)_{lim}}{\partial q_i} = -\frac{1}{4} \left( \frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right). \quad (2.45)$$

### Singularity Avoidance

The manipulability measure defined by

$$\omega(q)_{sing} = \sqrt{\det(J_r J_r^T)} \quad (2.46)$$

is typically used to characterize and visualize the possible end effector velocities from joint velocities [63, 76]. It is defined as the product of singular values of the Jacobian. The manipulability measure disappears at singular configurations and its magnitude is indicative of the range of possible velocity vectors at the end effector, making it a useful objective function to maximize and avoid singular configurations [76]. The

Matlab Symbolic toolbox was used to obtain the partial derivatives of (2.46) with respect to  $q_i$ .

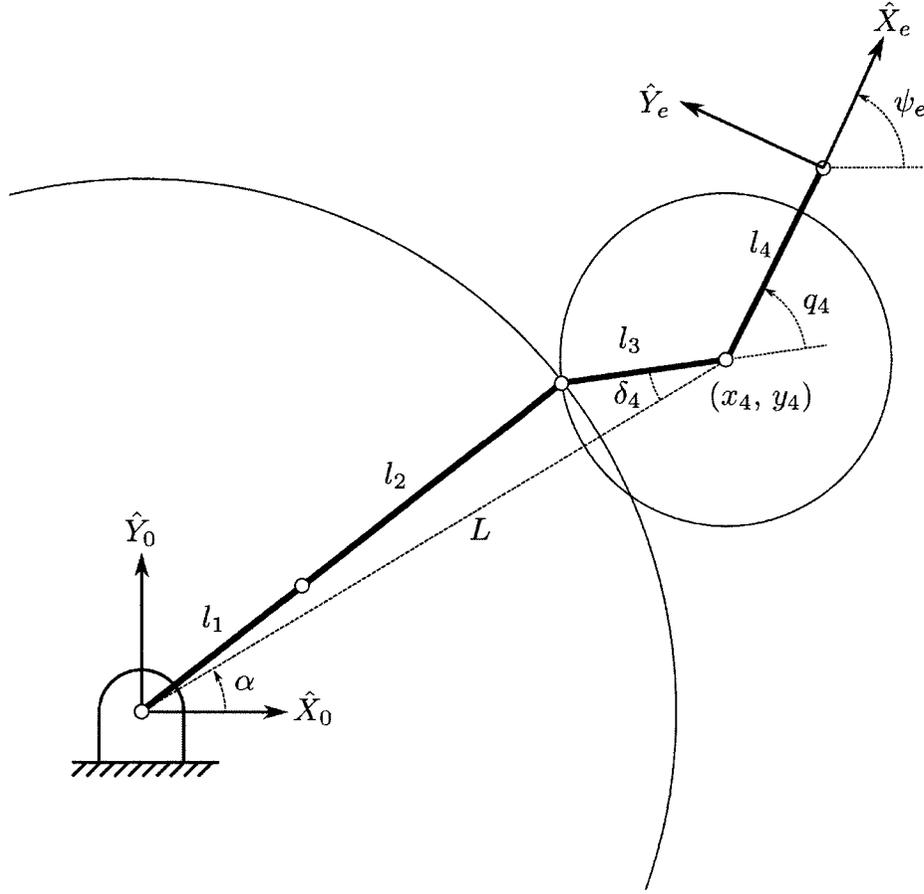
### Incompatible Configuration Avoidance

The desired pose of the end effector provides a desired angle and position for link 4. If the 4th linkage coordinate frame cannot be reached the desired pose is incompatible with the robot. Due to the planar configuration and only one extra degree of freedom, it is possible to calculate the allowable angle range for joint 4 given an end effector position and orientation. This is useful when calculating the exact inverse kinematics using (2.33). The  $q_4$  joint limits can also be applied to initializing the inverse kinematics and continuously checking the robot compatibility to prevent computational issues or controller instability. The upper limit for  $q_4$  is defined as  $q_L$ , the lower limit is  $q_l$  and their midpoint is  $\bar{q}_{lim}$ . A quadratic objective function can be constructed similar to the joint limit avoidance scheme:

$$\omega(q)_{comp} = -\frac{1}{8} \left( \frac{q_4 - \bar{q}_{lim}}{q_L - q_l} \right)^2. \quad (2.47)$$

The geometric derivation of the calculation for the  $q_4$  compatibility joint limits is based on Figure 2.18 and is essentially the intersection between two circles. One circle is defined by a centre at the base and a radius  $(l_1 + l_2)$  constructed from fully extended links 1 and 2. The other circle is defined by a centre at the link 4 frame origin with a radius of  $l_3$ . Link 4 is constrained by the end effector position and orientation.

The angle of  $q_4$  directing link 3 toward the base origin is the midpoint between the two  $q_4$  limits and is calculated by



**Figure 2.18:** Parameters and diagram for incompatible configuration avoidance.

$$\begin{aligned}
 x_4 &= x_e - l_4 \cos(\psi_e) \\
 y_4 &= y_e - l_4 \sin(\psi_e) \\
 L &= \sqrt{x_4^2 + y_4^2} \\
 \alpha &= \text{atan2}(y_4, x_4) \\
 \bar{q}_{lim} &= \psi_e - \alpha.
 \end{aligned} \tag{2.48}$$

The compatibility can now be checked. If  $(L > l_1 + l_2 + l_3)$ , then there are no intersections and the end effector position and orientation is not reachable. If  $(L < l_1 + l_2 - l_3)$  all values of  $q_4$  are valid (ignoring joint limits) and the limits are set to  $q_L = \bar{q}_{lim} + \pi$  and  $q_l = \bar{q}_{lim} - \pi$ . Finally, if the circles are intersecting, the  $q_4$  compatibility limits are calculated from the cosine law as follows:

$$\begin{aligned}
\delta_4 &= \cos^{-1} \left( \frac{l_3^2 + L^2 - (l_1 + l_2)^2}{2Ll_3} \right) \\
q_L &= \bar{q}_{lim} + \delta_4 \\
q_l &= \bar{q}_{lim} - \delta_4.
\end{aligned} \tag{2.49}$$

The incompatible configurations defined in this section are also singular configurations of the robot. The singularity avoidance scheme and associated manipulability measure may be redundant with this incompatible configuration avoidance scheme.

Finding the compatible limits of  $q_4$  is not only useful for the objective function (2.47), but also for checking desired end effector positions fed into the controller for compatibility, limiting  $q_4$  before calculating the exact inverse kinematics in (2.33), and initializing the redundancy resolution scheme.

### 2.3.4 Drive Train Dynamics

The robot dynamics were derived with external torques applied at each joint to obtain (2.28). This section introduces the motor and gear dynamics. For a motor and gear assembly at joint  $k$  the dynamics are given by

$$\tau_{m_k} - \tau_{r_k}/r_k = I_{m_k} \ddot{\theta}_{m_k} + B_{m_k} \dot{\theta}_{m_k}, \tag{2.50}$$

where  $\theta_{m_k}$  is the motor shaft angle,  $\tau_{m_k}$  is the input motor torque,  $r_k$  is the gear ratio,  $\tau_{r_k}$  is the output torque,  $B_{m_k}$  is the viscous damping (friction) term for the motor and gear, and  $I_{m_k}$  is the inertia of the gear and motor armature [63].

The gear ratio provides a relationship between the motor shaft angles and output angle of the joint in

$$\theta_{m_k} = r_k q_k. \tag{2.51}$$

Substituting (2.51) into (2.50) yields

$$\tau_{r_k} = r_k \tau_{m_k} - r_k^2 I_{m_k} \ddot{q}_k - r_k^2 B_{m_k} \dot{q}_k \quad (2.52)$$

We can now combine the drive train dynamics (2.52) with the robot dynamics (2.28) and simplify to obtain

$$u = M'(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + B_{\text{eff}}\dot{q}, \quad (2.53)$$

where the input torque vector components are defined as  $u_k = r_k \tau_{m_k}$ , the effective inertia matrix diagonals are  $I_{\text{eff}_k} = r_k^2 I_{m_k}$ , the effective damping matrix diagonals are  $B_{\text{eff}_k} = r_k^2 B_{m_k}$ , and the modified inertia matrix is  $M'(q) = M(q) + I_{\text{eff}}$ .

Coulomb friction  $T_{c_k}(\dot{q})$  for each joint  $k$  is modeled as a hyperbolic function of the joint velocity scaled by  $n_k$  with a viscous friction coefficient  $\nu_k$ :

$$T_{c_k}(\dot{q}) = \nu_k \tanh(u_k \dot{q}_k). \quad (2.54)$$

At zero velocity, there is usually a larger static friction to overcome in typical gear assemblies. For harmonic drives, there is a lower starting torque due to flexibility in the components. Complex friction curves can be generated for low velocities with harmonic drives [77], but estimating the coefficients can be a difficult task and a simplified tanh function model should be sufficient for the design, simulation and control of ViGRR. By including coulomb friction (2.54) and end effector forces (2.22), the final robot dynamics model becomes

$$u = M'(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + B_{\text{eff}}\dot{q} + T_c(\dot{q}) + J_r f_e. \quad (2.55)$$

This model will serve to calculate actuator requirements for the robot design, simulate the robot, and provide a plant model for the controller.

### 2.3.5 Tool Transformations

The coordinate frame of the force/torque (FT) sensor is not aligned with the last link of the robot and a tool transformation must be applied to the torques and forces measured by the sensor. The FT sensor tool coordinate frame is denoted by the subscript  $t$  for a 3D orthogonal basis following the right hand rule convention. Assuming that the final link is a rigid body, the tool transformation is given by

$$\begin{Bmatrix} F_e \\ T_e \end{Bmatrix} = {}^eT_t \begin{Bmatrix} F_t \\ T_t \end{Bmatrix} = \begin{bmatrix} {}^eR_t & 0_3 \\ ({}^eP_t \times) {}^eR_t & {}^eR_t \end{bmatrix} \begin{Bmatrix} F_t \\ T_t \end{Bmatrix}, \quad (2.56)$$

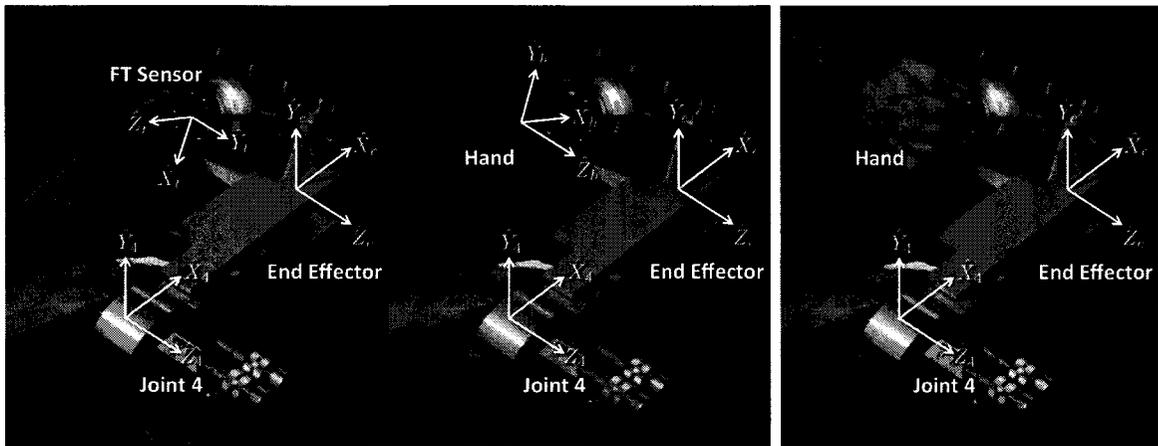
where  $F_e$  and  $T_e$  are the three dimensional forces and torques at the end effector frame and  $F_t$  and  $T_t$  are the forces and torques measured at the FT sensor coordinate frame.  ${}^eR_t$  is the rotation matrix from the tool frame to end effector frame and  $({}^eP_t \times)$  is the skew symmetric matrix constructed from the position of the tool frame origin relative to the end effector frame  ${}^eP_t = [P_x, P_y, P_z]^T$ :

$$({}^eP_t \times) = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix}. \quad (2.57)$$

### 2.3.6 Leg-Robot Interaction Transformations

Thus far, we have considered the leg and robot to be separate entities. For assessment of the individual using the device, the interaction between the robot and leg must be defined. For the interface between the robot and leg, it is assumed that the foot is rigidly attached to the robot end effector. In reality, the person is wearing a shoe and is strapped into the robot at the foot with unknown interaction dynamics. The rigidity assumption allows for the assessment of the individual using only the robot state and interaction forces without additional sensors. With average gait data having large deviations in healthy populations and by using the simple bulk torque model of the leg joints, accuracy is not a stringent requirement for the purposes of online gait

assessment in the control loop. However, an assessment of the uncertainty in the leg model and especially the robot-leg interaction should be characterized experimentally in future work (Chapter 6) as a part of the validation process.



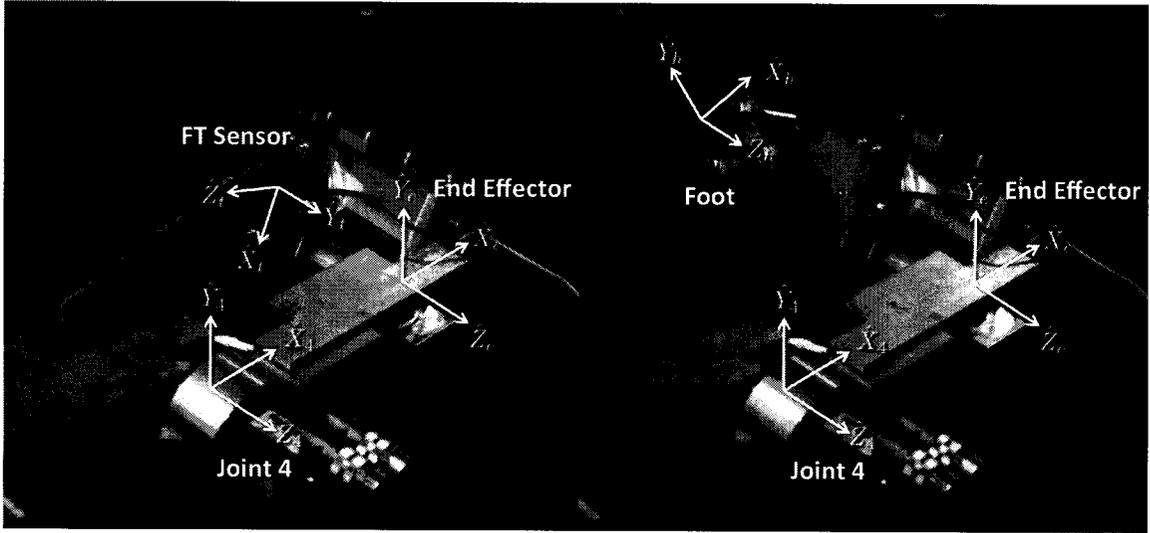
(a) Link 4, end effector and FT sensor frames

(b) Hand grip frame

(c) Hand holding grip

**Figure 2.19:** Link 4 axes with FT sensor and hand coordinate frames.

The coordinate frame at the end of the foot, denoted by subscript  $h$ , can also be referred to more generically as the human interaction frame attached to the robot. In order to visualize the transformations between the end effector frame (subscript  $e$ ), human interaction frame (subscript  $h$ ), and force torque (FT) sensor (subscript  $t$ ), Figure 2.19 shows these coordinate frames attached to the 4th link with a hand grip as the human interaction frame. Figure 2.20 shows the foot release mechanism and where the foot plate will be placed corresponding to the frame  $h$  in the ViGRR leg model. The transformations between each of these frames is characterized by a translation and orientation. Kinematic and FT transformations are required for calculating the end effector and human forces and trajectories from the FT sensor and robot kinematics. The six axis FT sensor is transformed to the end effector for all 6 DoFs with the tool transformation from the above section. However, the robot is planar, so only in-plane motion and forces are considered when transforming from



(a) Link 4, end effector and FT sensor frames

(b) Foot coordinate frame

**Figure 2.20:** Link 4 axes with FT sensor and foot coordinate frames.

the end effector to the human interaction frame.

The kinematic transformations between the end effector and human interaction frame are in plane for a rigid body. The fixed planar position and orientation of the foot coordinate frame relative to the end effector frame is defined by  $P_{eh} = [x_{eh} \ y_{eh} \ \psi_{eh}]^T$ . The kinematic transformations from end effector to foot are (2.58),

$$\begin{aligned} X_h &= X_e + {}^0P_h = X_e + {}^0R_e {}^eP_h \\ \dot{X}_h &= \dot{X}_e + {}^0\dot{R}_e {}^eP_h \\ \ddot{X}_h &= \ddot{X}_e + {}^0\ddot{R}_e {}^eP_h, \end{aligned} \quad (2.58)$$

where  ${}^0R_e = R_Z(\psi_e)$  is the rotation matrix about the  $\hat{Z}_0$  axis from the end effector frame to the fixed global frame by angle  $\psi_e$ .

In the above equations, it is assumed that the hip joint and the base of the robot are at the same location. If the hip position is offset by the displacement  $X_\delta = [x_\delta, y_\delta, 0]^T$  relative to the robot base frame, the position kinematics (2.58) can be modified to include the fixed displacement as in

$$X_h = X_e + X_\delta + {}^0R_e {}^eP_h. \quad (2.59)$$

Rigid body force and torque transformations can now be applied. The planar leg forces and torques  $f_h$  can be calculated by transforming the forces and torques at the end effector  $[F_e, T_e]^T$  to the leg end effector frame  $[F_h, T_h]^T$  using the tool transform (2.56) derived from the components of  ${}^eP_h$  then rotating the force vector  $F_h$  so that it is given relative to the base frame orientation:

$${}^0F_h = {}^0R_h F_h = R_Z(\psi_e + \psi_{eh})F_h. \quad (2.60)$$

Finally,  $f_h$  is composed of the  $x$  and  $y$  components of  ${}^0F_h$  and the  $z$  component of  $T_h$ .

The coordinate frame transformation for the ViGRR prototype with a hand grip at the end effector is defined by the vector  $P_{eh} = [-0.1023, 0.0564, -0.4189]^T$ . The tool transformation from the end effector coordinate frame to the hand is

$${}^hT_e = \begin{bmatrix} 0.9135 & -0.4067 & 0 & 0 & 0 & 0 \\ 0.4067 & 0.9135 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -0.0618 & -0.1389 & -0.0099 & 0.9135 & -0.4067 & 0 \\ 0.1389 & -0.0618 & -0.1164 & 0.4067 & 0.9135 & 0 \\ 0.0564 & 0.1023 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.61)$$

and the tool transformation from the FT sensor tool frame to end effector frame is

$${}^eT_t = \begin{bmatrix} -0.4067 & 0 & -0.9135 & 0 & 0 & 0 \\ -0.9135 & 0 & 0.4067 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.0252 & 0 & -0.4067 & 0 & -0.9135 \\ 0 & 0.0328 & 0 & -0.9135 & 0 & 0.4067 \\ 0.0402 & 0 & 0.0097 & 0 & 1 & 0 \end{bmatrix}. \quad (2.62)$$

### 2.3.7 Robot Model Verification

The same procedure for the leg dynamics verification (Section 2.2.6) is applied to the robot equations of motion. The iterative Newton-Euler formulation was used to

derive the dynamics independently from the Euler-Lagrange method for verification of the ViGRR equations of motion. The ViGRR forward and inverse kinematics were also verified.

## 2.4 Summary

The leg model and gait trajectories detailed in this Chapter provide the basis for the design requirements in Chapter 3. The robot model is used for detailed design of the robot and is also essential for the ViGRR model-based controller approaches and calibration as described in Chapter 4.

## Chapter 3

# Robot Design and Apparatus

In this chapter, the leg model with appropriate trajectories from average gait data is used for generating ViGRR design requirements. These requirements are then applied to the iterative hardware selection procedure and the hardware specifications and implementation details are finalized.

### 3.1 Requirements

As a lower extremity rehabilitation device, the performance requirements of the robot are largely based on a person's gait. Ambulation is not only a desirable functional outcome for gait rehabilitation, but is also involves high loads, a large workspace, and fast motion kinematics. This makes the average gait trajectories and the leg model from Chapter 2 ideal for generating required workspace loading requirements to be used for simulation and design of the robot. Other considerations are discussed in this section, such as safety, real-time control, and human factors.

#### 3.1.1 Workspace

The workspace of the robot is defined by the limits of movement in the lower extremities and the foot trajectory during gait. The primary purpose of the robot is to

assist gait rehabilitation for a large spectrum of individuals and capabilities. To this end, statistical studies of the stature and mass of individuals in a typical population are used to determine the leg properties of users for the robot. Table 3.1 summarizes statistical data for height and mass of adult Americans based on a survey conducted between 2003 and 2006 [75]. The 95th percentile male and 5th percentile female heights and mean masses are used to determine the required bounds, trajectories and loading in the workspace. Anthropometry relationships presented in Section 2.2.4 are applied to provide leg properties for the required workspace bounds and end effector loading of the 3R manipulator leg model.

**Table 3.1:** Weight and height of American males and females between 2003 and 2006.

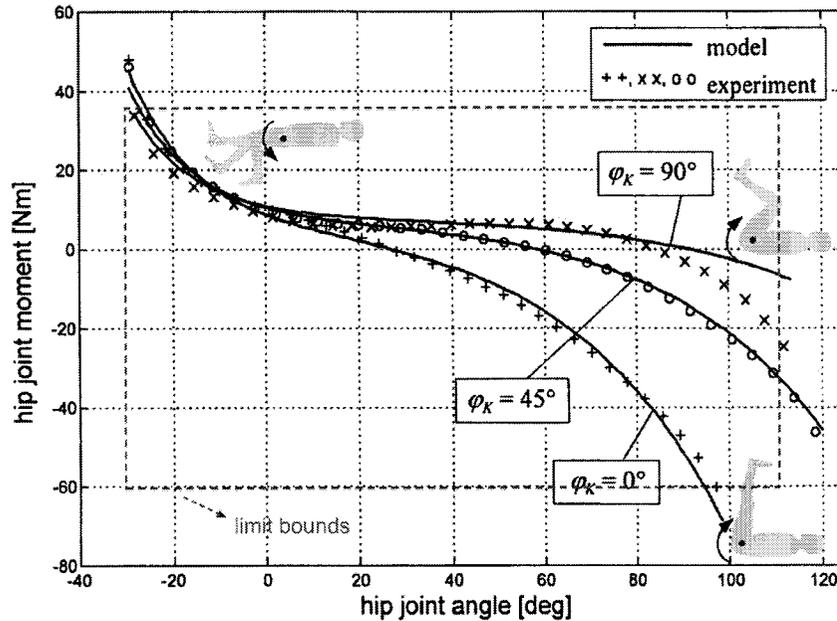
Group (Aged 20 and over)	Mass M (kg)			Height H (cm)		
	5%	Mean	95%	5%	Mean	95%
Males (N = 4,482)	62.2	88.3	122.6	163.6	176.3	188.7
Females (N=4,857)	50.5	74.7	113.6	150.7	162.2	173.1

### Workspace Boundary

A range of configurations of the user from standing upright to a fully reclined posture of the torso is desired. An analysis of the workspace requirements will focus on a single torso angle of 90 degrees (user lying flat). For other torso angles, the positions and angles of the foot can simply be rotated about the base coordinate frame. The chosen workspace bounds are based on limits of the joint angles and passive joint torques. Joint angle limits alone are not sufficient to account for the workspace since a person's range of motion varies depending on the angles of the other joints. For example, if the hip is flexed, the extension of the knee is much more limited compared to when the thigh is aligned with the torso.

To account for the variable joint limits, the passive joint torque model from [70]

introduced in Section 2.2.3 is used with torque limits to define the required workspace boundary. These limits are provided in Table 3.2. These joint torques and angles follow biomechanical conventions. In order to visualize how these torque and joint limits are applied, the hip, knee, and ankle passive joint torque stiffness curves from [70] are shown in Figures 3.1, 3.2, and 3.3 with the bounding joint angle and torque limits displayed visually as a dotted line box. For example, in Figure 3.2 if a knee joint angle limit of 3 degrees is chosen and the hip is at 120 degrees, the passive restoring torque would be enormous according to the model and it is unlikely anyone could reach such an extreme leg joint configuration. By limiting the passive knee torque in the model to 15 Nm, this unrealistic leg configuration would not be considered for the required workspace.



**Figure 3.1:** Passive hip joint moments and bounded limits for workspace requirements [70].

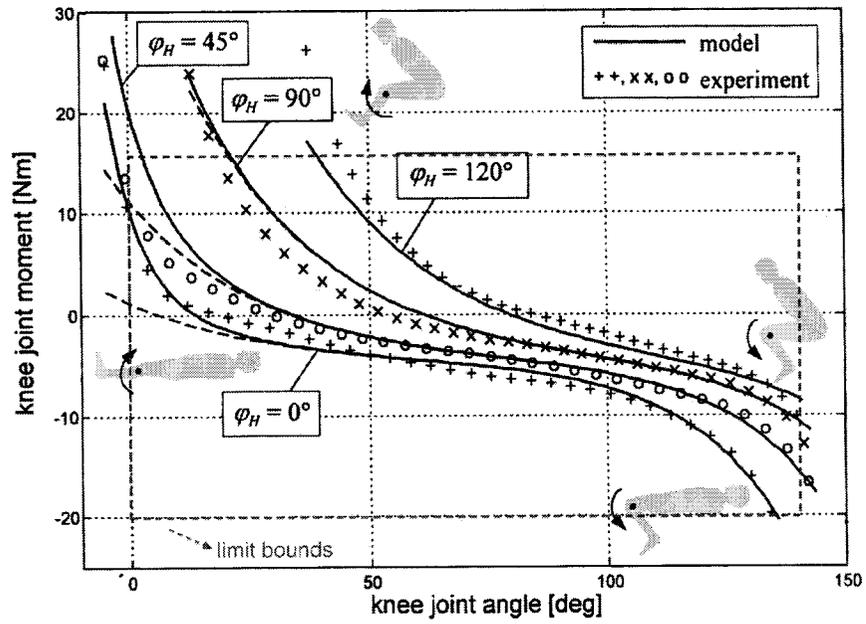


Figure 3.2: Passive knee joint moments and bounded limits for workspace requirements [70].

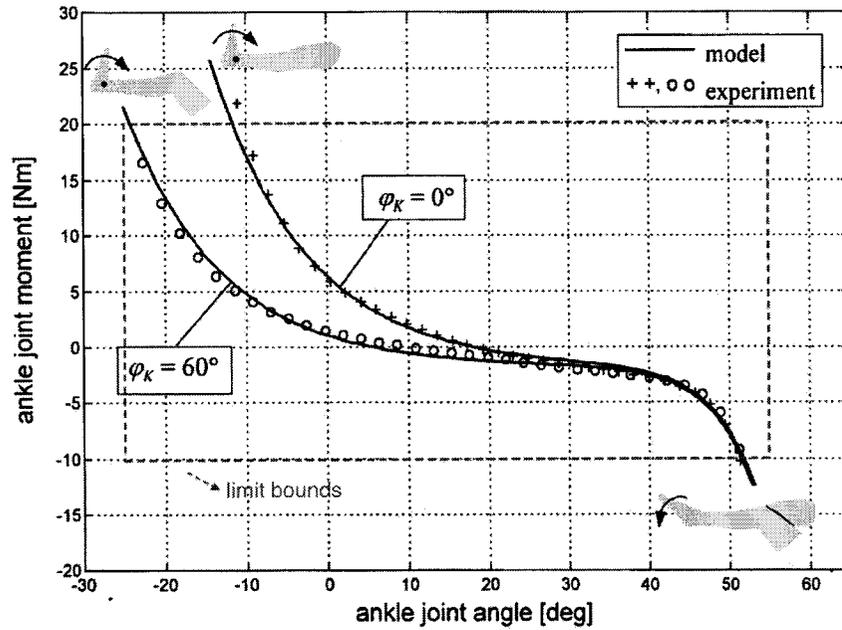
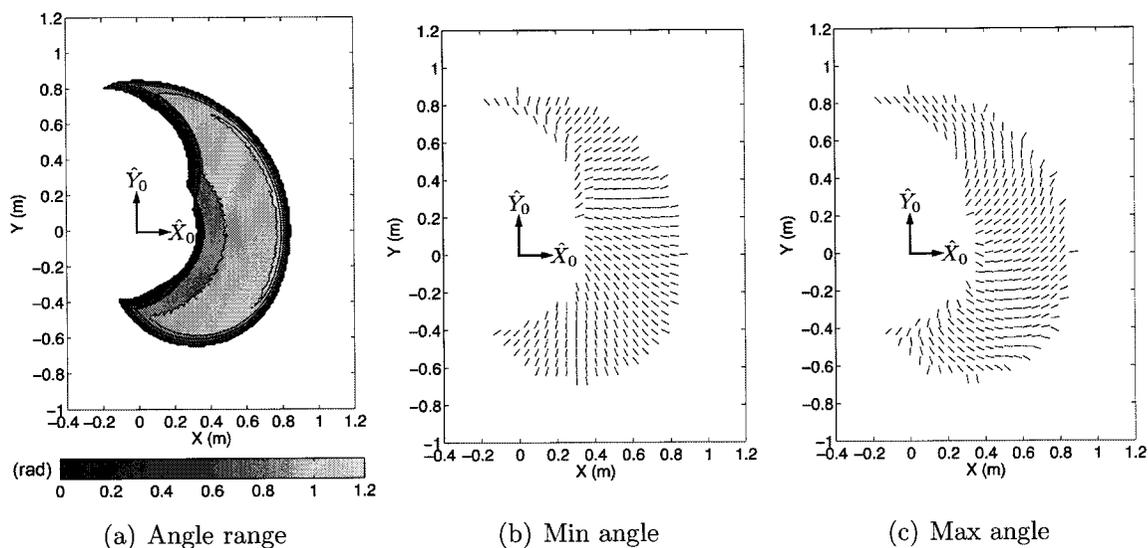


Figure 3.3: Passive ankle joint moments and bounded limits for workspace requirements [70].

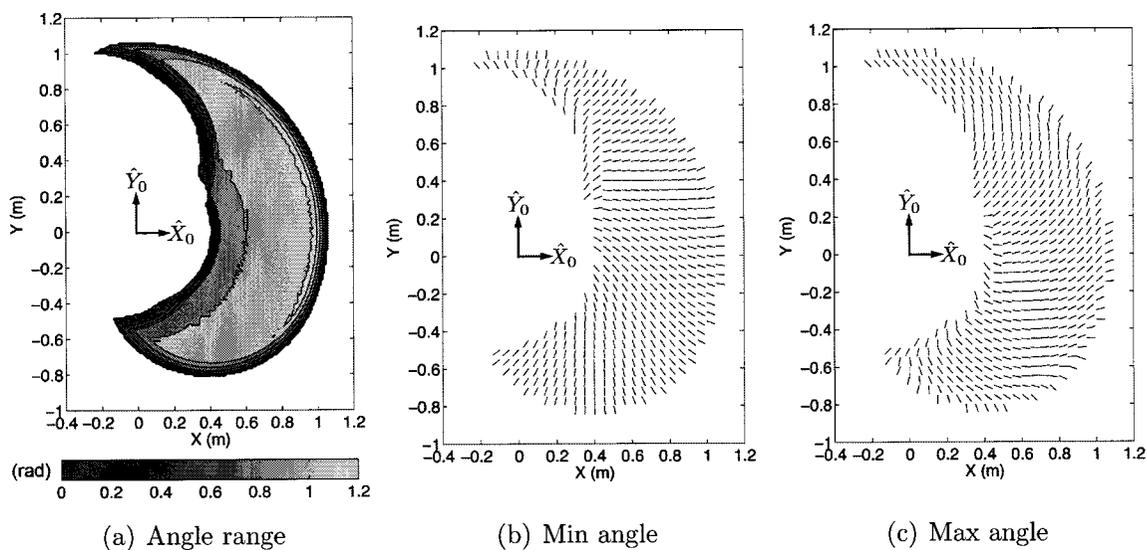
**Table 3.2:** Angle and Torque limits for leg joints.

	Joint	Flexion	Extension
Angles	Hip	110 deg	-30 deg
	Knee	140 deg	0 deg
	Ankle	-20 deg (dorsi)	55 deg (plantar)
Torques	Hip	-60 Nm	35 Nm
	Knee	-20 Nm	15 Nm
	Ankle	-10 Nm (dorsi)	20 Nm (plantar)

To determine the workspace bounds, the leg model inverse kinematics were applied at foot locations and orientations in the Cartesian space surrounding the user with a grid of points tested at 1 cm intervals for the position and 1 degree intervals for the orientation. The 95th percentile male and 5th percentile female heights and leg models were used and passive torques at each joint were calculated. If the passive torque or joint limits were exceeded, that point was set as incompatible. If there was a compatible configuration, the range, maximum, and minimum permissible foot angles were calculated. The results are presented in Figure 3.4 for an individual with a height of 1.887m (95th percentile male) and in Figure 3.5 for an individual with a height of 1.507m (5th percentile female). These two body heights and associated anthropometric limb segment lengths represent the maximum and minimum heights of individuals required for use with the ViGRR prototype.



**Figure 3.4:** 5th percentile female height leg model workspace bounds.



**Figure 3.5:** 95th percentile male height leg model workspace bounds.

Near the edges of the workspace, the maximum and minimum foot angles are very similar, having a much smaller set of permissible configurations. From the contour plots, the angle range approaches zero as expected.

### 3.1.2 End Effector Loading Requirements

The end effector loading requirements for supporting gait motion can now be calculated based only on the cadence and chosen height and mass of the individual(s) for use of the robot. Forces and torques applied at the end of the foot  $f_h$  (at the 5th metatarsal) can be calculated from average leg joint trajectories and applied torques by rearranging the leg model dynamics (2.24) to give

$$f_h = J_h^{-T} \left( -M(\theta)\ddot{\theta} - B(\theta)\dot{\theta}\dot{\theta} - C(\theta)\dot{\theta}^2 - G(\theta) + \tau_{pass} + \tau_h \right). \quad (3.1)$$

Because the Jacobian is inverted, the result becomes poorly conditioned at a configuration where the leg model is close to a singularity when the knee angle approaches zero degrees. During average gait motion, the knee angle is close to zero degrees at heel strike. This is expected as it conserves energy, where little torque at the knee can support a large ground reaction force. However, this does pose some numerical issues when calculating the end effector forces from (3.1). In order to diminish this effect, the joint angle trajectory of the knee is truncated slightly to limit its maximum value. In the following analysis, the maximum knee angle is limited to  $\theta_2 = -3$  degrees. The process for determining the loading and trajectory requirements of the end effector is as follows:

1. Leg model parameters are determined from height and mass of individual with anthropometric relationships defined in Section 2.2.4;
2. Average angular positions, velocities, and accelerations are calculated from cadence and spline fit in Figures 2.5, 2.6, and 2.7;
3. End effector positions, velocities and accelerations calculated from forward kinematics in Section 2.2.2;
4. Average applied joint torques calculated from spline fit in Figures 2.8, 2.9, and

2.10;

5. Passive joint torques are calculated from (2.23);
6. Joint torques, positions velocities and accelerations are used in (3.1) to obtain the end effector forces and torques;

### 3.1.3 Visualizing the Workspace

A Matlab graphical user interface (GUI) was created to visualize the workspace requirements and simplify the process of generating end effector trajectories for detailed design of the robot. This program provides several capabilities that make it a versatile and powerful tool in the design process. Foot trajectory data relative to Cartesian coordinates is displayed using three distinct visual representations: An animated leg with coloured foot trajectory, a colorbar indicator, and a plot of the current variable being visualized. The GUI also provides export options of the generated end effector data. A screenshot of the GUI is shown in Figure 3.6 with the components of the GUI labeled by numbers 1 to 7. The detailed capabilities of the GUI are explained for each of the components in reference to Figure 3.6 as follows:

1. Panel for specifying the required properties to generate the leg model and gait trajectory. Pushing the “Apply” button recalculates the trajectory according to the specified parameters. The end effector Cartesian space positions, velocities, accelerations, forces, and torques are calculated from the process described in Section 3.1.2 for one gait cycle. The inputs required to generate the end effector data are:

**Height (m):** Height of the individual;

**Mass (kg):** Mass of the individual;

**Points:** Number of data points to calculate over one gait cycle;

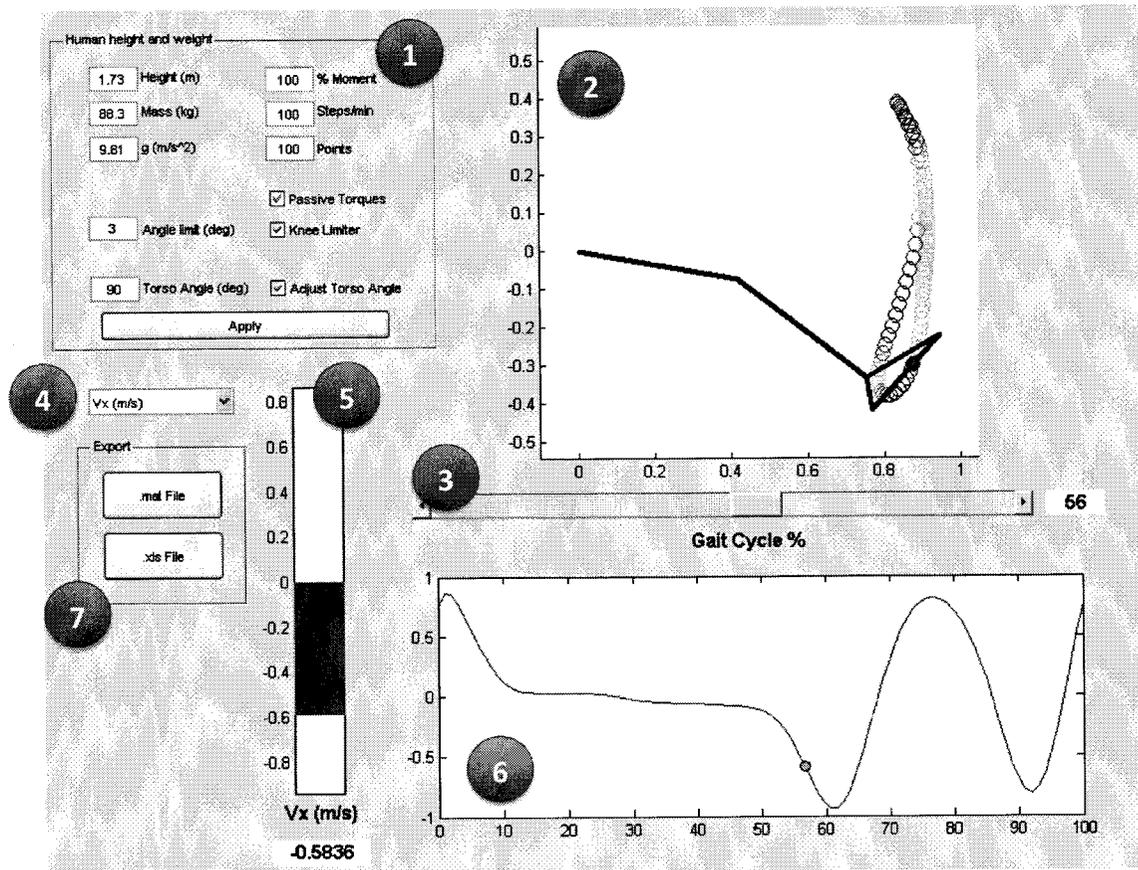


Figure 3.6: Workspace Visualization GUI.

**% Moment:** Percent of average applied joint moments;

**Steps/min:** Cadence of the gait;

**Torso Angle (deg):** Torso angle of individual relative to fixed frame;

**Angle Limit (deg):** Maximum knee angle limit. This can be switched on or off with the Knee Limiter checkbox;

**Passive Joint Torques:** A checkbox to include passive joint torques in the calculations.

2. Plot of a stick figure representation of the leg and the end effector trajectory coloured to scale of selected variable being analyzed. The position of the leg changes and the end effector circle is filled according to the current percent of

gait cycle. The leg is animated and moves as the gait percent slider is moved.

3. Slider used to choose the current percent of gait cycle.
4. Drop down menu to select a variable for visualization. Variables included in the drop-down list:
  - Vx**: Velocity in the x direction in m/s;
  - Vy**: Velocity in the y direction in m/s;
  - Vphi**: Angular velocity in rad/s;
  - Ax**: Velocity in the x direction in m/s<sup>2</sup>;
  - Ay**: Velocity in the y direction in m/s<sup>2</sup>;
  - Aphi**: Angular velocity in rad/s<sup>2</sup>;
  - Fx**: Force in the x direction in N;
  - Fy**: Force in the y direction in N;
  - T**: Torque about the z axis in Nm.
5. A bar scale showing the current variable value at the specified percent of gait cycle. The Y value limits of the scale bar are fixed to the maximum and minimum values of the variable over the entire gait cycle. The bar changes colour according to the colour displayed on the stick figure plot.
6. A plot of the selected variable over the entire gait cycle. The current gait cycle value is highlighted by a green marker.
7. Panel to export all end effector data to a Matlab variable file or Excel spreadsheet. The exported data includes all the inputs (Height, mass, cadence, etc.), leg model parameters ( $m_i$ ,  $l_i$ , etc.), and Cartesian foot trajectory data (positions, velocities, accelerations, forces and torques). A Matlab data file or Excel spreadsheet can be saved.

### 3.1.4 Safety Requirements

Robots with human interaction require a great deal of care with regards to safety. Several safety mechanisms must be built in to the design and remain a priority during the development and operation. A list and descriptions of generic safety concerns in regards to human-robot interaction are as follows

**Collisions** Collisions must be avoided between the robot and itself, its static surroundings, and any person(s) within its workspace. The controller must be able to avoid collisions. However, hard stops and mechanical trips are essential, due to the unpredictable nature of human participation and potential controller instability or hardware complications.

**Hyper extension or flexion** Having a person's leg being manipulated by the robot introduces the potential for injury by pushing the person's joints beyond their comfort or injury threshold limits. The controller should be able to accept sensory information of the leg joint angles to avoid the leg joint limits.

**Emergency or panic events** In the case of pain or any emergency, a panic button should be available for the user and therapist which will disable the robot and disengage the user safely. Unlatching the user from the robot should be sufficient to prevent injury.

**Velocity and dynamic limits** Velocities and torques of the robot and human leg should be measurable and programmable so that their limits are not exceeded.

### 3.1.5 Controller Hardware and Software Considerations

The software for the controller must be highly flexible to allow for implementing complex digital control schemes. As a research-focused device, the controller should be

easily modified for quick prototype implementations with data logging and visualization tools being necessary for the development and visual feedback of the system. A list of general requirements are as follows:

- A hard real-time computing environment is essential for controller stability with deterministic computation at a fixed step time. A frequency of 1000 Hz is required for the system;
- The data acquisition device should have an appropriate number of analog inputs and outputs, digital IOs and encoder inputs to suit the sensor and actuator needs of the system. At least 4 analog outputs for the motor drive commands and 6 analog inputs for the FT sensor are required. Eight digital inputs are required for limit switches, one digital input and one output for emergency event monitoring such as stop buttons and impact bars, and 16 digital outputs for motor drive units commands.
- A force/torque sensor is required to provide force feedback (haptic) control schemes as dictated by the objectives for the prototype device.
- System should be scalable to a full scale implementation with more DoFs and a second robot for the other leg.

Detailed specifications of the hardware are explored in the design and apparatus of the robot described throughout the following sections.

### **3.1.6 Human Factors Considerations**

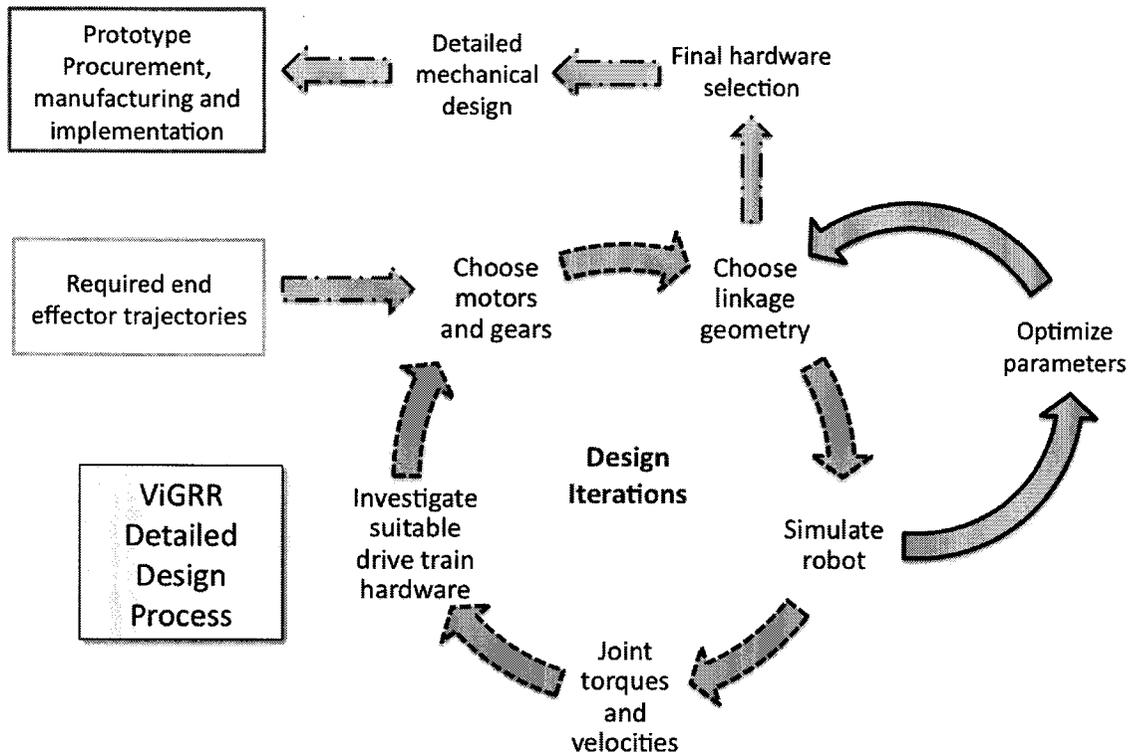
As a device focused on interacting with people, there are a number of human factors that affect the implementation of the device. The ViGR concept is motivated by the potential to improve the outcomes of patients through the rehabilitation process in a way that is less time consuming and burdensome on the therapist and patient than

conventional therapies. A second motivation is for better tools in assessment of the patient that the therapist can use to implement improved therapy strategies. If such a device were to be designed, the aesthetics, noise, ease of use, approachability and accessibility would be primary to the device's development. However, for the single leg prototype on a relatively short development cycle, the manufacturing time, cost, and flexibility of the mechanical design takes precedence. With a shorter development time, focus can be made on creating novel controllers and performing preliminary tests on healthy subjects before moving to a full scale clinical implementation.

## 3.2 ViGRR Robot Design

The design process for the ViGRR prototype is shown as a flow chart in Figure 3.7. The configuration of the robot was chosen and motivated in Section 1.2.2 and the simulations in this chapter are based on the leg and robot models presented in Chapter 2. Simplifying assumptions covered in Section 3.2.1 were made regarding the parameters of the robot model in a simulation covering the end effector trajectories. The basic simulation was used in the design iterations for optimization routines and to generate the final hardware and robot parameters.

The design process shown in Figure 3.7 begins with the required end effector trajectories specified in Section 3.1. Initial transmission hardware (motors and gears) and linkage geometry were selected. The first design iteration process used the selected robot parameters in simulation and the joint torques and velocities were analyzed over the end effector trajectories. Transmissions were selected to meet the requirements of the joint torque-velocity curves and the process was iterated until appropriate hardware was chosen and finalized for the robot. During the design iterations, a non-linear optimization algorithm employed an objective function to automatically



**Figure 3.7:** Overview of the design process for ViGRR.

determine optimal linkage lengths for the ViGRR prototype. The optimization routine is represented by the solid line arrows in Figure 3.7 and is detailed in Section 3.2.3.

Once the hardware was selected, the links and joints were designed and analyzed using CAD software and FEM analysis. This completes the ViGRR prototype design before the hardware was procured and the robot was manufactured (Red text box in Figure 3.7). A detailed description of the ViGRR apparatus are provided in Section 3.4.

### 3.2.1 Basic Simulation Modelling

The model of the robot detailed in Section 2.3 was used to find the motor joint torques  $\tau_{m_k}$  and velocities  $\theta_{m_k}$  for each set of end effector loading requirements. Four

DC brushless servo motors, gears with ratios  $r_k$ , linkage lengths  $l_i$ , angle offset at end effector  $\psi_{eh}$  and redundancy resolution scheme constants were selected. Only the singularity avoidance and joint limit avoidance schemes were applied in the motor and gear selection process. All the properties and dynamic calculations of the robot model were based on these selected components with simplifications and assumptions about the design described in the following list.

**Drive train inertia ( $I_{m_k}$ )** The inertia  $I_{m_k}$  is assumed to be the sum of the motor armature and gear inertia published in the data sheets of the selected components.

**Drive train damping ( $B_{m_k}$ )** The harmonic drive units do have published parameters for a viscous friction model in use with ViGRR. Given the lack of available model data, an estimate of the viscous friction coefficients was set to be  $0.001 \text{ Nm}/(\text{rad}/\text{s})$  for purposes of simulation for the design process. This value is based on results from a parameter identification for a brushless DC motor with a CSF-20 harmonic drive and a gear ratio of 160 in [77].

**Mass properties ( $m_i, l_{Ci}, I_i$ )** The mass properties of each link were determined based on the motor and gear masses at the joint and an ideal rigid bar made of 2024 T6 Aluminum with 2 inch outer diameter tubing and 1/4 inch wall thickness. The tubing was based on early CAD models of the robot mechanical design. The gear and motor assembly for joint  $i$  was assumed to be a point mass on the end of link  $i - 1$  and the center of mass location, second moment of inertia, and masses were calculated accordingly. A 20% mass increase was applied to the center of mass in order to account for extra material and fasteners expected to be used for the mechanical design.

**Joint limits** Joint limits were set to  $\pm 140$  degrees. These values were based on the

serial planar design which limits the angle of the links as self collisions would occur beyond these joint angles.

**Coulomb friction** ( $T_{c_k}(\dot{q})$ ) Coulomb friction was assumed to be negligible for the motor and gear selection process.

### 3.2.2 Drive Unit Sizing Requirements

Four sets of end effector trajectories and loading requirements were used with the robot model to select the appropriate drive train components based on joint torques and velocities. The trajectories are based on walking at 100 steps per minute with 60 percent of the average applied leg joint torques and leg properties derived from body height and weight specifications in Table 3.3. These values correspond to the requirements provided in Section 3.1.1, where the 0 and 30 degree torso angles correspond to the most extreme configurations of the user in terms of high loading requirements compared to an upright 90 degree orientation.

**Table 3.3:** Properties for generating end effector loading requirements used in drive train selection.

Trajectory	Body Height (m)	Body Mass (kg)	Torso Angle (deg)
a)	1.887	88.3	0
b)	1.887	88.3	30
c)	1.507	74.7	0
d)	1.507	74.7	30

### 3.2.3 Optimization

The loading requirements of the motor are dependant on the motor and gear properties themselves requiring an iterative approach to select them as well as to optimize the geometry of the robot. An initial brute force search for an optimal set of linkage

lengths and appropriate motor and drive combinations was performed (Design Iterations loop in Figure 3.7) so that the chosen transmissions are appropriately sized for further optimization routines to find optimal linkage lengths and other robot properties (loop in Figure 3.7 with solid arrow lines). This section describes the blue nonlinear optimization process to find ViGRR prototype link lengths ( $l_1, l_2, l_3, l_4$ ), end effector offset angle ( $\phi_{eh}$ ), and redundancy resolution scheme gains ( $k_{sing}^0, k_{lim}^0$ ). These variables form the vector of robot parameters to be optimized using an objective function.

The objective function is a result of simulating the robot given input required end effector trajectories and robot parameters. The joint torques and velocities are calculated over these trajectories from the robot model in Section 2.3 and simplifications in Section 3.2.1. The root mean square joint torques and peak absolute torques are included in an objective function to be minimized in the form of

$$\begin{aligned}
 f(\tau) &= f_{rms} + 0.5f_{max} & (3.2) \\
 f_{rms} &= \sqrt{\frac{1}{4n}\beta(\tau\tau^T)} \\
 f_{max} &= \beta \max_j |\tau_{i,j}|,
 \end{aligned}$$

where  $\tau$  is a matrix of ( $4 \times n$ ) elements with each row corresponding to its respective joint torques over all of the required end effector trajectories. Each required trajectory from Section 3.2.2 consists of 100 points over the gait cycle making a total of  $n = 400$  points. The objective function is a sum of the root mean square values of the joint torques and the maximum absolute value of each joint over the trajectory, both weighted by the unit vector  $\beta = [0, 0.2, 0.3, 0.5]$ . These weights were chosen to increase as the joints move further toward the end effector since the torque capabilities of the motor and drive unit are usually related to its size and mass, increasing the requirements of the previous joints that need to hold up its weight. The first joint is not optimized for its torque requirements since it is fixed to the base.

The objective function (3.2) is a result of a complex function of the linkage lengths, redundancy resolution scheme gains and end effector angle offset. In an attempt to find an optimal solution to these design parameters, Matlab’s genetic algorithm toolbox was applied. Genetic algorithms are based on natural selection, where a population of individuals (ie. set of test points for the objective function) is generated at each iteration based on a set of rules. These rules are based on selection, crossover, and mutations. Selection rules decide which parents from the previous population will survive and contribute to the next population. Survival depends on the fitness (ie. objective function) of the individuals (test points). Crossover rules decide how children are created from two parents in the current iteration. Mutation rules apply random changes to the parents and children. The iterations continue as the population moves (or evolves) toward an optimal solution [78].

The Matlab function “ga” accepts bounds on the input parameters, and nonlinear inequalities. The genetic algorithm rules and options were left as default values. The bounds of the input variables are given in Table 3.4. A nonlinear inequality function was also created that marked the function parameters as invalid if the trajectory had an incompatible configuration or the specified joint limits of  $\pm 140$  degrees were exceeded.

**Table 3.4:** Nonlinear optimization objective function input parameter bounds (search space).

Parameter	Minimum	Maximum
$l_1$	0.1 m	0.7 m
$l_2$	0.1 m	0.7 m
$l_3$	0.1 m	0.7 m
$l_4$	0.1 m	0.4 m
$k_{lim}^0$	0	2
$k_{sing}^0$	0	2
$\phi_{eh}$	-10 deg	90 deg

**Table 3.5:** Nonlinear optimization results for design parameters.

$l_1$ (m)	$l_2$ (m)	$l_3$ (m)	$l_4$ (m)	$k_{sing}^0$	$k_{lim}^0$	$\psi_{eh}$ (rad)
0.2005	0.3441	0.2749	0.1515	0.0962	0.0165	0.5842

The resulting parameters from the genetic algorithm optimization are given in Table 3.5. There are limitations in this optimization routine. Simplifications in the linkage parameters and robot model were a source of error in determining the torque and velocity requirements of the motor. The “ga” function in MATLAB may also present a local optimal, or sub-optimal set of parameters. The default settings for mutations, crossover heuristics, survival rates, etc. were used. This made the algorithm a “black box” solution, where the details of the algorithm were not investigated or tuned. Although this may have provided a sub-optimal solution, further time invested in optimizing the parameters for the robot was deemed unnecessary. The final compatible parameters optimized by the genetic algorithm with the chosen motor and harmonic drive units provided a practical solution to the optimization problem. The next step of the ViGRR design process, described in the next section, is finalizing the drive train components (Figure 3.7).

### 3.2.4 Drive Train Components

The final DC motor drive units and harmonic drives were tested in simulation with the optimized parameters. Maximum and nominal torque and velocity limits for the chosen motor and drive combinations were compared to the maximum and root mean squared values from the computed trajectory to ensure the drive units were appropriately sized. Table 3.6 lists the components selected for the robot. The paired motor and drive units are from Danaher-Motion and Harmonic Drives were used for the joint transmissions. Data sheets relevant to the design process for the motors and harmonic drives can be found in Appendices B and C. The product selection

guides [79, 80] provide more detailed specifications. The following section provides details regarding the S200 base drive units for controlling the AKM motors. Detailed specifications for the s200 drive units are given with the product manual [81].

**Table 3.6:** Selected Components. Detailed specifications in [79–81].

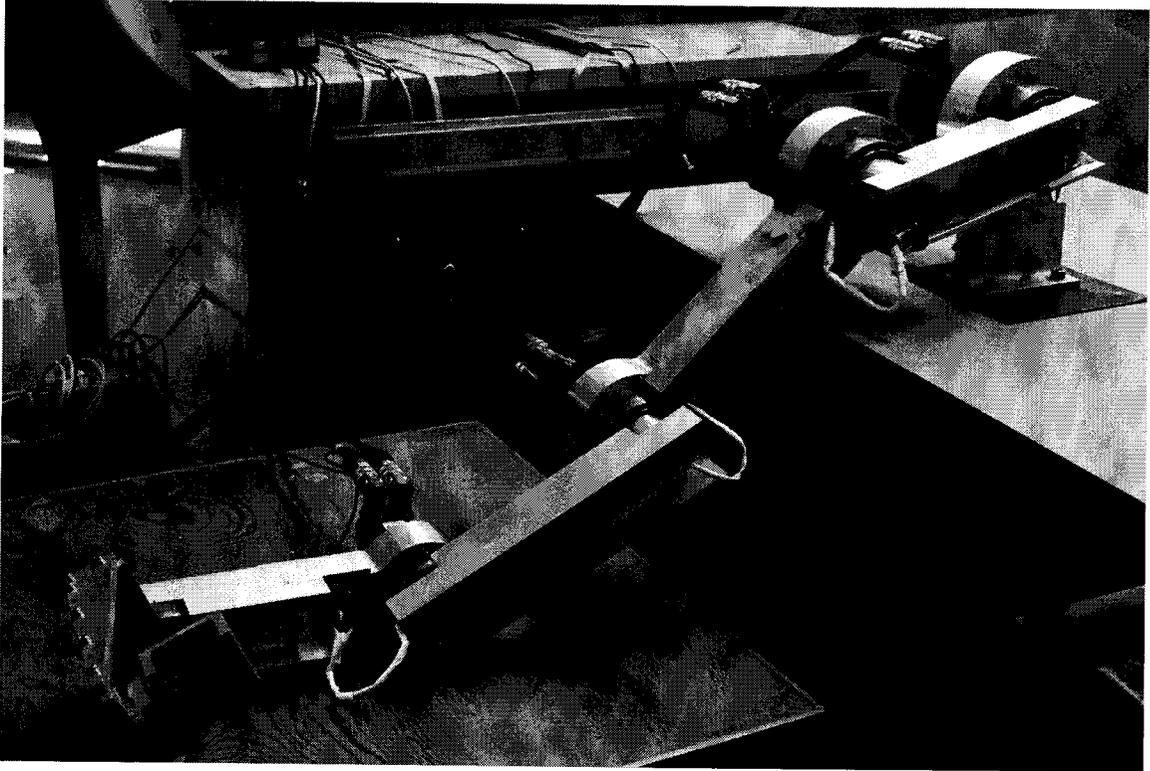
Joint	Drive Unit (s200)	Motor	Harmonic Drive
1	S21260-VTS	AKM42J-ACCNC-00	CSF-32-120 ( $r_1 = 120$ )
2	S21260-VTS	AKM43K-ACCNC-00	CSF-32-120 ( $r_2 = 120$ )
3	S20360-VTS	AKM22E-ACBNC-00	CSG-25-160 ( $r_3 = 160$ )
4	S20360-VTS	AKM22E-ACBNC-00	CSG-20-100 ( $r_4 = 100$ )

### 3.3 Mechanical Design

The final design step before implementation is the detailed mechanical design of the robot (Figure 3.7) which was completed by a colleague Adam Mullins. This process involved an FEM analysis, material selection, and creative use of i-beams as stock material for the linkages. The design and manufacturing using i-beams saved time, money and made the manufacturing process easier. The joint connections were CNC milled. A detailed analysis of the mechanical design and manufacturing was made. The final assembled robot joints and links are shown in Figure 3.8. Details of the mechanical design process are beyond the scope of this thesis.

### 3.4 Hardware Overview

Overview of the Robot Hardware setup is given in Figure 3.9. This chapter describes the relevant specifications, wiring and setup of the apparatus.

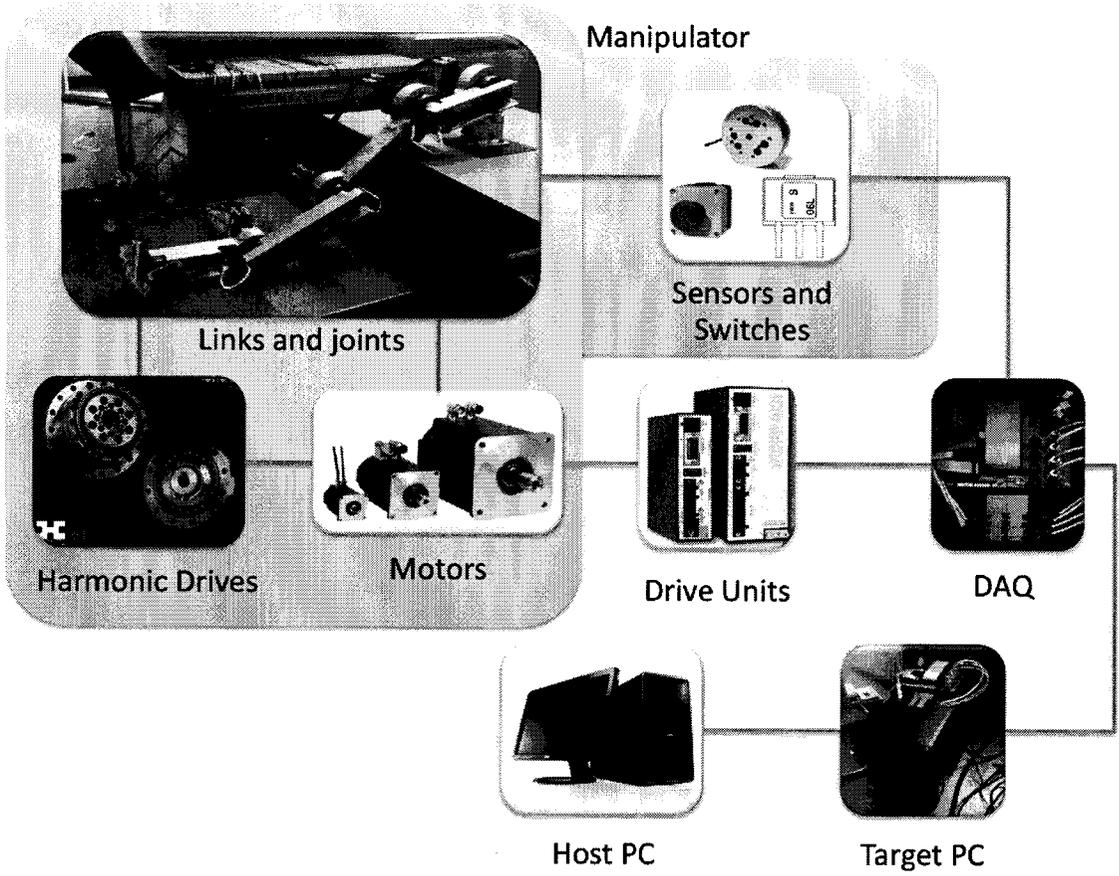


**Figure 3.8:** Assembled joints and links of ViGRR prototype.

### 3.4.1 Danaher Motion S200 Base Units

The Danaher Motion S200 base units boast advanced functionality for controlling the motors. The units incorporate a “Smart Feedback Device” (SFD) which communicates between the motor and base unit through a feedback cable. The SFD can provide information on the motor specifications and automatically set up command and controller feedback gains, depending on the control mode of the base unit.

A command IO connector allows for wiring and communication between the base unit and the data acquisition board (DAQ) from the real-time controller. When operating in current command mode, an analog  $\pm 10$  Volt input into the command IO from the DAQ specifies the desired armature current in the DC motor. The base unit uses the control scheme given by Figure 3.10 [81]. The descriptions of the variables in this control scheme are provided in Table 3.7. The command voltage bias



**Figure 3.9:** Topographical Layout of Robot Apparatus.

offset (CmdOffset) from the DAQ card can be set automatically by the S200 base unit when a zero voltage command is sent to the drive.

The motor torque output  $\tau_{m_k}$  for joint  $k$  is related to the armature current  $A_k$  and the motor constant  $K_{m_k}$  by

$$\tau_{m_k} = K_{m_k} A_k = \frac{u_k}{r_k}. \quad (3.3)$$

In order to command the desired output torque  $u_k$  of joint  $k$ , the real-time controller must output a scaled voltage  $V_{out_k}$  from the DAQ using

$$V_{out_k} = \frac{A_k}{K_{a_k}} = \frac{u_k}{r_k K_a K_{m_k}}, \quad (3.4)$$

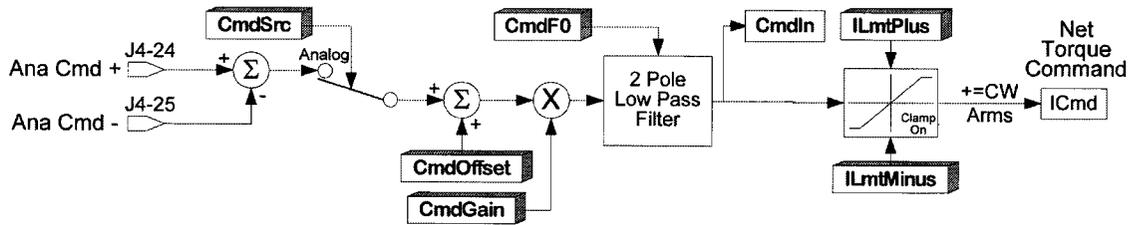


Figure 3.10: S200 drive unit torque mode controller [81].

Table 3.7: S200 base unit current controller parameters.

Parameter	Description
CmdSrc	Sets the command mode of the drive unit. For the robot, this is always set to current/torque mode
CmdOffset	A voltage offset for the input command. This can be set automatically by the s200 base unit with a digital input is set to high when a zero voltage command is sent to the base unit.
CmdGain	The scaling factor from command voltage to desired current in $A_{rms}/V$
CmdF0	Low pass filter cutoff frequency
ILmtPlus, ILmtMinus	Minimum and maximum armature current saturation for the motor. These are set automatically by the drive unit to ensure safe operation of the motor

where  $K_{a_k}$  is the CmdGain variable scaling the command voltage to armature current in the S200 base unit controller.

The drive unit also outputs an 4X quadrature emulated encoder with programmable pulses per revolution  $E_{PPR_k}$ . Two double-ended counter channels and an index pulse signal are sent to the DAQ card. The joint output angle  $q_k$  in radians for joint  $k$  is calculated from the set pulses per revolution of the emulated encoder, the pulse count from the DAQ  $N_k$ , and gear ratio  $r_k$  using:

$$q_k = \frac{2\pi N_k}{r_k 4E_{PPR_k}}. \quad (3.5)$$

Another notable feature of the S200 base units is the capability to inhibit torque being supplied to the motor in either the clockwise or counter clockwise direction by setting digital inputs to high through the command IO connector. This feature is useful for joint limit sensors that are fed back to the base units when triggered. A list of the relevant command IO connector pin assignments on the base unit are provided in Appendix A. The data acquisition card wiring descriptions in the following section correspond to the appropriate inputs and outputs of this connector.

**Table 3.8:** Drive unit properties.

Property	Joint	Value	Dimension
Nominal and Max Torques	$\tau_{1nom}, \tau_{1max}$	254.52, 892	Nm
	$\tau_{2nom}, \tau_{2max}$	281, 892	Nm
	$\tau_{3nom}, \tau_{3max}$	90.72, 309.12	Nm
	$\tau_{4nom}, \tau_{4max}$	56.7, 191	Nm
Nominal and Max Velocities	$\dot{q}_{1nom}, \dot{q}_{1max}$	2.618, 5.236	rad/s
	$\dot{q}_{2nom}, \dot{q}_{2max}$	2.182, 5.2359	rad/s
	$\dot{q}_{3nom}, \dot{q}_{3max}$	2.291, 4.9087	rad/s
	$\dot{q}_{4nom}, \dot{q}_{4max}$	3.665, 8.378	rad/s
Torque Constants	$K_{m1}$	0.41992	Nm/Amp
	$K_{m2}$	0.51172	Nm/Amp
	$K_{m3}$	0.31641	Nm/Amp
	$K_{m4}$	0.31641	Nm/Amp
CmdGain	$K_{a1}$	3.0001497	Amp/Volt
	$K_{a2}$	3.0001497	Amp/Volt
	$K_{a3}$	0.9000449	Amp/Volt
	$K_{a4}$	0.7496511	Amp/Volt
Encoder pulses per revolution	$E_{PPR1}$	6250	Pulses/Revolution
	$E_{PPR2}$	6250	Pulses/Revolution
	$E_{PPR3}$	4096	Pulses/Revolution
	$E_{PPR4}$	4096	Pulses/Revolution

In summary, the output capabilities of the joint transmissions and relevant parameters for controlling the torque outputs of the motors are given in Table 3.8. The nominal and maximum torques and velocities are based on the gear ratios and performance specifications of the motors and harmonic drives (See appendices B and C).

### 3.4.2 ATI Force/Torque Sensor

An ATI Industrial Automation 6 axis force/torque (FT) sensor shown in Figure 3.11 is used at the end effector. The model is a Mini85 transducer with the data acquisition option for signal processing and interfacing with a DAQ card. The installation and operation manual detail the sensor's general specifications, data acquisition terminal connections, operating ranges, and mounting details [82]. The sensor range, resolution, and maximum relative full-scale errors are summarized in Table 3.9 for the SI-1900-80 calibration. The calibration accuracy report with percent errors for various loading conditions performed during calibration are provided in Appendix E. The calibration has a much higher force range than is required by the end effector loading requirements provided a lower resolution. However, the lower force calibration for this FT model with improved resolution also specifies half the torque range which does not meet the loading requirements at the end effector. This sensor was best suited to meet the loading requirements.

**Table 3.9:** Mini85 FT sensor ranges and specifications for the SI-1900-80 calibration.

	$F_x, F_y$ (N)	$F_z$ (N)	$T_x, T_y$ (Nm)	$T_z$ (Nm)
Sensing Ranges	1900	3800	80	80
Resolution	1/3	5/12	9/700	3/350
Percent error (%)	1.5	1.0	1.25	1.25

The FT sensor signal conditioning base unit interfaces between the Mini85 and



**Figure 3.11:** The Mini85 force/torque sensor.

the DAQ. The unit requires an unregulated +5V DC power supply and outputs 6 analog voltages to the DAQ corresponding to the strain gauges from the sensor. A calibration matrix  $C_t$  is applied to the input DAQ voltages in order to get the 6-axis forces and torques measured in N and Nm using

$$\begin{Bmatrix} F_t \\ T_t \end{Bmatrix} = C_t V_{in}, \quad (3.6)$$

where

$$C_t = \begin{bmatrix} 5.609 & -0.742 & -4.234 & 180.893 & 1.880 & -186.620 \\ 2.563 & -214.831 & 2.510 & 105.190 & -6.322 & 108.442 \\ -239.440 & -2.775 & -234.178 & -1.427 & -239.572 & -3.096 \\ -0.043 & -0.032 & 7.687 & 0.041 & -7.783 & -0.092 \\ -8.981 & -0.086 & 4.358 & 0.013 & 4.424 & 0.048 \\ -0.049 & 4.921 & -0.074 & 4.827 & -0.031 & 5.019 \end{bmatrix}.$$

### 3.4.3 Safety Mechanisms

There are several mechanical and electromechanical safety apparatus implemented for the prototype robot. Very powerful motors and a large capable workspace of the robot can cause severe injury and it is important to minimize the chances of injury to the participant using the device. The footplate design, magnetic release mechanism, and impact bars described in the following list were designed and manufactured by Adam Mullins. The list of safety mechanisms are provided as follows:

**Emergency stop buttons** Two stop buttons are available for the user and the supervisor, or robot operator. They are connected in series to a monitoring input of the DAQ. If a button is pressed, the circuit is opened and the robot motors become disabled;

**Magnetic release foot plate** The mechanical interface between the foot and robot is supported by an electromagnet and power supply. If an emergency button is pressed, the circuit supplying the power to the electromagnet is cut and the footplate is released from the robot;

**Mechanical release foot plate** If too much load is applied to the foot, a passive mechanical release serves to have the foot fall away from the robot. In addition, the emergency stop is triggered in the case of such an event;

**Plexiglass shield** A plexiglass shield is installed to separate the user from the robot arm to prevent injury from pinch points;

**Impact bar** An vertical impact bar, placed at the edge of the plexiglass shield, prevents the robot end effector from reaching the user or colliding with the support structure. Upon impact with the bar, switches trigger the motors to stop;

**Software safety** A safety monitoring function for the robot controller is used to detect configurations, velocities, torques, and emergency stop situations and act accordingly. A description of these controls are provided in Section 4.4.4.

### 3.4.4 Quanser Real-Time Motion Control Platform

The real-time controller is implemented using a host Windows PC with Simulink and QuaRC software, a target PC with QNX hard real-time operating system, and a Quanser Q8 data acquisition card. The host computer is where the controller development is performed using Simulink and communicates with the target PC through TCP-IP communications protocol. Quanser's QuaRC is software integrated with Simulink and automatically generates real-time code on the QNX machine from Simulink models. QuaRC also provides tools for advanced visualization, communication, and integration with the Q8 data acquisition board.

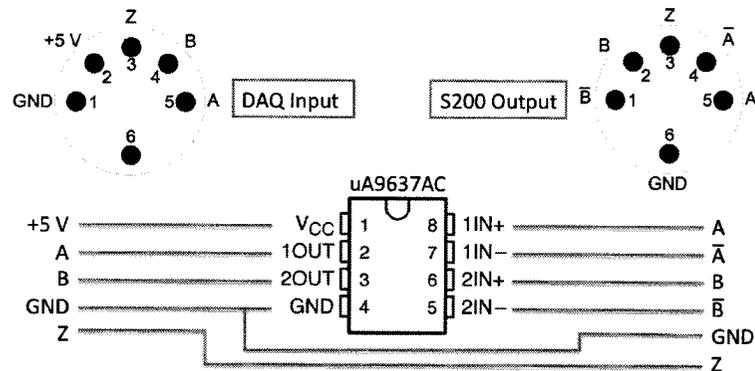
#### Q8 Data Acquisition Card and Breakout Board

The Quanser Q8 data acquisition card and breakout board has 32 digital I/Os, 8 analog inputs, 8 analog outputs, and 8 encoder inputs. The DAQ interfaces with the sensors and actuators of the robot. Figure 3.13 a) shows an image of the board with wired inputs and outputs for the robot. The breakout board features RCA connectors for the analog inputs and outputs, 5-pin DIN connectors of the encoder inputs, and ribbon connectors for the digital I/Os. An image of the Q8 DAQ with the labelled terminals used for the rehabilitation robot are shown in Figure 3.13 b) [83].

Description of the labeled inputs and outputs on the Q8 are provided in Appendix A. The Q8 specification sheet is provided in Appendix A [84].

### Differential to Single-Ended Encoder Conversion

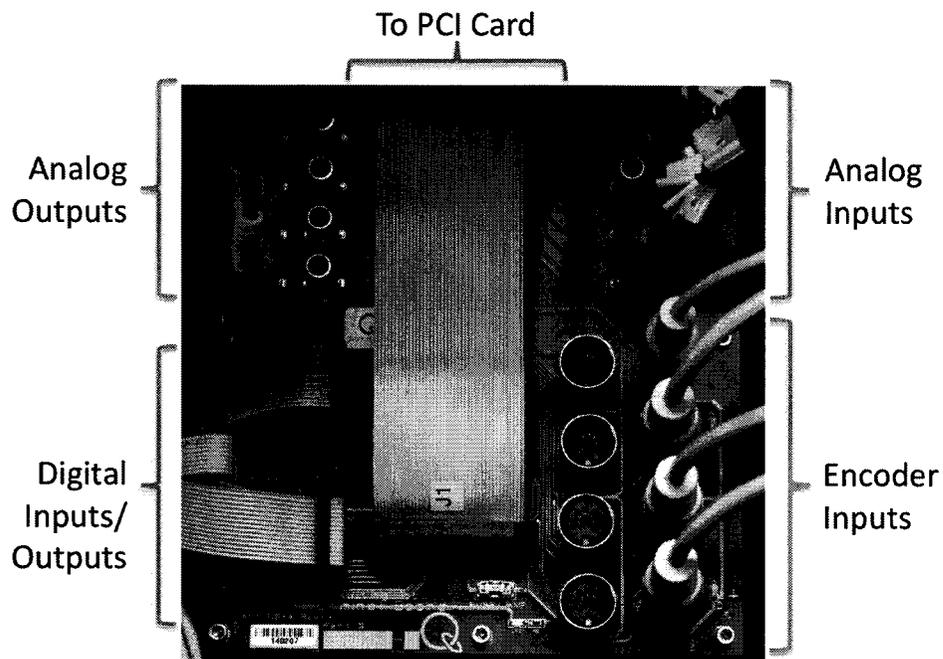
The DAQ encoder terminals are single-ended and the s200 drive units provided differential encoder outputs. A Texas Instruments uA9637AC line receiver is used to convert the A and B channel differential signals to single-ended. Figure 3.12 shows the wiring diagram for the line receiver. The cables and terminals connecting the encoder inputs on the DAQ to the s200 drive are 5-pin DIN cables with shielding. Figure 3.12 displays the associated pin definitions for these connectors. Detailed specifications of the uA9637AC are provided in [85].



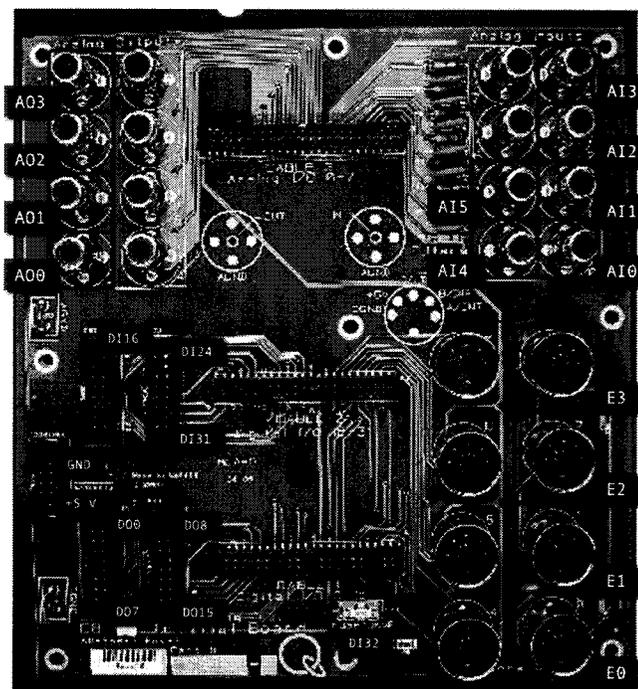
**Figure 3.12:** Wiring for Encoder conversion from differential to single-ended.

### Hall-Effect Proximity Sensors

Non-latching single pole Hall effect proximity sensors by Infineon Technologies were installed on the linkages to detect joint limits. These sensors trigger a high (+5 V) voltage when a single-pole magnetic field with enough strength passes through the sensor. At each joint, two small magnets are installed opposite Hall-effect sensors on opposing links for triggering both +ve and -ve joint limits. A diagram of the sensor



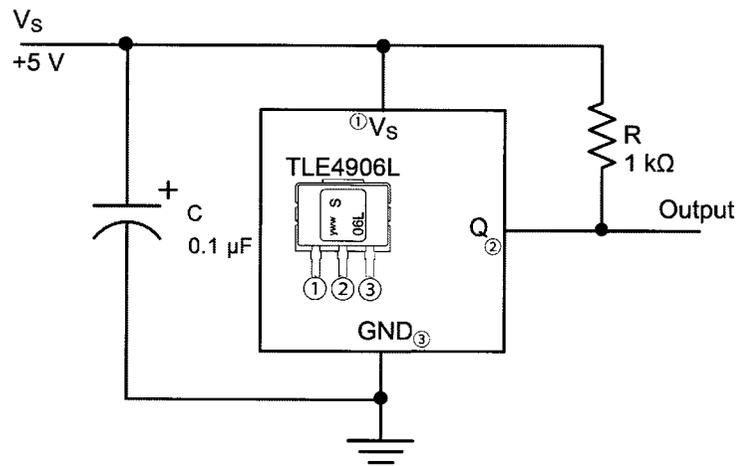
(a) The Q8 breakout board with wired terminals



(b) Terminal labels on Q8 in use with ViGRR [83]

**Figure 3.13:** The Quanser Q8 DAQ with wiring and terminal labels in use with ViGRR. Descriptions of terminals are provided in Appendix A.

and small circuit are given in Figure 3.14. The +5V voltage supply  $V_S$  and ground are wired to the DAQ supply voltage. The digital output Q from the sensor is wired to the digital inputs on the DAQ. In a small circuit wired close to the sensor (within 15 cm), a  $0.1 \mu F$  capacitor is installed between the supply voltage and ground to prevent jitter. A  $1 k\Omega$  resistor is wired between the supply voltage and output in order to limit the current drawn by the output transistor of the Hall effect integrated circuit. Detailed sensor specifications are provided in Appendix D [86].



**Figure 3.14:** Wiring for Hall-effect proximity sensors.

### 3.5 Summary

This chapter provided the details for the design and implementation of the ViGRR prototype robot to meet the design requirements. The integrated hardware setup and apparatus can now be utilized with a motion controller and tested with a user. Chapter 4 describes the controller theory and implementation as it interacts with the Quanser motion control system, drive units, sensors and mechanical apparatus.

## Chapter 4

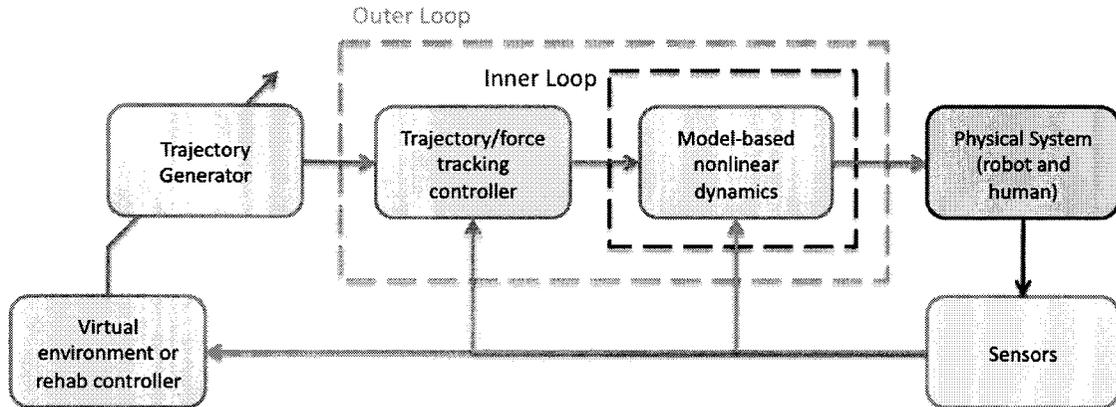
# Controller Implementation

Beyond the design, procurement and manufacturing of the ViGRR apparatus, the next phase of implementation is the development of the controller in order to fully realize the system. The low level model-based control of ViGRR is built alongside the higher level trajectory tracking and force control. The software is custom made for the ViGRR prototype, allowing for flexibility in the development of new control methods and experiments. The controllers and calibrations presented in this chapter and implemented with ViGRR provide the basis for experimentation in Chapter 5 and future research using the prototype platform.

The overall control scheme for ViGRR follows an inner-loop and outer-loop model shown in Figure 4.1 [63]. The inner loop controller compensates for nonlinear dynamics of the robot and the outer loop consists of the trajectory tracking algorithms. A trajectory generator provides the desired positions velocities and accelerations to be tracked by the robot. The tracking feedback outer loop can be based at the joint level with desired joint angular trajectories, or at the task space level with positions, velocities and accelerations of the end effector. External forces can also be included in the outer feedback loop in the task space.

In application to rehabilitation robotics and haptic devices, the controller may

consist of a rehabilitation controller or virtual environment where the interaction between human and environment is modelled in a “sandbox” simulation and outputs the desired motion or interaction force with the human. This motion is then transformed to a desired robot trajectory tracked by the controller.



**Figure 4.1:** Outer loop and inner loop controls architecture [63].

The overall controller implementation utilizes robotics theory related to task and joint space trajectory generators, nonlinear trajectory control, force control and calibration of the kinematic and dynamic parameters of the robot. This chapter provides the theory for control of the robot and the algorithms implemented with ViGRR. A graphical user interface, the Simulink control architecture, simulation, and robot calibration are also presented.

## 4.1 Trajectory Generators

Trajectory generators can be specified in the joint space  $q \in \mathbf{R}^4$  or task space  $X \in \mathbf{R}^3$ . Time-varying positions, velocities and accelerations are outputs from the trajectory generator to the outer loop tracking controllers shown in Figure 4.1. The type of trajectory generated can be decided by user input, virtual environment, or rehabilitation-focused control. The following trajectory generators were created for the ViGRR

prototype.

### 4.1.1 Joint Space

In order to provide smooth motion of the robot, a trajectory for the joints can incorporate continuous position, velocity and acceleration. Such a trajectory can take the form of a 5th order polynomial for the position as a function of time:

$$q(t) = \sum_{i=0}^5 a_i t^i, \quad (4.1)$$

where  $t$  is the time and  $a_i$  are the polynomial coefficients. Given an initial position, velocity, and acceleration at time  $t_0$  and final position, velocity, and acceleration at time  $t_f$ , a set of linear equations can take the form

$$\begin{Bmatrix} q_0 \\ \dot{q}_0 \\ \ddot{q}_0 \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{Bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{Bmatrix} = Sa, \quad (4.2)$$

where  $S$  is a  $6 \times 6$  time matrix. The coefficient vector  $a$  for the trajectory from time  $t_0$  to  $t_f$  is generated by taking the inverse of the time matrix  $S$  and multiplying by the vector of initial and final values for the trajectory. The trajectory can then simply be followed by providing the input time for the polynomial (4.1).

The 5th order trajectory generator is used to move the joints or end effector smoothly to a desired position over a specified time. It is also useful for moving to the initial position, velocity and acceleration of a cyclic trajectory or stop the trajectory by smoothly moving to a set position.

A sine wave generator was created with position, velocity, and acceleration outputs to help with the dynamic calibration of the robot. The bias, amplitude and frequency can be modified on the fly with a smooth transition from one set point of bias,

amplitude and frequency to another. This is accomplished by providing a 5th order blend to the bias and amplitudes as in (4.1) over a single time period of the sine wave. When the desired bias or amplitude is changed, their positions velocities and accelerations are calculated as their values are blended from the current values to the new desired values. The corresponding output position, velocity and acceleration ( $p_s$ ,  $v_s$ , and  $a_s$ ) are given by

$$\begin{aligned} p_s &= A_s \sin(\omega t) + B_s \\ v_s &= \dot{A}_s \sin(\omega t) + A_s \cos(\omega t)\omega + \dot{B}_s \\ a_s &= \ddot{A}_s \sin(\omega t) + 2\dot{A}_s \cos(\omega t)\omega - A_s \sin(\omega t)\omega^2 + \ddot{B}_s, \end{aligned} \quad (4.3)$$

where  $A_s$ ,  $B_s$ , and  $\omega_s$  are the amplitude, bias, and frequency of the sine wave, respectively. When the desired frequency is changed, the set frequency is modified when the velocity is equal to zero. This allows for the position and velocity output to remain continuous but provides a step acceleration.

### 4.1.2 Cartesian Space Trajectory Generators

The 5th order trajectory generator can also be applied to the task space with 3 degrees of freedom. Two other task space cyclic trajectory generators were created. The first is a circular trajectory with an oscillating sinusoid for the angle  $\psi_h$ . The second is a gait trajectory generator based on cadence, normal joint trajectories and limb lengths.

The circular trajectory is used as a proof of concept controller for performing a simple motor task which will be described later in Section 5.3 and for testing the end effector tracking performance. The trajectory is based on a number of parameters, described as follows:

- Circle centre ( $c_x$ ,  $c_y$ ) (m);

- Circle radius  $r_c$  (m) ;
- Frequency of periodic trajectory  $\omega_c$  (rad/s). The time it takes to complete one circle revolution is  $2\pi/\omega_c$ ;
- The bias end effector angle sinusoidal trajectory  $\psi_b$  (rad);
- Amplitude of the end effector angle sinusoidal trajectory  $\psi_a$  (rad) ;
- Phase offset of the end effector angle sinusoidal trajectory  $\psi_p$  (rad).

For the circular trajectory, the output position, velocity, and acceleration of the end effector are calculated using

$$\begin{aligned}
 X_h &= \begin{Bmatrix} c_x + r_c \cos(\omega_c t) \\ c_y + r_c \sin(\omega_c t) \\ \psi_b + \psi_a \sin(\omega_c t + \psi_p) \end{Bmatrix} \\
 \dot{X}_h &= \begin{Bmatrix} -r_c \omega_c \sin(\omega_c t) \\ r_c \omega_c \cos(\omega_c t) \\ \psi_a \cos(\omega_c t + \psi_p) \omega_c \end{Bmatrix} \\
 \ddot{X}_h &= \begin{Bmatrix} -r_c \omega_c^2 \cos(\omega_c t) \\ -r_c \omega_c^2 \sin(\omega_c t) \\ -\psi_a \sin(\omega_c t + \psi_p) \omega_c^2 \end{Bmatrix}.
 \end{aligned} \tag{4.4}$$

When first starting the circular trajectory, the end effector is moved to the initial position, velocity, and acceleration at  $t = 0$  using the 5th order trajectory generator.

A gait trajectory is based on the smoothed cubic spline fit for the average leg joint trajectories in Section 2.2.5. The cubic spline output joint angle positions, velocities and accelerations from (2.27) are converted to the foot ( $h$  coordinate frame) trajectory by the leg model forward kinematics (Section 2.2.2). Like the circular trajectory, when first starting the periodic motion the end effector is moved to the initial position, velocity, and acceleration at  $t_g = 0$  using the 5th order trajectory generator.

## 4.2 Model-Based Nonlinear Control

This section describes the controllers which encompass the inner and outer loops in Figure 4.1 used with the ViGRR robot. These control strategies employ sensor feedback and the dynamic model of the robot to track a desired joint trajectory. A modified formulation of the dynamics is first presented and two control methods are described: The inverse dynamics control and passivity based nonlinear control. The former cancels out the dynamics in the plant (robot model) to create a linearized system for simple feedback control and the latter employs a nonlinear controller that modifies the closed loop dynamics for robust and adaptive control laws.

### 4.2.1 Parameter-Regressor Formulation of the Dynamics

A useful property of the dynamics of a manipulator is the linearity in the dynamic parameters (masses, lengths, inertias, and centre of masses). The equations of motion can be rearranged to involve a regressor matrix  $Y(q, \dot{q}, \ddot{q})$  as a function of the joint positions, velocities and accelerations and a parameter vector  $\phi$  containing the dynamic parameters are related by

$$\begin{aligned}\tau &= M'(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + B_{\text{eff}}\dot{q} + T_c(\dot{q}) \\ &= Y(q, \dot{q}, \ddot{q})\phi .\end{aligned}\tag{4.5}$$

This form simplifies the equations of motion and is useful for performing a dynamic calibration of the robot to find the parameter  $\phi$  experimentally. It can also be used for robust and adaptive controllers which modify the parameter vector.

Using Matlab's symbolic toolbox, the ViGRR dynamics (4.5) were expanded as a function of  $\tau$ , simplified, and the dynamic parameters were collected. For the ViGRR prototype, there are 25 unique dynamic parameters. The parameter vector  $\phi$  was found to be

$$\phi = \left\{ \begin{array}{l} (I_1 + l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4 + l_{C1}^2 m_1 + I_{\text{eff}1}) \\ (I_3 + l_3^2 m_4 + l_{C3}^2 m_3) \\ (I_4 + l_{C4}^2 m_4) \\ (I_2 + l_2^2 m_3 + l_2^2 m_4 + l_{C2}^2 m_2) \\ (l_{C4} g m_4) \\ (l_2 l_3 m_4 + l_2 l_{C3} m_3) \\ (l_2 g m_3 + l_2 g m_4 + l_{C2} g m_2) \\ (l_3 l_{C4} m_4) \\ (l_1 g m_2 + l_1 g m_3 + l_1 g m_4 + l_{C1} g m_1) \\ (l_3 g m_4 + l_{C3} g m_3) \\ (l_1 l_{C4} m_4) \\ (l_2 l_{C4} m_4) \\ (l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 l_{C2} m_2) \\ (l_1 l_3 m_4 + l_1 l_{C3} m_3) \\ B_{\text{eff}1} \\ B_{\text{eff}2} \\ B_{\text{eff}3} \\ B_{\text{eff}4} \\ \nu_{c1} \\ \nu_{c2} \\ \nu_{c3} \\ \nu_{c4} \\ I_{\text{eff}2} \\ I_{\text{eff}3} \\ I_{\text{eff}4} \end{array} \right\}. \quad (4.6)$$

The  $25 \times 4$  regressor matrix  $Y(q, \dot{q}, \ddot{q})$  is provided in Appendix F. The regressor and parameter formulation of the dynamics was verified by comparing it to calculations performed by the Euler-Lagrange and the iterative Newton-Euler equations of motion.

### 4.2.2 Inverse Dynamics

The dynamics of a serial manipulator is highly nonlinear. A technique used for nonlinear inner loop control (Figure 4.1) is the method of feedback linearization which can be applied by specifying a control law that results in a linear closed-loop system. It then becomes a simple task to design a controller based on classical control methods for linear systems. The following derivation is based on the inverse dynamics or

computed torque method for the inner loop controller described in [63].

By inspecting (2.55), we can introduce a controller of the form

$$u = M'(q)a_q + C(q, \dot{q})\dot{q} + G(q) + B_{\text{eff}}\dot{q} + T_c(\dot{q}) = Y(q, \dot{q}, a_q)\phi \quad (4.7)$$

to calculate the applied torque, where  $a_q$  as a new controller input. When (4.7) is substituted into (2.55), the dynamics of the system reduces to a closed loop linear system of the form

$$\ddot{q} = a_q. \quad (4.8)$$

The inner loop input  $a_q$  can now be chosen to produce four decoupled, closed-loop linear systems. An outer loop (Figure 4.1) joint space tracking controller can be specified as

$$a_q = \ddot{q}^d - K_1\dot{\tilde{q}} - K_0\tilde{q}, \quad (4.9)$$

where  $\tilde{q} = q - q^d$  and  $\dot{\tilde{q}} = \dot{q} - \dot{q}^d$  are the joint position and velocity errors, respectively. The positive diagonal matrices  $K_0$  and  $K_1$  specify the position and velocity gains for each joint. This is a proportional-derivative (PD) controller for tracking a trajectory with a feedforward acceleration term.

The closed-loop dynamics are

$$\ddot{\tilde{q}} + K_1\dot{\tilde{q}} + K_0\tilde{q} = 0. \quad (4.10)$$

By choosing a natural frequency  $\omega_k$  for each joint  $k$ , and setting the gains to  $K_{0_k} = \omega_k^2$  and  $K_{1_k} = 2\omega_k$ , the system response becomes a critically damped second order system for each joint (ie. the damping ratio  $\zeta$  is equal to 1).

### 4.2.3 Passivity-Based Nonlinear Control

An alternative to the inverse dynamics controls approach is a nonlinear controller based on the skew-symmetric properties of the Euler-Lagrange equations of motion [63]. This control does not necessarily pose advantages over the inverse dynamics method in terms of tracking performance, but does have advantages for robust and adaptive control. The controller is formulated as

$$u = M'(q)a + C(q, \dot{q})v + G(q) + B_{\text{eff}}\dot{q} + T_c(\dot{q}) - Kr, \quad (4.11)$$

where  $K$  is a positive diagonal gain matrix. The terms  $a$ ,  $v$ , and  $r$  are controller feedback variables defined by

$$\begin{aligned} v &= \dot{q}^d - \Lambda \tilde{q} \\ a &= \dot{v} = \ddot{q}^d - \Lambda \dot{\tilde{q}} \\ r &= \dot{q} - v = \dot{\tilde{q}} + \Lambda \tilde{q}, \end{aligned} \quad (4.12)$$

where  $\Lambda$  is a positive diagonal matrix. Substituting (4.11) into (2.55) yields

$$M'(q)\dot{r} + C(q, \dot{q})r + Kr = 0. \quad (4.13)$$

The resulting closed loop system is nonlinear. Error convergence to zero and stability analysis of this controller are provided in [63]. The advantage of such a system is that it lends itself to simplified robust and adaptive control laws compared to the inverse dynamics. The regressor-parameter formulation of the dynamics can be used in adaptive and robust control laws. For the passivity-based nonlinear control, (4.11) can be reformulated as:

$$u = Y(q, \dot{q}, a, v)\phi - Kr. \quad (4.14)$$

The regressor  $Y(q, \dot{q}, a, v)$  was created using the Matlab Symbolic toolbox and is provided in Appendix F. The analysis and implementation of robust or adaptive

control is beyond the scope of this thesis. Regardless, the controller was implemented for ViGRR with the potential of applying more advanced control laws in future work.

### 4.3 Task Space and Force Control

The task space control derivation in this section is based on the inverse dynamics control in Section 4.2.2 but employs the transformation between task space and joint space accelerations (2.37) to modify the inverse dynamics inner loop controller (4.7). By introducing the task space acceleration controller input term  $a_X$  and transforming it from task space to joint space using the pseudoinverse, we get the modified inner loop input term  $a_q$ , as in

$$a_q = J_r^\dagger (a_X - \dot{J}_r \dot{q}). \quad (4.15)$$

Substituting the acceleration differential kinematics (2.37) into (2.55) and rearranging to solve for  $\ddot{X}$  yields

$$\ddot{X} = \dot{J}_r \dot{q} + J_r M^{-1} (u - (C(q, \dot{q})\dot{q} + G(q) + B_{\text{eff}}\dot{q} + T_c(\dot{q}))). \quad (4.16)$$

Combining the input (4.15) and inverse dynamics inner loop controller (4.7), then substituting into (4.16) yields

$$\ddot{X} = a_X, \quad (4.17)$$

which is a simple linear system in the task space. For the outer loop task space trajectory controller, let the new input  $a_X$  be

$$a_X = \ddot{X}^d - K_1 \dot{\tilde{X}} - K_0 \tilde{X}, \quad (4.18)$$

where  $\tilde{X} = (X - X^d)$  and  $\dot{\tilde{X}} = (\dot{X} - \dot{X}^d)$  are the errors in the Cartesian space. We now have an uncoupled linear system with a closed loop PD controller and feedforward

acceleration term in the task space. The positive diagonal matrices  $K_0$  and  $K_1$  specify the position and velocity gains for the task space position and orientation. The closed loop tracking error dynamics are

$$\ddot{\tilde{X}} + K_1\dot{\tilde{X}} + K_0\tilde{X} = 0. \quad (4.19)$$

Not having to compute desired joint trajectories can be an advantage in terms of complexity and computation time. There is no need to invert the Jacobian, and in the case of redundant robotics, employ iterative optimization functions for the joint configuration. However, this is also a limitation of using this task space inverse dynamics approach since it does not employ optimizations that take advantage of the redundancy in the robot. The pseudoinverse in (4.15) minimizes the joint velocities, but the joint configuration is not optimized to avoid joint limits or singular configurations presented in Chapter 2.3.

Instead of applying this task space outer-loop controller, the joint trajectories can be calculated from the redundancy resolution scheme and inverse kinematics in Chapter 2.3 then set as inputs to the computed torque or passivity-based joint-level control.

### 4.3.1 Impedance Control

The discussion thus far has focused on trajectory (position) control for ViGRR. When interacting with an external environment, force control in the task space is required. With the external forces included in the inverse dynamics controls approach, the torque command is set as

$$u = Y(q, \dot{q}, a_q)\phi + J_r^T a_f, \quad (4.20)$$

where  $a_f$  is another control input we added. The external force  $f_e$  at the robot end effector is a disturbance to the system. If this force is measured using the FT

sensor, it can be compensated by the controller. If  $a_f$  is made to be equal to the measured external interaction force  $a_f = f_e$  then the linearized system is once again recovered (4.8). In this case, the force control can be handled entirely by the task space dynamics using the input  $a_X$  and (4.15). A common force control technique for use with the task space controller input  $a_X$  is impedance control. An impedance controller applies desired inertia  $M_I$ , viscous damping  $B_I$  and stiffness  $K_I$  to the task space dynamics. The impedance model involves measuring the error between a reference trajectory and the actual task space trajectory to find a net force according to the following dynamics:

$$-(f_h - f_h^r) = M_I \ddot{\tilde{X}} + B_I \dot{\tilde{X}} + K_I \tilde{X}, \quad (4.21)$$

where  $\tilde{X} = (X_h - X_h^r)$ ,  $\dot{\tilde{X}} = (\dot{X}_h - \dot{X}_h^r)$ , and  $\ddot{\tilde{X}} = (\ddot{X}_h - \ddot{X}_h^r)$  are the errors in the Cartesian space. Subscript  $r$  refers to a reference trajectory. As the trajectory deviates from the reference trajectory, a net force is applied to the user. The trajectory and forces are measured by the encoders and FT sensor of the robot. To apply the desired force back to the user, the input acceleration for the task space inner loop is set to

$$a_X = \ddot{X}_h^r - M_I^{-1}(B_I \dot{\tilde{X}} + K_I \tilde{X} + (f_h - f_h^r)) \quad (4.22)$$

so that the task space dynamics simplify back to the desired impedance model (4.21).

### 4.3.2 Admittance Control

The admittance task space model is similar to the impedance model and is given by

$$(f_h - f_h^r) = M_a \ddot{\tilde{X}}_a + B_a \dot{\tilde{X}}_a + K_a \tilde{X}_a. \quad (4.23)$$

In the admittance control scheme, the input is the measured force, and the output is the error trajectory generated as a disturbance to the reference trajectory with

a discrete-time integration process. The input to the inner loop is a task space trajectory

$$\begin{aligned} X_h^d &= X_h^r + \tilde{X}_a \\ \dot{X}_h^d &= \dot{X}_h^r + \dot{\tilde{X}}_a \\ \ddot{X}_h^d &= \ddot{X}_h^r + \ddot{\tilde{X}}_a, \end{aligned} \quad (4.24)$$

where the subscript  $a$  refers to the disturbance trajectory generated from the impedance model and superscript  $d$  refers to the desired trajectory generated for the outer loop task space PD control. The admittance disturbances are generated using the following state-space model:

$$\begin{Bmatrix} \tilde{X}_{a_{k+1}} \\ \dot{\tilde{X}}_{a_{k+1}} \\ \ddot{\tilde{X}}_{a_{k+1}} \end{Bmatrix} = \begin{bmatrix} I_{3 \times 3} & T_s I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & T_s I_{3 \times 3} \\ -K_a & -B_a & 0_{3 \times 3} \end{bmatrix} \begin{Bmatrix} \tilde{X}_{a_k} \\ \dot{\tilde{X}}_{a_k} \\ \ddot{\tilde{X}}_{a_k} \end{Bmatrix} + \begin{bmatrix} 0_{3 \times 3} \\ 0_{3 \times 3} \\ -M_a^{-1} \end{bmatrix} (f_h - f_h^r). \quad (4.25)$$

## 4.4 ViGRR Controller Software Implementation

The ViGRR hard real-time controller was developed in Simulink using QuaRC by Quanser Inc. for real-time code generation on a target PC with a QNX Neutrino operating system. The ViGRR controller features safety mechanisms, multiple control modes, visual feedback and a GUI. The controller is highly flexible and modifications to the code involve simply changing the Simulink model. The architecture of the ViGRR controller shown in Figure 4.2 is setup with four major modules: Emergency Stops and Enables, Sensors, Controllers, and Safety Block. These subsystems are described in the following sections, and the order of execution for the ViGRR controller is as follows:

1. The QuaRC Hardware In the Loop (HIL) initialization block initializes the DAQ input and output functions.

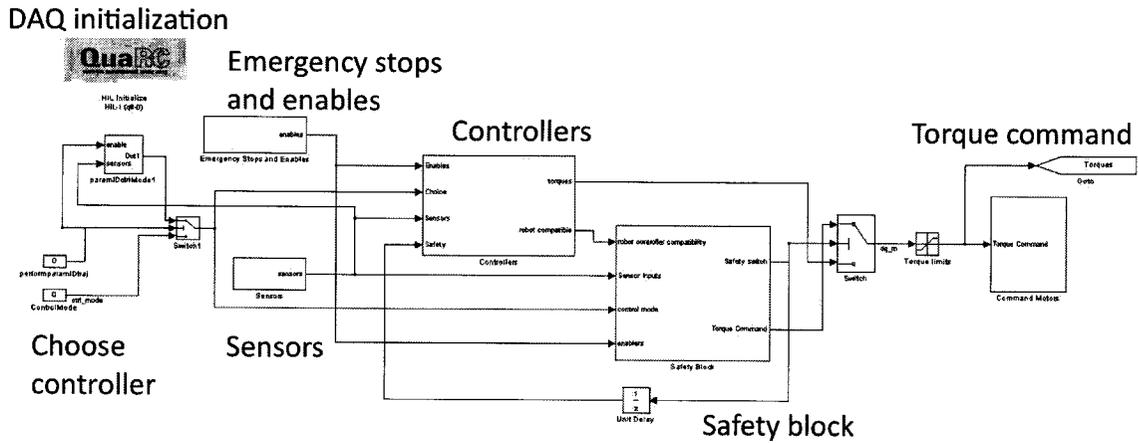


Figure 4.2: ViGRR Controller Simulink model.

2. Emergency triggers are monitored and motor enables are set.
3. Sensor data is obtained from the DAQ and processed.
4. The chosen control mode is enabled in the controllers block.
5. Safety limits and controller stability are monitored.
6. If a safety limit is triggered, the command torque is routed from the Safety Block. Otherwise, the enabled controller outputs a command torque.
7. The command joint torques are saturated to specified limits and scaled to a voltage command for the motor drive units.

The ViGRR controller also incorporates a graphical user interface (GUI) and utilizes a Simulink library created for the ViGRR prototype. The library features custom ViGRR blocks integrated into the Simulink library browser, providing subsystem masks for ease of use and wrapping of complex functions written in C.

### 4.4.1 Emergency Stops and Enables

This subsystem incorporates all emergency triggers from the emergency stop buttons, mechanical stops (ie. impact bar), and DAQ fuse monitoring. Upon triggering an emergency event, the motors are disabled and the magnet in the foot plate is released from the robot using a simple relay switch.

### 4.4.2 Sensors

The sensor block receives and processes all the sensor inputs from the DAQ and outputs a bus containing all the relevant data. The scaling, filtering, data logging, forward robot kinematics, joint limit switches, and transformations between coordinate frames are contained in this block. The desired trajectories from the controllers subsystem are also routed into this block for visualization and data logging. The output bus from this subsystem contains the variables  $q$ ,  $\dot{q}$ ,  $X_e$ ,  $\dot{X}_e$ ,  $X_h$ ,  $\dot{X}_h$ ,  $f_e$ ,  $f_h$ , and time  $t$ .

There is noise and quantization issues in the FT sensor and joint velocity calculation that requires filtering. The joint velocities are determined from the encoder counts and a finite backward difference discrete time differentiation. The FT sensor has a very high range with only a 0.5 N resolution in the  $z$  axis and there is significant noise in the FT sensor, especially when applying small loads. In order to filter the FT sensor and joint velocities, a 5th order butterworth finite impulse response (FIR) digital filter is implemented. The discrete time filter model is

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \dots + a(n+1)z^{-n}} \quad (4.26)$$

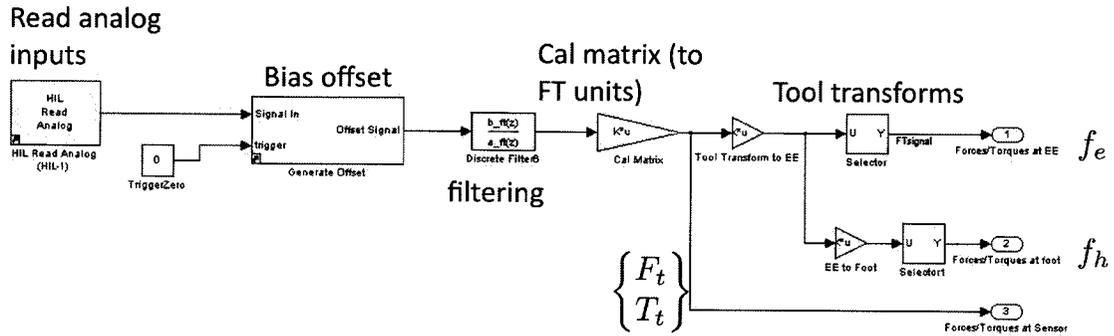
where  $B(z)$  is a polynomial of coefficients in vector  $b$  for the numerator that correspond to the filter zeros and  $A(z)$  is a polynomial of coefficients in vector  $a$  for the denominator that correspond to the filter poles.

A number of frequencies were tested for the encoder velocities and FT sensor. While the motors were running, noise from the S200 base unit ground was also observed to affect the FT sensor signal despite proper grounding of the hardware components. Cutoff frequencies of 3Hz, 5Hz, 15Hz and 30Hz were all tested with the admittance force controller and the 3Hz was still required for operation using a hand grip to prevent unwanted vibrations with reasonable task space controller dynamics. Table 4.1 displays the filter coefficients for the joint velocities and FT sensor with 225Hz and 3Hz cutoff frequencies, respectively.

There are a number of potential solutions for decreasing the FT sensor noise. One solution may be to recalibrate the sensor with a lower range at the tradeoff of being unable to effectively control the robot at limit loads. The orientation and placement of the FT sensor on the end effector may be optimized to avoid reaching the sensor limits according to the tool transformations and maximum loading at the human interaction frame. For example, rotating the FT sensor so that only the forces in tool frame x-y plane are used with the ViGRR planar prototype prevents the higher range z-axis and increased noise from being included in the control loop. The sensor is also currently wired to a single-ended analog input with one end of the FT differential signals terminated to the DAQ ground. A differential to single-ended analog circuit may be added to the hardware near the DAQ. The Q8 also supports hard-wiring capacitors to the analog input signals filtering.

**Table 4.1:** Sensor filter coefficients for joint velocities (225 Hz) and FT sensor (3 Hz).

Sensor	Coefficients
Joint Velocities	$b = [0.03490, 0.1745, 0.3490, 0.3490, 0.1745, 0.03490]$ $a = [1, -0.4923, 0.7183, -0.1733, 0.06885, -0.004679]$
FT Sensor	$b = [7.21e-11, 3.61e-10, 7.21e-10, 7.21e-10, 3.60e-10, 7.21e-11]$ $a = [1, -4.9390, 9.7579, -9.6395, 4.7615, -0.9408]$



**Figure 4.3:** Force/torque sensor subsystem.

A number of transformations are applied to the FT sensor analog inputs in order to obtain the forces and torques at the end effector and human interaction frame. A subsystem in the ViGRR controller Sensor block provides these transformations and is shown in Figure 4.3. In this subsystem, a bias offset for the input voltages from the FT sensor is generated, the signal is filtered, then the calibration matrix and tool transformations are applied. The bias offset averages the input voltages over a specified period of time when no load is applied and the procedure is triggered. The filtering with coefficients in Table 4.1 is applied and the voltages are then transformed to 6-axis force and torque loads from (3.6). Finally, the tool transforms (2.62) and (2.61) are applied to get  $f_e$  and  $f_h$ .

### 4.4.3 Controllers

This subsystem module enables controllers individually from a set of different control modes, routes the sensor information and output command torques, and performs redundant safety checks. The available control modes are:

#### Initialization Mode

In this control mode, the user can home the robot using the joint limit sensors, reset encoder values, hold or release joints, null the voltage command input to the drive

unit, and disable or enable the motors.

### **Hold Mode**

The hold function initiates a PD control and inverse dynamics feedback linearization (Section 4.2.2) with the current joint position as the reference command. This control mode is usually set after initializing the robot.

### **Joint Step Mode**

The joint step control mode requires an input joint configuration and trajectory time. A smooth 5th order trajectory for each joint is created to move from the current position to the desired joint angles over the specified length of time (Section 4.1.1). If the desired joint configuration is changed, the controller automatically moves to the new configuration.

### **Move to Optimal Configuration Mode**

This control mode must be initiated before an end effector trajectory is specified for the outer loop control. Selecting this control mode calculates the initial joint configuration via the redundancy resolution scheme for the current end effector position. The robot then performs a smooth self motion from its current joint configuration to the optimal joint positions. A detailed description of the redundancy resolution scheme is provided in Section 4.4.5.

### **End Effector Step Mode**

This control mode uses a Cartesian trajectory generator that functions similar to the joint step controller but creates the trajectory in task space. The redundancy resolution scheme is employed to convert to joint space trajectory and apply the PD control with the inverse dynamics inner loop. A detailed description of this control

mode is provided below for Figure 4.4. The end effector step trajectory can also be used to safely stop a cyclic trajectory by moving smoothly from its current position, velocity and acceleration to a stationary position.

### **Circle Trajectory Mode**

In this control mode, a task space circle trajectory is generated with the parameters described in Section 4.1.2. The desired end effector is smoothly moved from its current position to the initial position, velocity and acceleration of the circular trajectory, which is then initiated and run continuously.

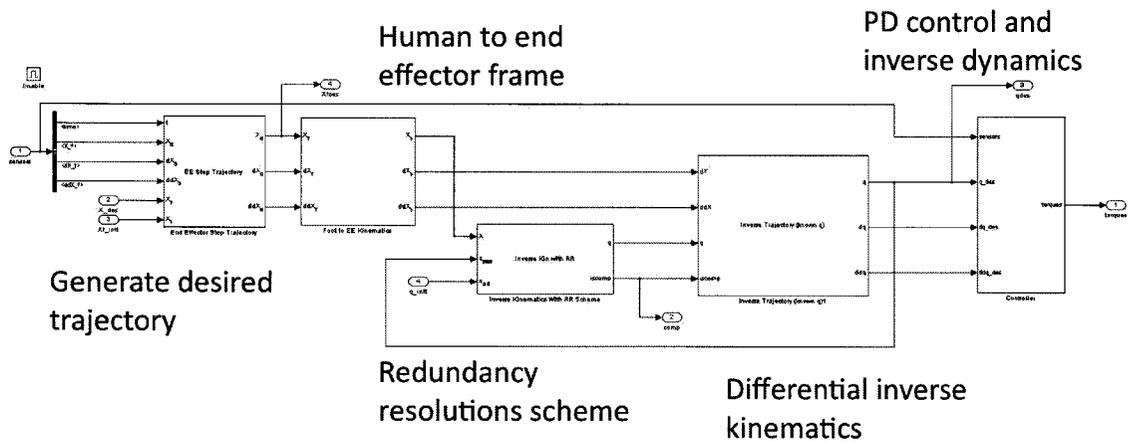
### **Admittance Control Mode**

The admittance model in Section 4.3.2 is applied in this control scheme, where the reference trajectory is the current end effector position with zero force, velocity and acceleration. The stiffness, damping, and inertia parameters are applied for the Cartesian space coordinates. A circular reference trajectory can also be applied, where a simple 2D plot visualization is provided to the user for tracking the trajectory.

### **Parameter Identification Mode**

This control scheme applied a smooth sine trajectory for each joint (Section 4.1.1) with varying amplitudes, bias and frequencies. The output torque and joint position data was automatically logged for post processing. Section 4.6.3 details the dynamic parameter calibration procedure.

The Cartesian space step input subsystem shown in Figure 4.4 is an example of a control mode implemented in the controller subsystem. The sensors are input into the block as well as initial positions and desired human coordinate frame pose. The 5th order smooth trajectory generator provides a trajectory from the initial current position to the desired position over the specified length of time. The human



**Figure 4.4:** ViGRR end effector step input enabled subsystem.

coordinate frame trajectory is then transformed to the robot end effector frame. The redundancy resolution scheme is used to find the optimal joint positions for the current end effector position and orientation. The inverse differential kinematics are then calculated and a joint space PD control is applied with inverse dynamics feedback linearization. The other control modes involving Cartesian space trajectories follow a similar procedure, where the transformation and inverse kinematics are applied before joint level control of the robot.

#### 4.4.4 Safety Controls

The safety block subsystem monitors the sensor data and holds the robot if a specified limit is reached. This block handles software limits and hardware joint limit switches. There are software safety limits in place for the joint and Cartesian space positions and velocities, forces and torques. These limits currently implemented on ViGRR are provided in Table 4.2 and were set for dynamic parameter identification and testing the force controller by hand. In addition, if a hardware (hall-effect proximity sensor) joint limit switch is triggered, motion is inhibited in that direction to prevent damage and self collision of the robot. The configuration of the robot is also continuously

monitored so that if a singularity is almost reached, the safety block is triggered and the robot is held using a simple PD controller for each joint. Triggering the safety block also automatically sets the controller to the initialization control mode, where the safety trip can be reset.

**Table 4.2:** Controller Safety Limits.

Parameter	Minimum Limit	Maximum Limit	Units
$q_1$	-0.2	3.288	(rad)
$q_2$	-2.26	2.23	(rad)
$q_3$	-2.67	2.68	(rad)
$q_4$	-2.35	2.22	(rad)
$\dot{q}_1$	-2.2	2.7	(rad/s)
$\dot{q}_2$	-2.5	2.5	(rad/s)
$\dot{q}_3$	-3	3	(rad/s)
$\dot{q}_4$	-3	3	(rad/s)
$x_h$	-1.5	1.5	(m)
$y_h$	-0.6	2	(m)
$\psi_h$	-2	3	(rad)
$\dot{x}_h$	-3	3	(m/s)
$\dot{y}_h$	-2.5	2.5	(m/s)
$\dot{\psi}_h$	-5	4.5	(rad/s)
$f_{h_x}$	-200	200	(N)
$f_{h_y}$	-200	200	(N)
$\tau_{h_z}$	-30	30	(Nm)
$\tau_1$	-240	250	(Nm)
$\tau_2$	-200	200	(Nm)
$\tau_3$	-100	100	(Nm)
$\tau_4$	-50	50	(Nm)

The safety block may also detect leg joint angles and velocities based on inverse kinematics from the ViGRR leg model so that leg joint limits can also trigger appropriate safety mechanisms. However, the ViGRR has not yet been tested with a person and the ability to determine leg joint angles from the robot kinematics has not yet been validated.

#### 4.4.5 Redundancy Resolution Scheme

The redundancy resolution scheme, introduced in Section 2.3.3, is applied for the Cartesian space controllers. A custom Simulink block was developed to embed the redundancy resolution scheme into the controls software. The Simulink block and expanded while loop subsystem is shown in Figure 4.5. The expanded subsystem (redundancy resolution while loop) shows the gradient nonlinear optimization while loop iterations according to the joint limit avoidance and singularity avoidance objective functions. Before and after the iterations subsystem, the joint configuration is checked to ensure it has not reached a singular configuration. The while loop iteration process is performed as follows:

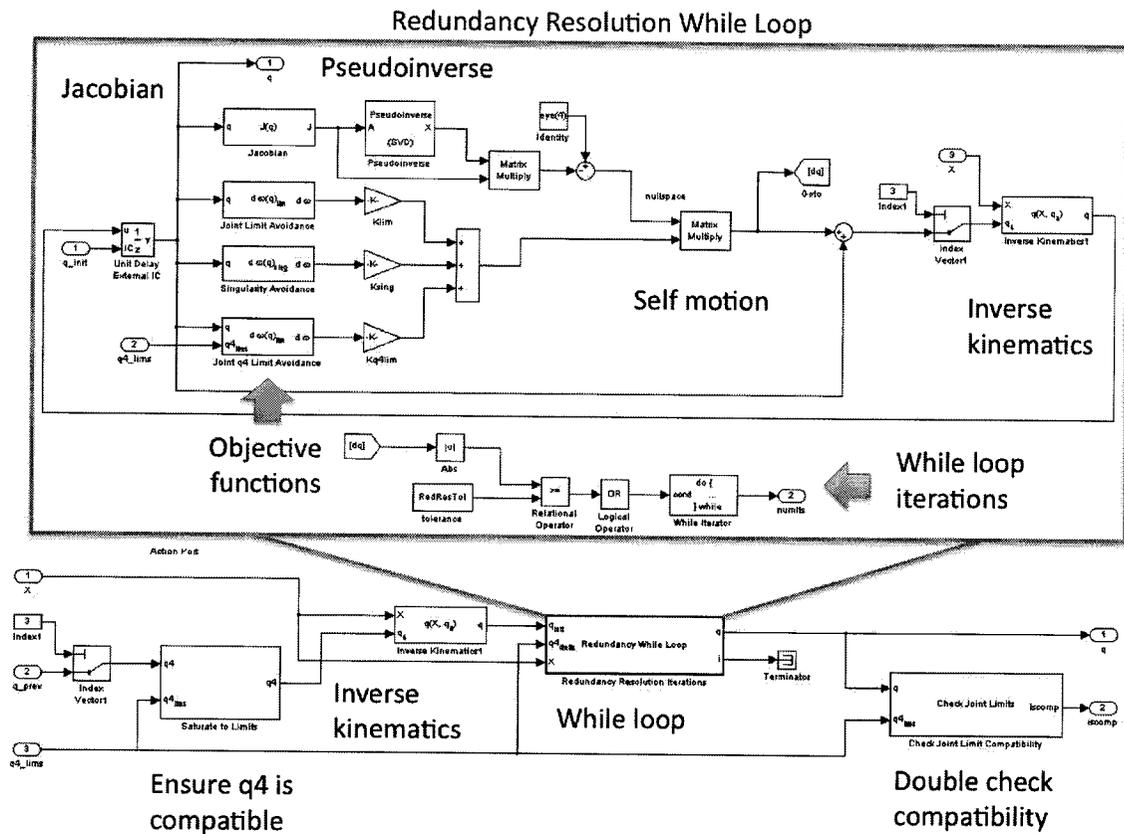
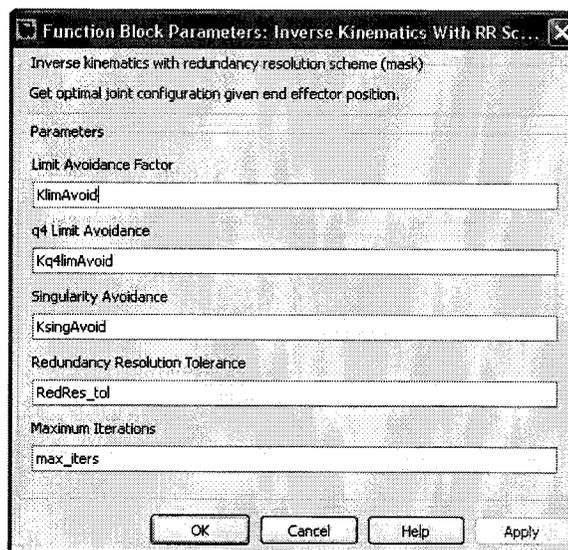


Figure 4.5: Redundancy resolution scheme Simulink subsystem.

1. The initial joint configuration is used to calculate the objective function  $\omega(q)$  partial derivatives with respect to the joint angles.
2. The objective function derivatives are scaled and summed to produce the joint angle gradient vector  $\delta q^0$  using (2.43).
3. The vector  $\delta q^0$  is applied as a self motion in the null space of the robot from the Jacobian and pseudoinverse calculation  $\Delta q = (I - J^\dagger J_r)\delta q^0$ .
4. Convergence criteria is tested by checking if all elements of  $|\Delta q|$  are less than the tolerance  $\epsilon^0$ .
5. If  $\max |\Delta q| < \epsilon^0$ , or the maximum number of iterations are reached, set output joint configuration to the current iteration  $q = q_i$  and break the loop. Otherwise, update the joint configuration for the next iteration  $q_{i+1} = q_i + \Delta q$ .



**Figure 4.6:** Inverse kinematics with redundancy resolution scheme subsystem mask.

A mask of the redundancy resolution was created to simplify modifications to the parameters. Figure 4.6 shows a screenshot of the mask with fields for editing the

objective function gains, convergence tolerance and maximum number of iterations. The parameters used for the ViGRR controller are shown in Table 4.3.

**Table 4.3:** ViGRR redundancy resolution scheme parameters.

Parameter	Value
$k_{sing}^0$	1
$k_{lim}^0$	0.5
$k_{comp}^0$	0.5
$\epsilon^0$	0.00001
Maximum iterations	130

#### 4.4.6 Controller Graphical User Interface

The ViGRR controller graphical user interface shown in Figure 4.7 was created using Matlab's GUIDE. The GUI is organized with panels corresponding to different controllers described in 4.4.3 and communicates with the target PC during execution of the controller. A control flow is built into the GUI where only the appropriate user controls are enabled. This prevents unwanted behaviour such as switching to a Cartesian space controller without performing an initial movement to the optimal joint configuration, or resetting a joint angle while it is used in a feedback loop. The GUI also allows for quick and easy changes to gains, trajectory parameters, control modes, initialization functions and data logging.

A second GUI was created for visualizing data. In Figure 4.8, the variables to be visualized are provided in a pull-down menu. The options for visualization include: End effector positions, joint angles, end effector velocities, joint angular velocities, command joint torques, and end effector forces and torques. For joint angles or end effector positions, both the desired and measured values are plotted continuously over time. Finally, a square x-y plot is shown for the cyclic trajectory following control schemes with desired and measured end effector trajectories. A second x-y

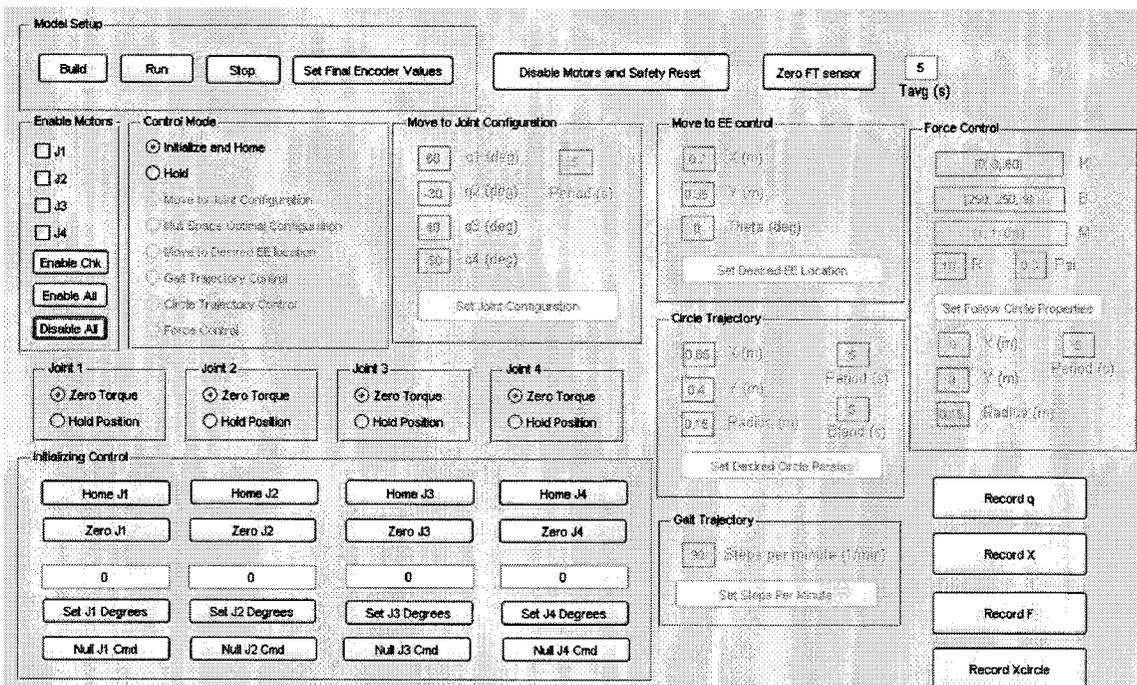


Figure 4.7: ViGRR graphical user interface for low-level control.

plot, exactly the same as the one in this GUI, is shown on a full screen for the user manipulating the robot in force control.

#### 4.4.7 ViGRR Robot Control Simulink Library

Many of the ViGRR functions require long, complex calculations that would be difficult or impossible to implement in a visual programming environment like Simulink. Therefore, many components of the controller code are written in C programming language and implemented as user-defined S-functions in Simulink. The wrapped C code is integrated seamlessly into S-function blocks that are used like any other blocks in Simulink. Some examples of C code and S-function implementation are: all of the smooth trajectory generators, the robot and leg dynamic matrices, Jacobians, robot dynamics regressor, redundancy objective functions, some safety logic code, etc. Many subsystems in the ViGRR controller are complex and it is sometimes

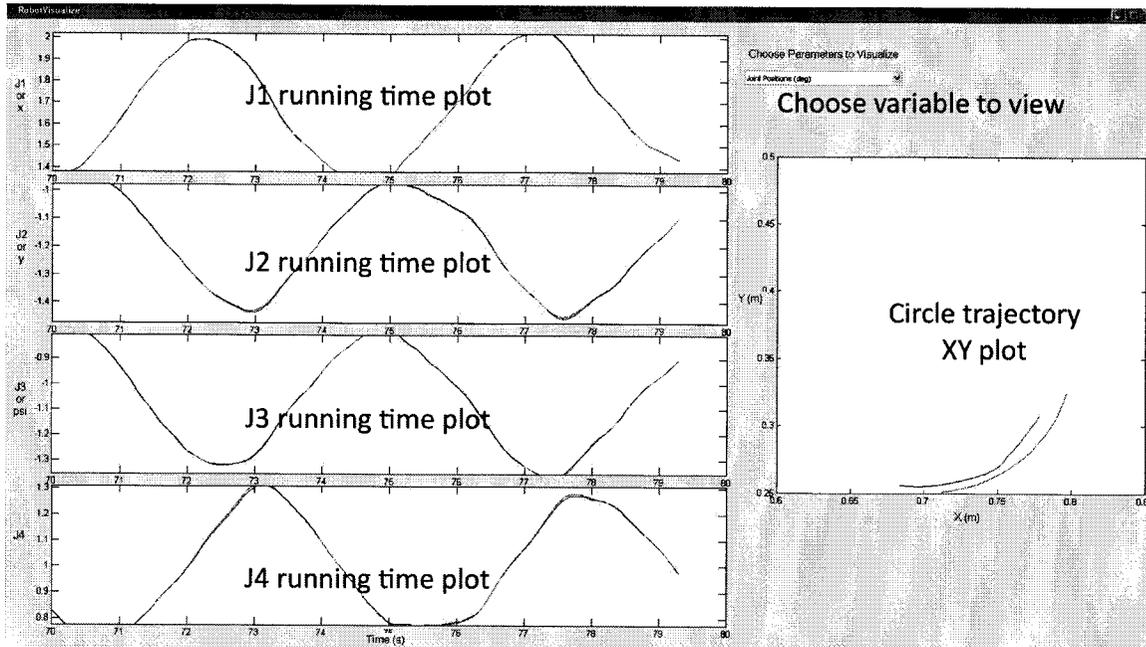


Figure 4.8: ViGRR visualization graphical user interface.

difficult to find appropriate gains or parameters to change. Also, many blocks and functions are used repeatedly, and it can become difficult to ensure all instances of the subsystem are updated to the current version. Using masks like the one described for the redundancy resolution scheme subsystem (Figure 4.6) and creating a Simulink library helps mitigate these issues in code maintenance and version control.

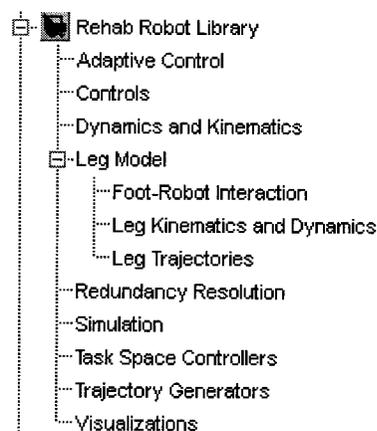


Figure 4.9: ViGRR simulink library hierarchy.

With a large number of subsystems and increasing complexity in the ViGRR controller, a Simulink library was created for the controller. A library provides a central location for the source and compiled C code, as well as singular subsystems to maintain which can be linked to any number of models or instances. For ViGRR, the library hierarchy is shown in Figure 4.9. Each library section contains relevant blocks for use in the ViGRR controller.

## 4.5 Simulation

In addition to the implemented controller for the robot, a detailed simulation was created in Simulink to test newly implemented controllers before applying them to the actual robot. Using a simulation reduces risk in damaging the robot and allows for quick testing and development of new controllers. Both the leg and robot were simulated using the derived equations of motion for each model. The robot dynamic simulation is shown in Figure 4.10, where the accelerations are calculated in the forward dynamics of the ViGRR model and integrated twice to get joint velocities and positions. The Simulink ode45 Dormand-Prince variable-step solver was used to perform the integration.

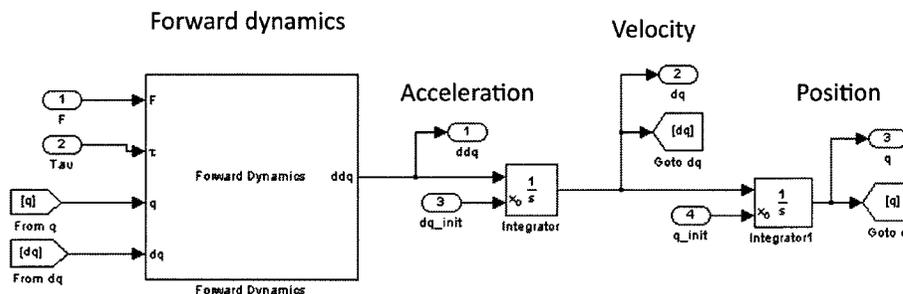


Figure 4.10: ViGRR dynamics simulation.

The ViGRR controller was implemented in simulation as an atomic subsystem with a fixed discrete step size of 0.001 seconds. The encoder and FT sensor inputs

were modelled according to the physical system parameters of the ViGRR apparatus, as described in the following section.

### 4.5.1 Sensor Models

Given an actual applied force at the robot end effector in the simulation, the FT sensor model generates a voltage from the DAQ for the simulated ViGRR controller. The FT sensor noise was modelled simply as Gaussian noise with a specified bias. Quantization is based on the published DAQ analog input resolution on the specification sheet in Appendix A. The standard deviation and bias for the Gaussian noise is based on experimentally acquiring raw voltages over 5 seconds from the Mini85. The simulated voltage input noise standard deviation and bias are set as  $\sigma_t = [0.01246, 0.01224, 0.01149, 0.01188, 0.01196, 0.01203]^T$  and  $b_t = [-0.06797, 0.0007387, 0.1368, 0.09660, -0.07943, 0.2331]^T$ , respectively.

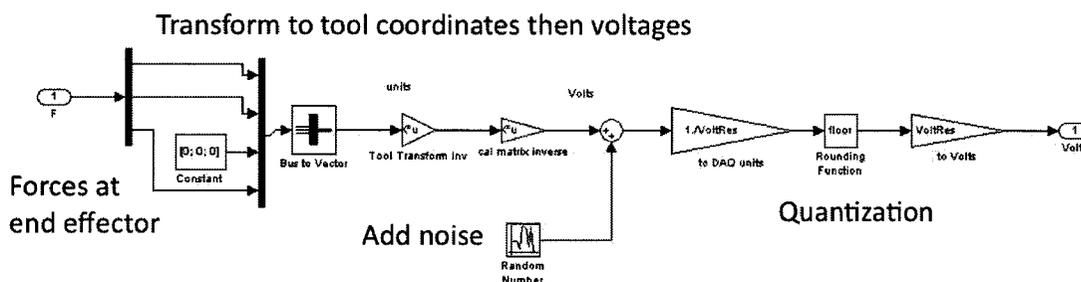
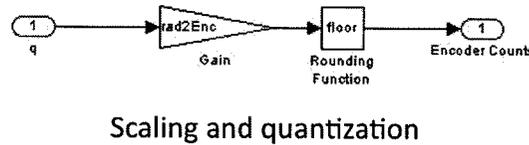


Figure 4.11: Force/torque sensor simulation.

The process for simulating the FT sensor is shown in Figure 4.11. In this subsystem, the forces at the end effector are transformed to the FT sensor tool coordinate frame. The inverse of the FT calibration matrix is applied to transform the units to voltages. Finally, noise is added to the voltages and the signal is rounded to the nearest DAQ voltage resolution (quantization).

The encoder is simulated very simply as the joint position scaled according to the

gear ratio and encoder counts per revolution, then rounded to the nearest encoder count, as shown in Figure 4.12. The encoder counts per revolution and gear ratios are provided in Section 3.4.1.



**Figure 4.12:** Encoder sensors simulation.

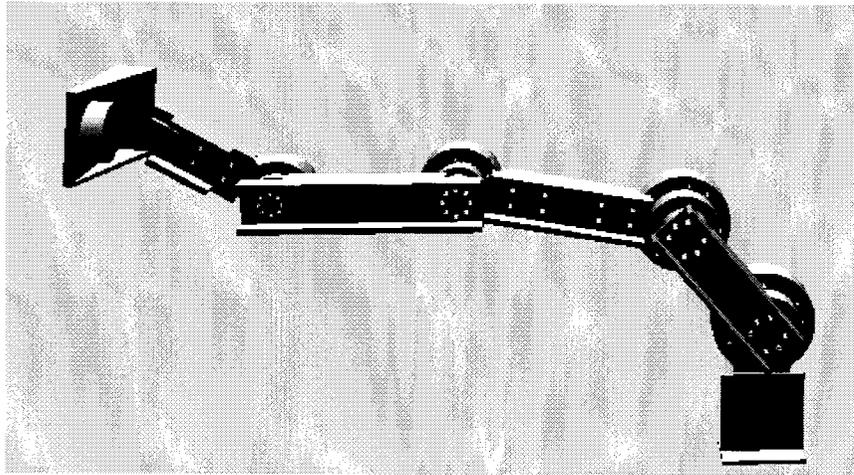
The output of the ViGRR controller is a command torque input to the S200 base units. The dynamics of the drive units and electrical components of the DC motors are not simulated and it is assumed that the command torque is equal the torque instantaneously supplied by the motors. This assumption is made since the S200 drive units have an input torque and operate the motor at a very high frequency with almost instantaneous torque generation.

### 4.5.2 Visualization

A 3D visualization for the simulation is implemented using the QuaRC visualization tools and a CAD model of ViGRR. The joint angles of the robot are animated in real-time using inputs from a Simulink signal. Figure 4.13 shows the ViGRR 3D model used in the animation. The visualization tools can be further utilized with a human model and visual feedback for the user.

## 4.6 Calibration

The calibration of ViGRR manipulator is essential to determine the kinematic and dynamic parameters of the robot. The approximate robot model parameters from



**Figure 4.13:** Visualization of ViGRR X3D model.

the CAD drawings and apparatus data sheets may not be accurate. Identifying these parameters as accurately as possible is necessary for the stability and precision of the model-based control schemes implemented in the ViGRR controller. The kinematic and dynamic calibrations presented in this section serve to determine these parameters.

#### 4.6.1 Basic Kinematic Calibration

The robot is redundant and a range of joint angles are permissible for a given end effector position and orientation. The kinematic calibration exploits this “self-motion” property to simplify the process. The end effector was fastened to the base of the robot stage at a known location and orientation. A self motion of the robot was performed by manually moving the robot to a number of different joint configurations. The linkage lengths were then calculated by a least squares fit of the data. The kinematic model used for the least squares fit is the forward kinematic model of the robot (2.15) rearranged to isolate the link lengths in

$$X_e = Q(q)L_r = \begin{bmatrix} c_1 & c_{12} & c_{123} & c_{1234} & 0 \\ s_1 & s_{12} & s_{123} & s_{1234} & 0 \\ 0 & 0 & 0 & 0 & q_1 + q_2 + q_3 + q_4 \end{bmatrix} L_r, \quad (4.27)$$

where  $L_r = [l_1, l_2, l_3, l_4, 1]^T$  is an augmented vector of linkage lengths. From  $n$  selected joint configurations and knowing the end effector positions, the link lengths can be calculated by least squares from a  $3n$  by 5 regressor matrix of data points defined by

$$\begin{aligned} \mathbf{X}_e &= \mathbf{Q}l_r \\ \begin{Bmatrix} (X_e)_1 \\ (X_e)_2 \\ \vdots \\ (X_e)_n \end{Bmatrix} &= \begin{bmatrix} Q(q)_1 \\ Q(q)_2 \\ \vdots \\ Q(q)_n \end{bmatrix} l_r. \end{aligned} \quad (4.28)$$

The least squares solution to the kinematic calibration is

$$\hat{l}_r = (\mathbf{Q}^T \mathbf{Q})^{-1} (\mathbf{Q}^T \mathbf{X}_e). \quad (4.29)$$

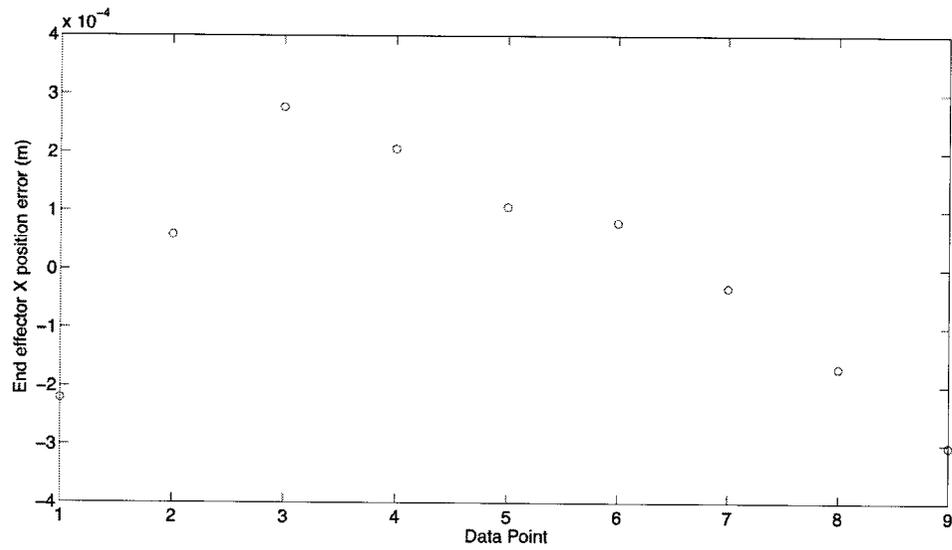
The recalculated end effector positions estimated using the calibrated linkage lengths and forward kinematic model are

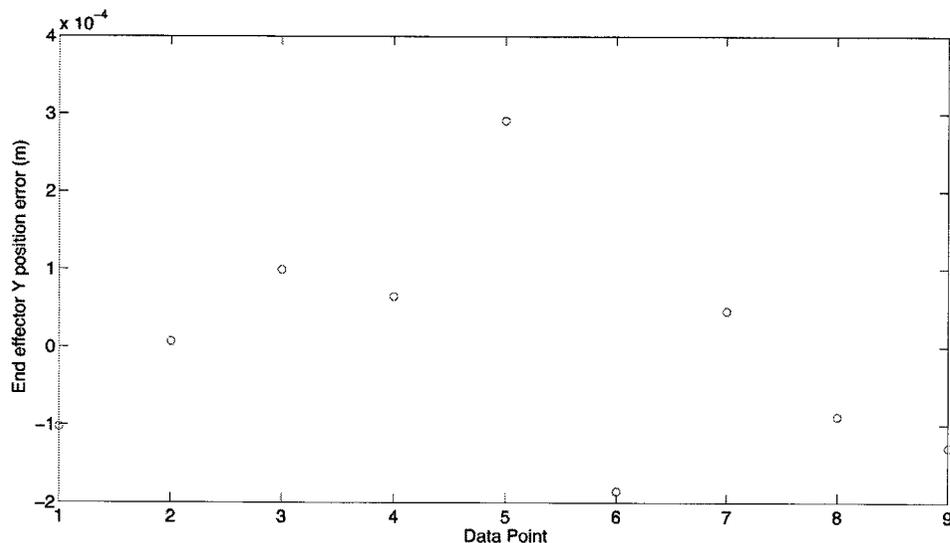
$$\hat{\mathbf{X}}_e = \mathbf{Q}\hat{l}_r \quad (4.30)$$

The kinematic calibration was performed for a single end effector location  $X_e = [-0.292208, -0.167589, 5.1051]$  and  $n = 9$  joint configurations. For each data point, the model errors  $\tilde{\mathbf{X}}_e$  were determined between the known  $X_e$  locations and the end effector positions calculated based on the forward kinematics and calibrated link lengths where  $\tilde{\mathbf{X}}_e = \hat{\mathbf{X}}_e - \mathbf{X}_e$ . Figures 4.14, 4.15, and 4.16 display the model forward kinematic errors. The final calibrated ViGRR linkage lengths are provided in Table 4.4.

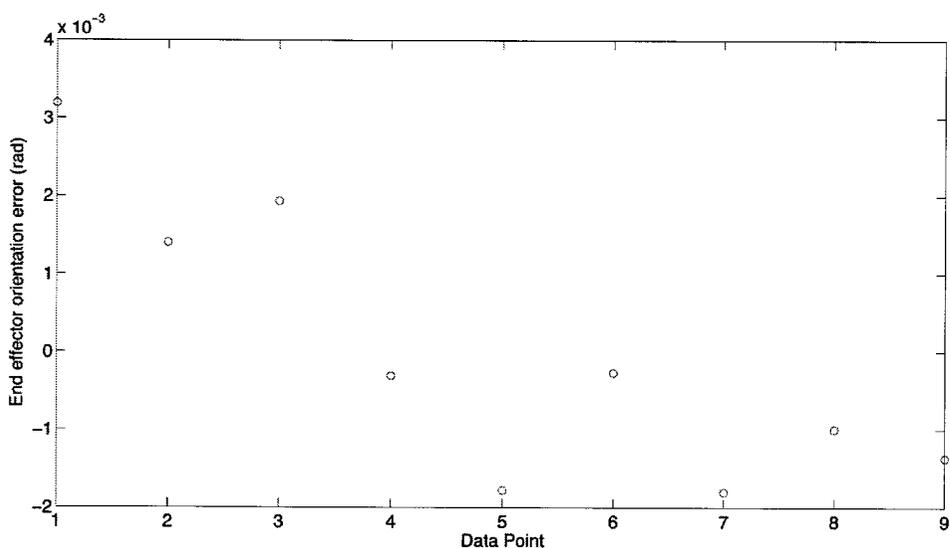
**Table 4.4:** Kinematic calibration results for link lengths.

Link	Length (m)
$l_1$	0.2065
$l_2$	0.3598
$l_3$	0.2823
$l_4$	0.3199

**Figure 4.14:** Kinematic calibration error for recalculated end effector X position.



**Figure 4.15:** Kinematic calibration error for recalculated end effector Y position.



**Figure 4.16:** Kinematic calibration error for recalculated end effector orientation.

The kinematic calibration errors are a measure of how well the least squares approximation worked and it is unknown how accurate the results are. The true value of  $X_e$  was based on CAD geometry and measurements on the surface of the robot stage using scales, callipers and a digital level. The certainty of these measurements

are not defined and thus the accuracy of the calibration results can still be improved. Also, only one end effector position and orientation was used. Multiple end effector locations, more joint configurations, and an absolute measurement tool such as a vision tracking system with known accuracy would improve the calibration.

The calibrated link lengths were used with the ViGRR controller and initial experiments in Chapter 5. These experiments show satisfactory results with the calibrated kinematic parameters.

### 4.6.2 Single Linkage Dynamic Parameter Identification

Before full assembly of the robot, a dynamic calibration was performed for each link to determine their dynamic properties of mass, centre of mass, inertia, and friction model coefficients. A sine wave trajectory was set for the single joint of the link in question, and the joint positions and applied torques were logged. The sine wave included varying frequencies, amplitudes and inertias with (4.4) in order to excite all the dynamic modes. Only the linkage being used for the calibration was actuated without the rest of the robot assembled or attached to the link. The test setup for link  $k$  involved holding link  $k - 1$  fixed to the base of the robot stage while the actuated joint was moved through the calibration trajectory. The applied torques and joint angles were logged for the single joint. Zero-phase filtering was then performed on the joint positions and the angular velocities and accelerations were determined by numerical differentiation, and a least squares regression was applied to obtain the dynamic parameters for each link. With the zero joint angle set as horizontal, the single link dynamic model used in the calibration process for joint  $k$  is

$$\tau_{r_k} = (I_k + I_{\text{eff}_k})\ddot{q}_k + B_{\text{eff}_k}\dot{q}_k + \nu_k \tanh(u_k \dot{q}_k) + m_k l_{Ck} g \sin(q_k). \quad (4.31)$$

In this model, the coefficients of each term can be isolated in the form

$$\tau_{r_k} = D(q_k, \dot{q}_k, \ddot{q}_k)\Theta = \begin{bmatrix} \ddot{q}_k & \dot{q}_k & \tanh(u_k \dot{q}_k) & \sin(q_k) \end{bmatrix} \Theta, \quad (4.32)$$

where

$$\Theta = \begin{Bmatrix} (I_k + I_{\text{eff}_k}) \\ B_{\text{eff}_k} \\ \nu_k \\ m_k l_{Ck} g \end{Bmatrix}.$$

From  $n$  data points over the calibration trajectory, the dynamic parameters are calculated from an  $n$  by 4 augmented matrix of data points defined by

$$\begin{aligned} \mathbf{T}_{r_k} &= \mathbf{D}\Theta \\ \begin{Bmatrix} (\tau_{r_k})_1 \\ (\tau_{r_k})_2 \\ \vdots \\ (\tau_{r_k})_n \end{Bmatrix} &= \begin{bmatrix} D(q_k, \dot{q}_k, \ddot{q}_k)_1 \\ D(q_k, \dot{q}_k, \ddot{q}_k)_2 \\ \vdots \\ D(q_k, \dot{q}_k, \ddot{q}_k)_n \end{bmatrix} \Theta. \end{aligned} \quad (4.33)$$

The least squares solution to the single DoF dynamic calibration is

$$\hat{\Theta} = (\mathbf{D}^T \mathbf{D})^{-1} (\mathbf{D}^T \mathbf{T}_{r_k}). \quad (4.34)$$

The recalculated joint torques estimated using the calibrated dynamic parameters and single link dynamic model are

$$\hat{\mathbf{T}}_{r_k} = \mathbf{D}\hat{\Theta}. \quad (4.35)$$

Dynamic properties in vector  $\hat{\Theta}$  for each link are solved by a least squares fit, but only if the coulomb friction velocity scaling term  $u_k$  is assumed. However, the term  $u_k$  is unknown and is a nonlinear term in the model. Therefore, the parameter  $u_k$  was determined by a brute force optimization as a search for the lowest sum of the squared errors of the calibration over a range of coefficients  $u_k$ . The search space of  $u_k$  for each joint was between 0.1 and 20 with a resolution of 0.01.

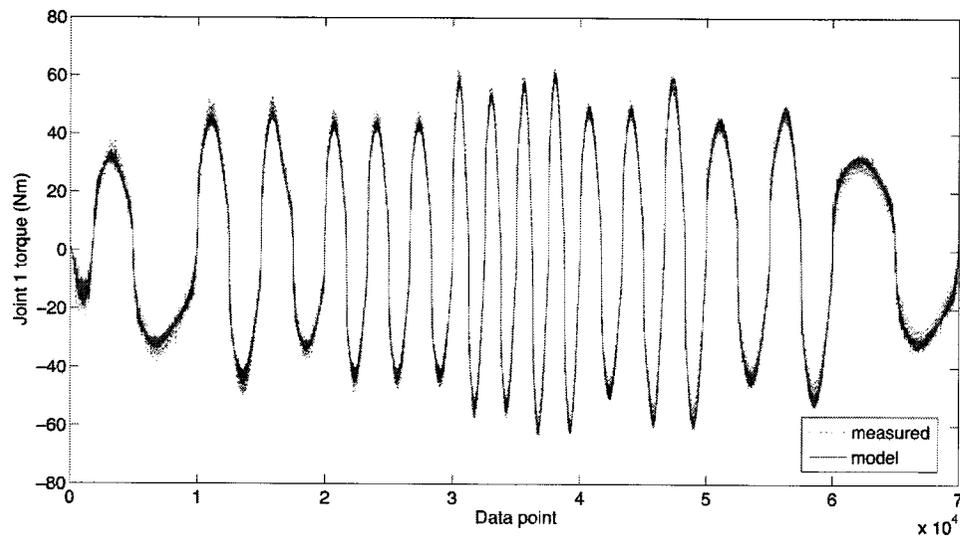
The model errors  $\tilde{\mathbf{T}}_{r_k}$  were determined between the known applied torques  $\mathbf{T}_{r_k}$  and the torques calculated based on the dynamic model and estimated parameters as

in

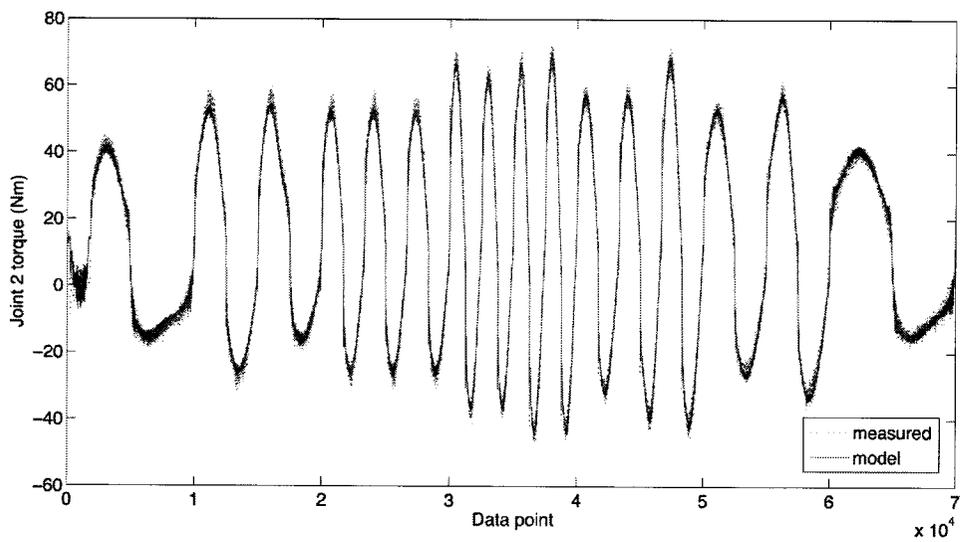
$$\tilde{\mathbf{T}}_{r_k} = \hat{\mathbf{T}}_{r_k} - \mathbf{T}_{r_k}. \quad (4.36)$$

Figures 4.21, 4.22, 4.23, and 4.24 display the calibrated dynamic model torque errors  $\tilde{\mathbf{T}}_{r_k}$  for links 1 to 4, respectively. Figures 4.17, 4.18, 4.19, and 4.20 display the model torques  $\hat{\mathbf{T}}_{r_k}$  and actual torques  $\mathbf{T}_{r_k}$  for links  $k = 1, \dots, 4$ , respectively.

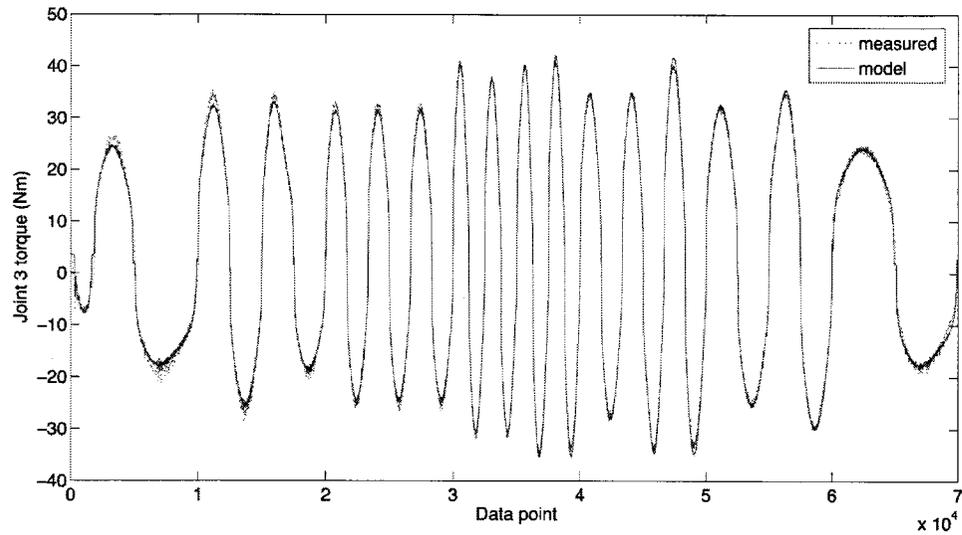
Finally, the individual dynamic parameters are calculated from the vector  $\hat{\Theta}$ . The mass  $m_k$  and centre of mass  $l_{Ck}$  are coupled in the fourth element of  $\hat{\Theta}$ . In order to solve for  $l_{Ck} = \hat{\Theta}_4 / (m_k g)$ , a digital scale was used to separately determine the mass of the link  $m_k$  with the acceleration due to gravity  $g = 9.81 \text{ m/s}^2$ . The link inertia  $I_k$  is estimated from the CAD model and  $I_{\text{eff}_k}$  is determined by  $I_{\text{eff}_k} = \hat{\Theta}_1 - I_k$ . The final results for all four links are presented in Table 4.5. These results yield the dynamic parameter  $\phi_s$  calculated from (4.6):



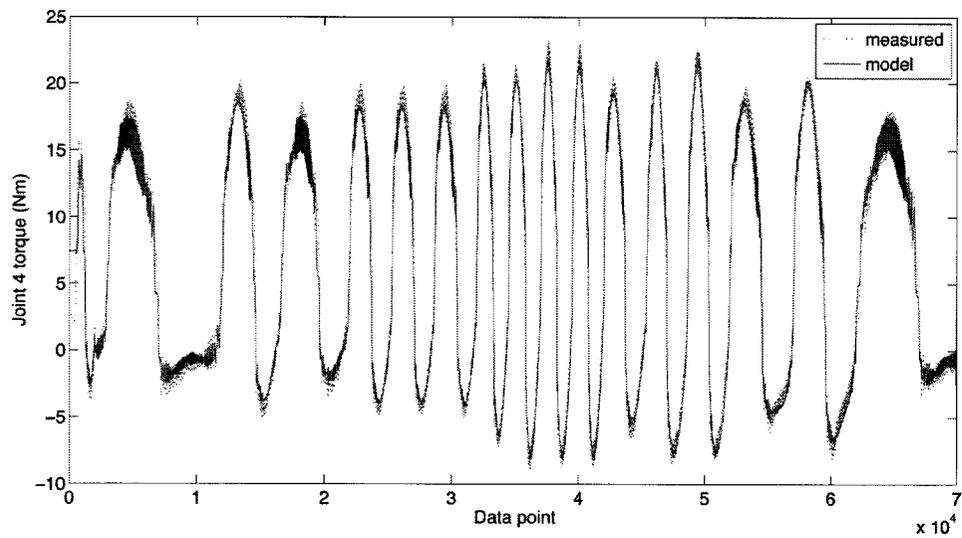
**Figure 4.17:** The command joint 1 torques (measured) and the computed torques (model) from the single DoF dynamic calibration.



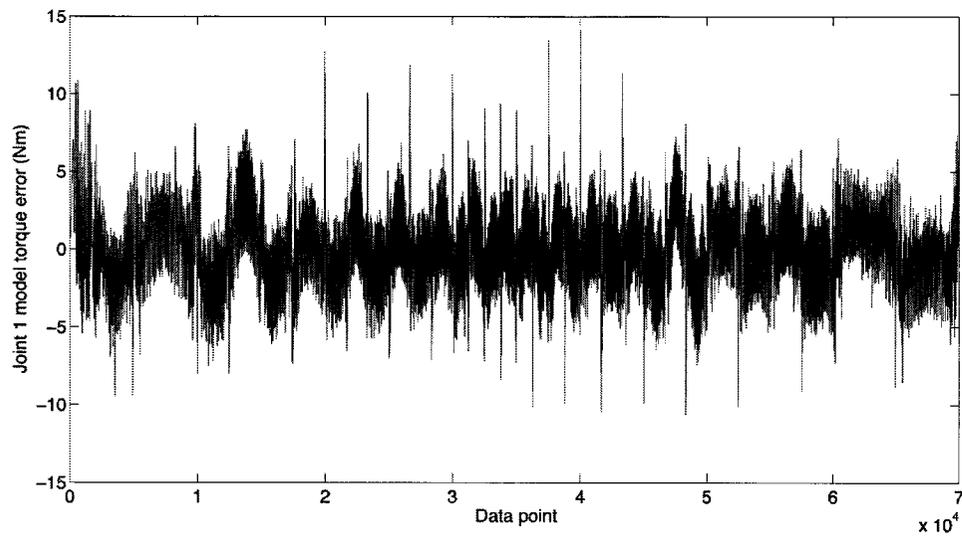
**Figure 4.18:** The command joint 2 torques (measured) and the computed torques (model) from the single DoF dynamic calibration.



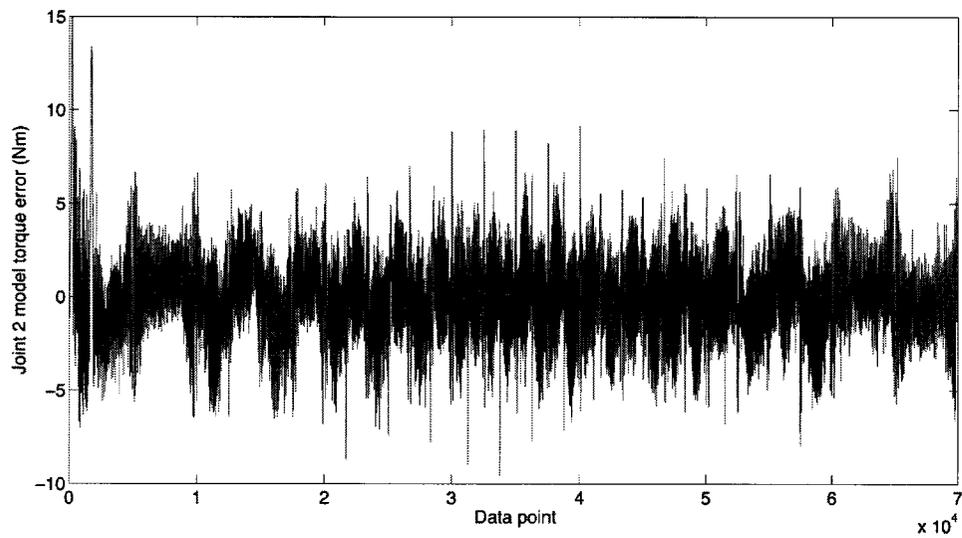
**Figure 4.19:** The command joint 3 torques (measured) and the computed torques (model) from the single DoF dynamic calibration.



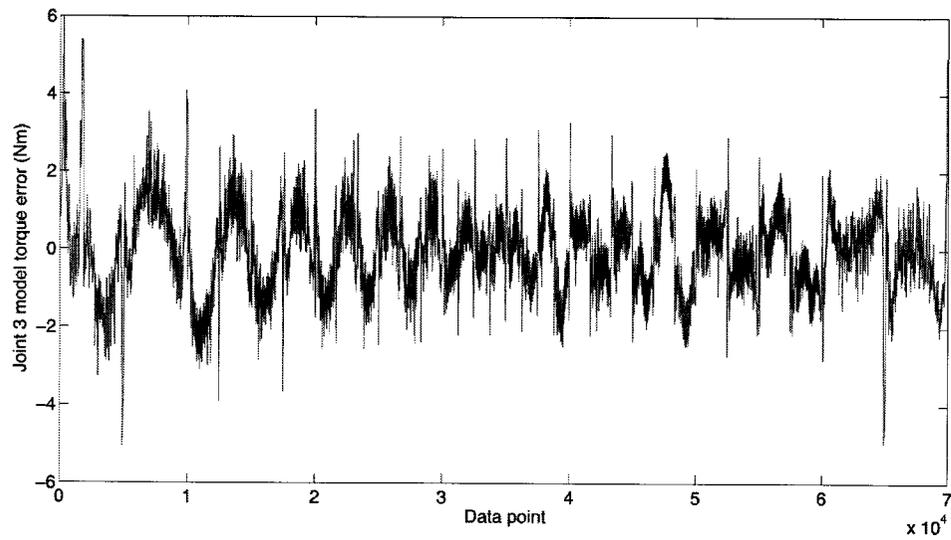
**Figure 4.20:** The command joint 4 torques (measured) and the computed torques (model) from the single DoF dynamic calibration.



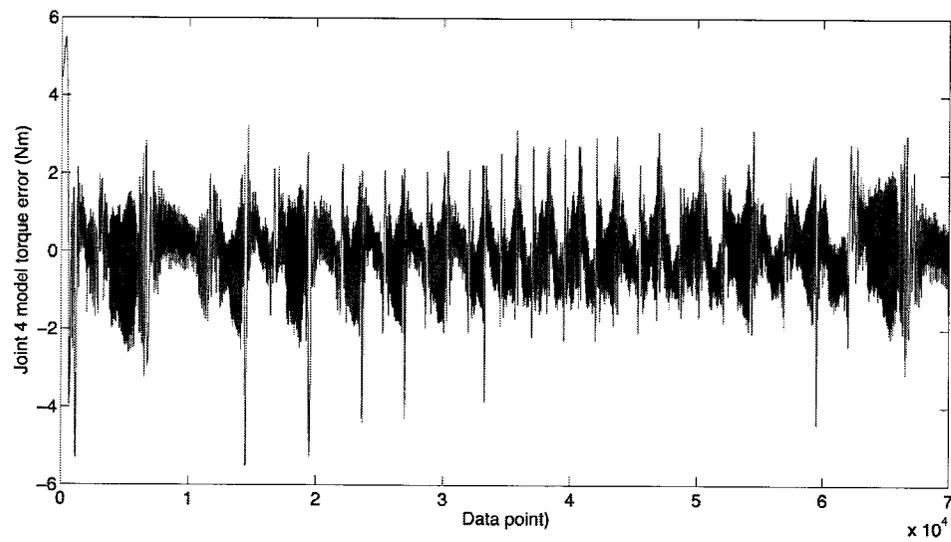
**Figure 4.21:** Dynamic calibration error for recalculated torques from Joint 1 single DoF calibrated model parameters.



**Figure 4.22:** Dynamic calibration error for recalculated torques from Joint 2 single DoF calibrated model parameters.



**Figure 4.23:** Dynamic calibration error for recalculated torques from Joint 3 single DoF calibrated model parameters.



**Figure 4.24:** Dynamic calibration error for recalculated torques from Joint 4 single DoF calibrated model parameters.

**Table 4.5:** Single DoF dynamic calibration results.

Link	Length (m)
$l_{C1}$	0.2633
$l_{C2}$	0.1051
$l_{C3}$	0.1961
$l_{C4}$	0.1325
$m_1$	1.1662
$m_2$	13.5075
$m_3$	1.8043
$m_4$	5.7489
$I_1$	0.007879
$I_2$	0.3992
$I_3$	0.0.02094
$I_4$	0.07775
$I_{\text{eff}1}$	4.2196
$I_{\text{eff}2}$	4.5830
$I_{\text{eff}3}$	1.5672
$I_{\text{eff}4}$	0.2675
$B_{\text{eff}1}$	27.2051
$B_{\text{eff}2}$	24.8834
$B_{\text{eff}3}$	18.0541
$B_{\text{eff}4}$	6.1571
$\nu_1$	15.6736
$\nu_2$	12.8680
$\nu_3$	10.6000
$\nu_4$	4.9839
$u_1$	11.70
$u_2$	17.11
$u_3$	11.72
$u_4$	12.23

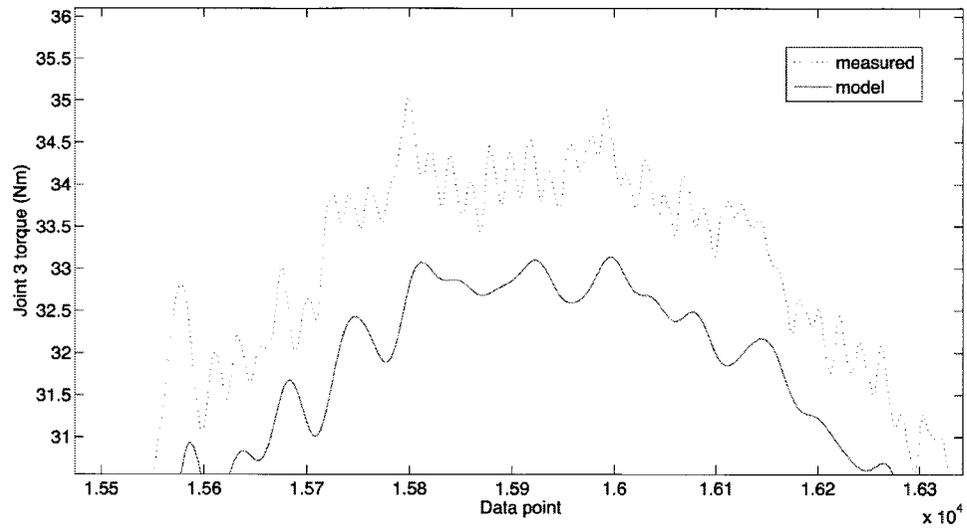
$$\hat{\phi}_s = \begin{pmatrix} 5.1553 \\ 0.5248 \\ 0.1786 \\ 1.5782 \\ 7.4668 \\ 0.7142 \\ 41.2672 \\ 0.2093 \\ 44.4264 \\ 18.9665 \\ 0.1527 \\ 0.2812 \\ 0.8439 \\ 0.3879 \\ 27.2051 \\ 24.8834 \\ 18.0541 \\ 6.1571 \\ 15.6736 \\ 12.8680 \\ 10.5999 \\ 4.9839 \\ 4.5830 \\ 1.5672 \\ 0.2675 \end{pmatrix}. \quad (4.37)$$

When comparing the model torques to the command torques in Figures 4.17 to 4.20, the model torques are similar to the applied torques with discrepancies most visible at the peaks and troughs where the model torques appear to have calculated lower magnitudes than the measured torques. The torque signals also appear to be noisy, especially for the 4th joint in Figure 4.20. Closer inspection shows that there are small regular oscillations in the torque as shown in Figure 4.26 that do not appear to be random or noise driven. In comparison, the torques in the 3rd joint (Figure 4.25) do not show these regular oscillations but do calculate lower torque values. Since the oscillations appear in the commanded torque values, the calibration process or dynamic model are not causing this phenomena. In fact, the model appears to match

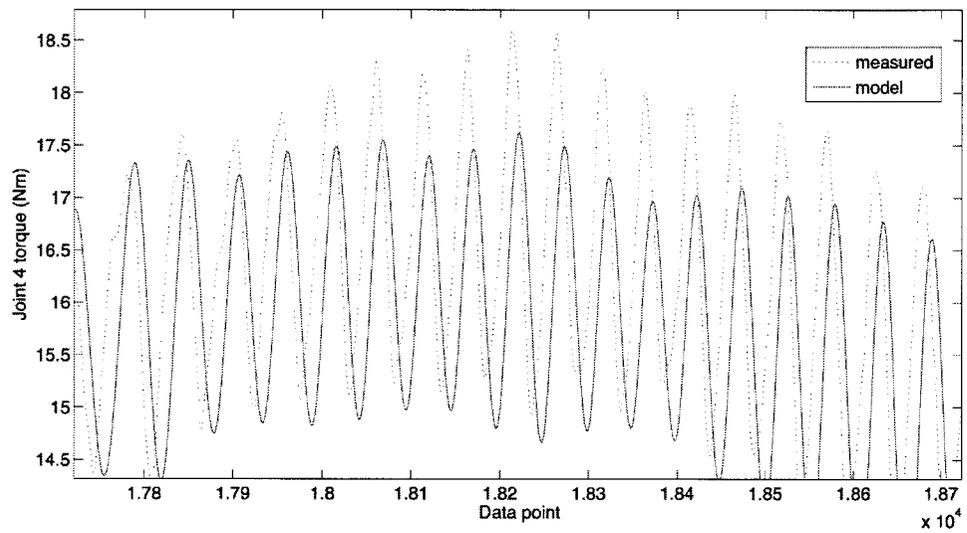
these small high frequency oscillations. The 4th joint oscillations may be due to mechanical issues with the joint related to motor shaft and harmonic drive alignment or perhaps delay issues with the velocity filter used in the PD feedback controller with derivative gains that are too high. The cause may also be due to high gains with unmodelled dynamics in the harmonic drive such as joint stiffness. This issue did not appear to be significant in future controller implementations of the entire assembled robot, and the model parameters were still deemed acceptable for model-based control with lower feedback gains until a second full robot dynamic calibration was performed. This full robot dynamic parameter calibration procedure is detailed in Section 4.6.3.

The torque errors in Figures 4.21 to 4.24 show fairly consistent errors with higher values for the first and second joints. These joints have a significantly larger range of capable torques than joint 3 and 4 and larger harmonic drive units and gear ratios that may be more sensitive to simplifications made in the friction model in Section 2.3.4, especially at lower velocities. The lack of any major peaks in the errors for each of the joints over the range of frequencies in the trajectory shows that the joint model and calibrated parameters hold up well for this initial calibration procedure.

The estimation of  $I_{\text{eff}_k}$  from the CAD model involved creating a boundary at the interface between the harmonic drives and the next linkage. This boundary includes the entire harmonic drive as part of link  $k - 1$  for joint  $k$ , which is an inaccurate assumption, since the output shaft on the harmonic drive gear unit should be considered part of link  $k$ . The same issue arises for calculating the mass of link  $k$  with the same boundary using a scale. Although this does not affect the single linkage dynamics, the inertias and centre of mass locations in the complex dynamic model of the assembled robot will be coupled with different terms. Therefore, a dynamic calibration procedure with the assembled robot is presented in the next section using the parameter-regressor formulation of the robot dynamics.



**Figure 4.25:** Close up of joint 3 model and measured torques for the single link dynamic calibration.



**Figure 4.26:** Close up of joint 4 model and measured torques for the single link dynamic calibration.

### 4.6.3 Full Robot Dynamic Parameter Identification

The single link dynamic calibration provided good initial values for controlling the robot with the inverse dynamics inner loop. However, these dynamic properties may change once the robot is fully assembled. Another calibration is performed for the robot dynamics using the Regressor-Parameter formulation introduced in Section 4.2.1. The dynamic model of the robot is in the form

$$\tau = Y(q, \dot{q}, \ddot{q})\phi \quad (4.38)$$

which is already linearized with respect to the dynamic parameter  $\phi$ , making it a simple matter for calibration by least squares fit. The Coulomb friction velocity scaling terms  $u_k$  required in the calculation of the regressor  $Y(q, \dot{q}, \ddot{q})$  are set as constant values from the single link dynamic calibration process.

A smooth sine wave trajectory was set for all of the joints with varying bias, amplitude and frequency. The joint positions and applied torques were logged. Zero-phase filtering was performed on the joint positions and the angular velocities and accelerations were determined by numerical differentiation. From  $n$  data points over the calibration trajectory, the dynamic parameters are calculated from a  $4n$  by  $25$  augmented matrix from the data points defined by

$$\tau = \mathbf{Y}\phi$$

$$\begin{Bmatrix} (\tau_r)_1 \\ (\tau_r)_2 \\ \vdots \\ (\tau_r)_n \end{Bmatrix} = \begin{bmatrix} Y(q, \dot{q}, \ddot{q})_1 \\ Y(q, \dot{q}, \ddot{q})_2 \\ \vdots \\ Y(q, \dot{q}, \ddot{q})_n \end{bmatrix} \phi \quad (4.39)$$

The least squares solution to the full robot dynamic calibration is

$$\hat{\phi} = (\mathbf{Y}^T \mathbf{Y})^{-1} (\mathbf{Y}^T \tau) \quad (4.40)$$

The recalculated joint torques estimated using the calibrated dynamic parameter and regressor are

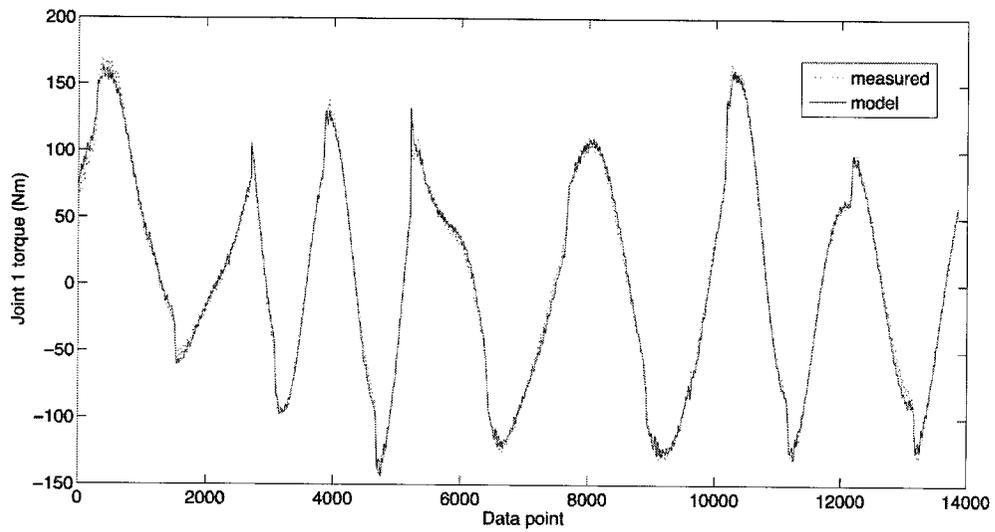
$$\hat{\tau} = \mathbf{Y}\hat{\phi}. \quad (4.41)$$

The dynamic calibration errors  $\tilde{\tau}$  were determined between the known applied torques  $\tau$  and the torques calculated based on the dynamic model and estimated parameters:

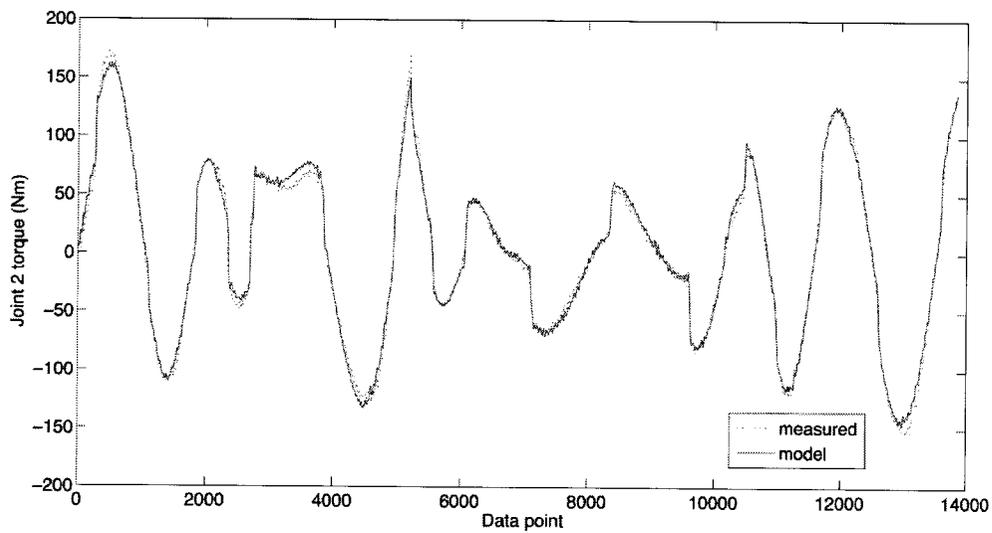
$$\tilde{\tau} = \hat{\tau} - \tau. \quad (4.42)$$

Figures 4.31, 4.32, 4.33, and 4.34 display the calibrated dynamic model torque errors  $\tilde{\tau}$  for joints  $k = 1, \dots, 4$ , respectively. Figures 4.27, 4.28, 4.29, and 4.30 display the calibrated model torques  $\hat{\tau}$  and actual torques  $\tau$  for links  $k = 1, \dots, 4$ , respectively.

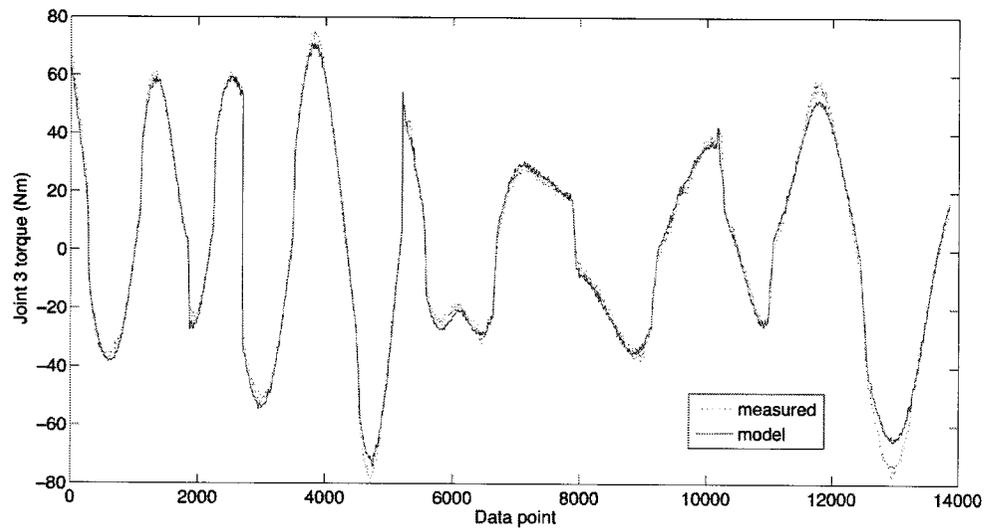
Figures 4.27 to 4.30 show that the model torques are very similar to the applied torques. The noisy signal does not appear to be significant in this calibration procedure compared to the single joint calibration. The applied torques also follow a trajectory that is not as smooth or periodic as the single joint parameters due to the complex dynamics involved with actuating all the joints of the assembled robot simultaneously. The torque errors in Figures 4.31 to 4.34 show consistent errors with higher values for the first and second joints, similar to the single joint calibration. The simplifications in the drive train models may have a greater effect on the first and second joints with larger harmonic drive units, gear ratios and torque range. The lack of any major peaks in the errors for each of the joints over the range of frequencies demonstrates that the joint model and calibrated parameters can be effective for the model based control of the robot.



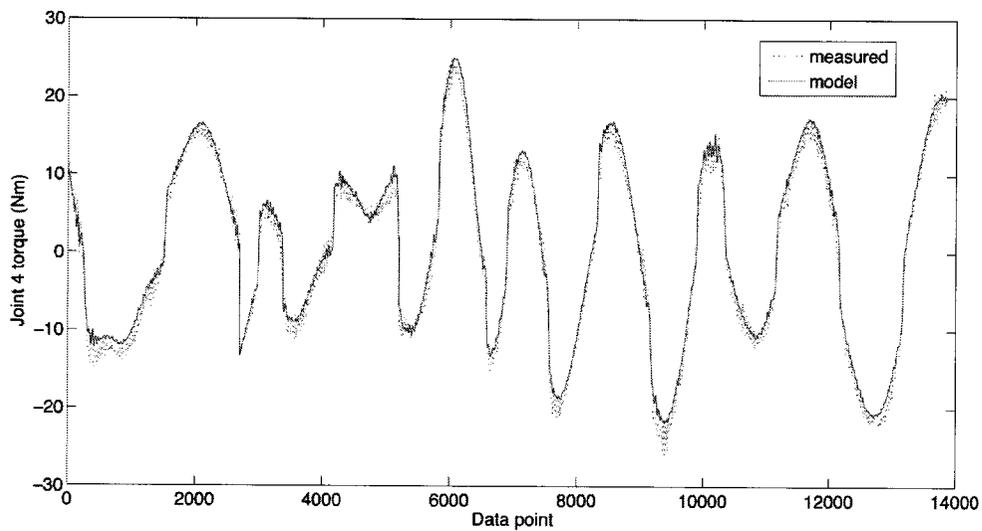
**Figure 4.27:** The command joint 1 torques (measured) and the computed torques (model) from the parameter dynamic calibration.



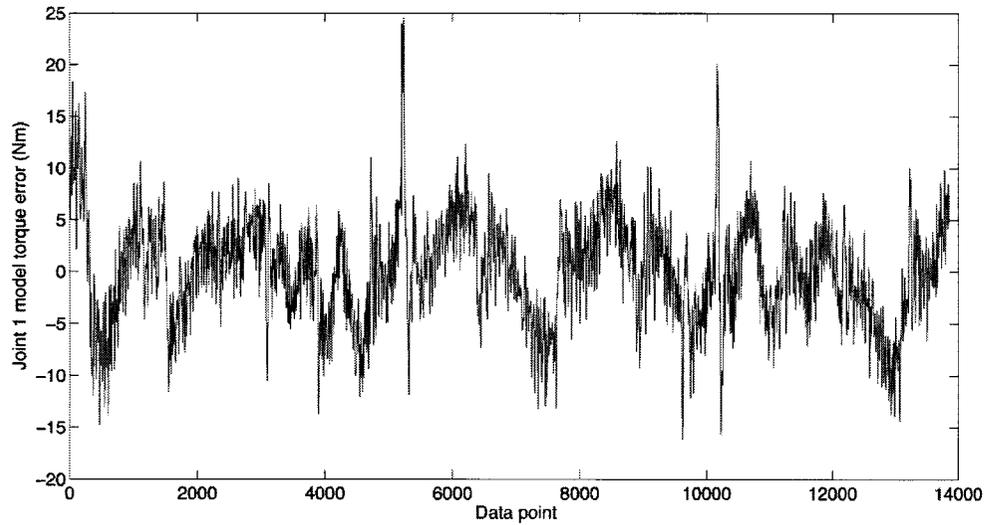
**Figure 4.28:** The command joint 2 torques (measured) and the computed torques (model) from the parameter dynamic calibration.



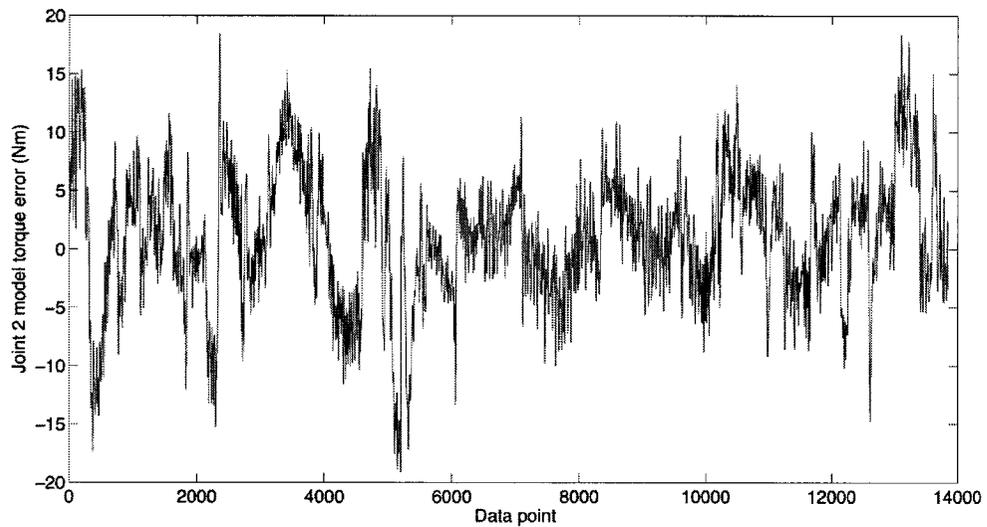
**Figure 4.29:** The command joint 3 torques (measured) and the computed torques (model) from the parameter dynamic calibration



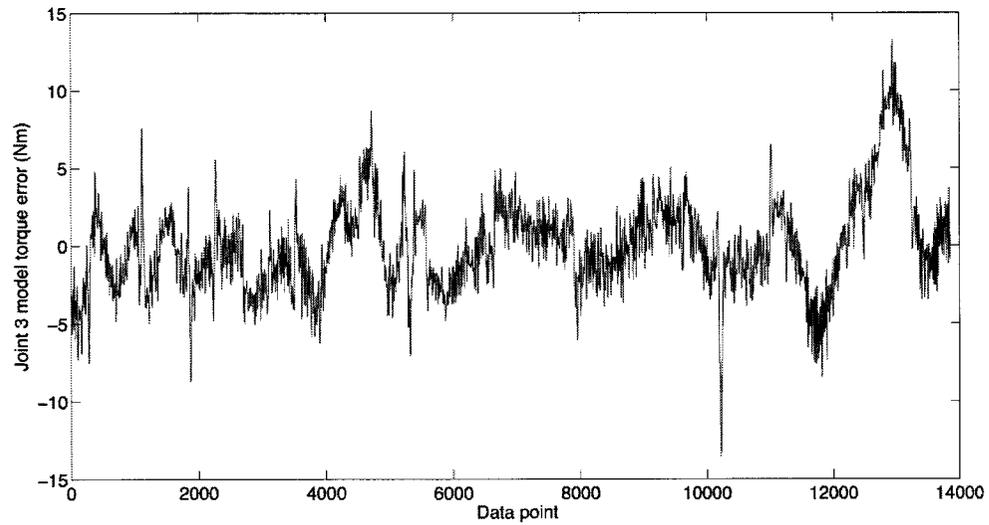
**Figure 4.30:** The command joint 4 torques (measured) and the computed torques (model) from the parameter dynamic calibration.



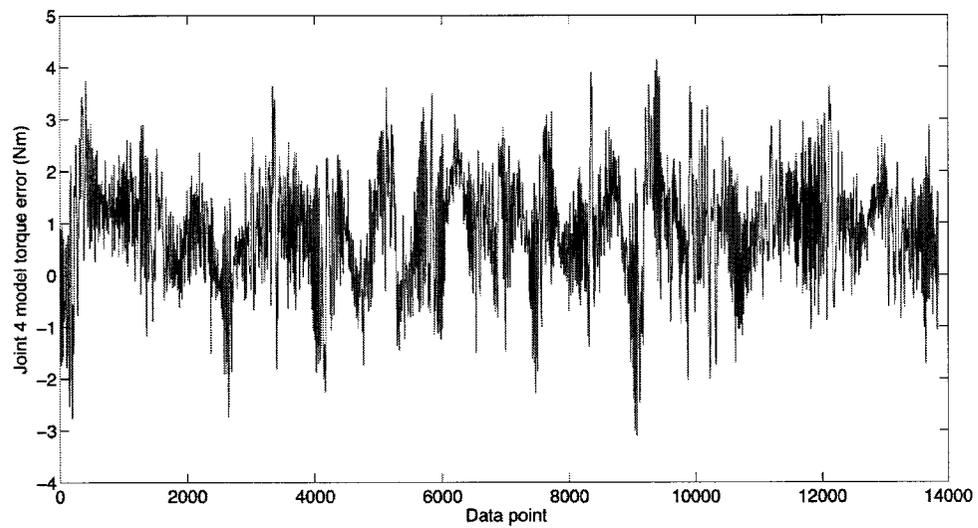
**Figure 4.31:** Dynamic calibration error for calculated joint 1 torques from 4DoF calibrated model parameters.



**Figure 4.32:** Dynamic calibration error for calculated joint 2 torques from 4DoF calibrated model parameters



**Figure 4.33:** Dynamic calibration error for calculated joint 3 torques from 4DoF calibrated model parameters.



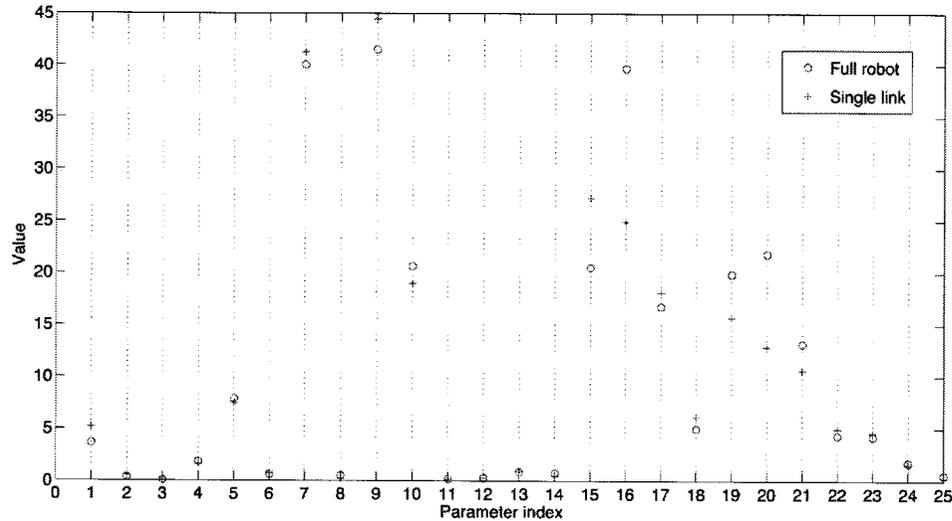
**Figure 4.34:** Dynamic calibration error for calculated joint 4 torques from 4DoF calibrated model parameters.

The final calibrated dynamic parameter for the full robot is

$$\hat{\phi} = \begin{pmatrix} 3.6275 \\ 0.3221 \\ 0.04393 \\ 1.8100 \\ 7.8466 \\ 0.5864 \\ 39.9933 \\ 0.4910 \\ 41.4924 \\ 20.6041 \\ 0.1637 \\ 0.2693 \\ 0.8439 \\ 0.7446 \\ 20.4829 \\ 39.6782 \\ 16.7666 \\ 5.03059 \\ 19.8307 \\ 21.7988 \\ 13.1578 \\ 4.3201 \\ 4.2331 \\ 1.7297 \\ 0.5267 \end{pmatrix}. \quad (4.43)$$

As a comparison to the single joint dynamic calibration results, the elements of the dynamic parameter from the full robot calibration  $\hat{\phi}$  is plotted with the resulting parameter from the single link calibration  $\phi_s$  in Figure 4.35. As seen from this plot, the dynamic parameter elements have similar values, with the largest absolute differences in the parameters being elements 15, 16, 19 and 20. These correspond to the first and second joint viscous friction and Coulomb friction coefficients. Its interesting that these errors are large considering the friction model is not coupled with other dynamic terms in the parameter  $\phi$ . However, there are significant axial loads and out of plane torques that may affect the friction in the joint when the robot is fully assembled.

The individual joint calibration was also performed shortly after the initial assembly of the new drive components and the properties of the harmonic drive units may have changed over the initial several hours of operation.



**Figure 4.35:** The dynamic parameter  $\phi$  from the single link dynamic calibration  $\phi_s$  and the full robot calibration  $\hat{\phi}$ .

## 4.7 Summary

This Chapter presented the ViGRR control theory, software implementation, simulation, and calibration. Inverse dynamics and passivity-based nonlinear control were applied with the parameter-regressor formulation of the dynamics. The implemented software features a graphical user interface, 3D visualizations, a flexible development platform (QuaRC and Simulink), a function library, and robust safety mechanisms. Finally, the robot was calibrated, completing the realization of the robot as a functional motion control platform. The next chapter involves experiments with the robot for assessing ViGRR's performance and demonstrating its capabilities and potential for rehabilitation with human interaction.

## Chapter 5

# Experimentation

The design process involved investigating the requirements, kinematic and dynamics modelling, mechanical and electrical apparatus, controls, systems integration and calibration. The system is ready for testing and operation and this chapter outlines a number of experiments performed to assess the capabilities of ViGRR and whether it can be used for rehabilitation and human interaction purposes. The trajectory tracking in both joint and Cartesian space is assessed. A final human interaction experiment using force feedback with an admittance control scheme and a visual display was applied. This final experiment demonstrates the capability of ViGRR to deliver a challenging task for the user in the task space, and can be extended for use with a leg and rehabilitation.

Figure 5.1 provides a flow chart of the chosen controls approach in the following experiments. For the joint trajectory generators used in these tests, the low level inner and outer loop controllers are the PD control with a feedforward acceleration term and inverse dynamics as presented in Section 4.2.2. The regressor-parameter formulation of the dynamics is used with the calibrated parameter  $\hat{\phi}$  (4.43). The PD gains for each joint were tuned to a critically damped second order system with a natural frequency  $\omega = 40$  rad/s. This corresponded to gains of  $K_0 = 1600\mathbf{I}_4$  and  $K_1 = 80\mathbf{I}_4$ , where  $\mathbf{I}_4$  is a  $4 \times 4$  identity matrix. This control strategy is also used for

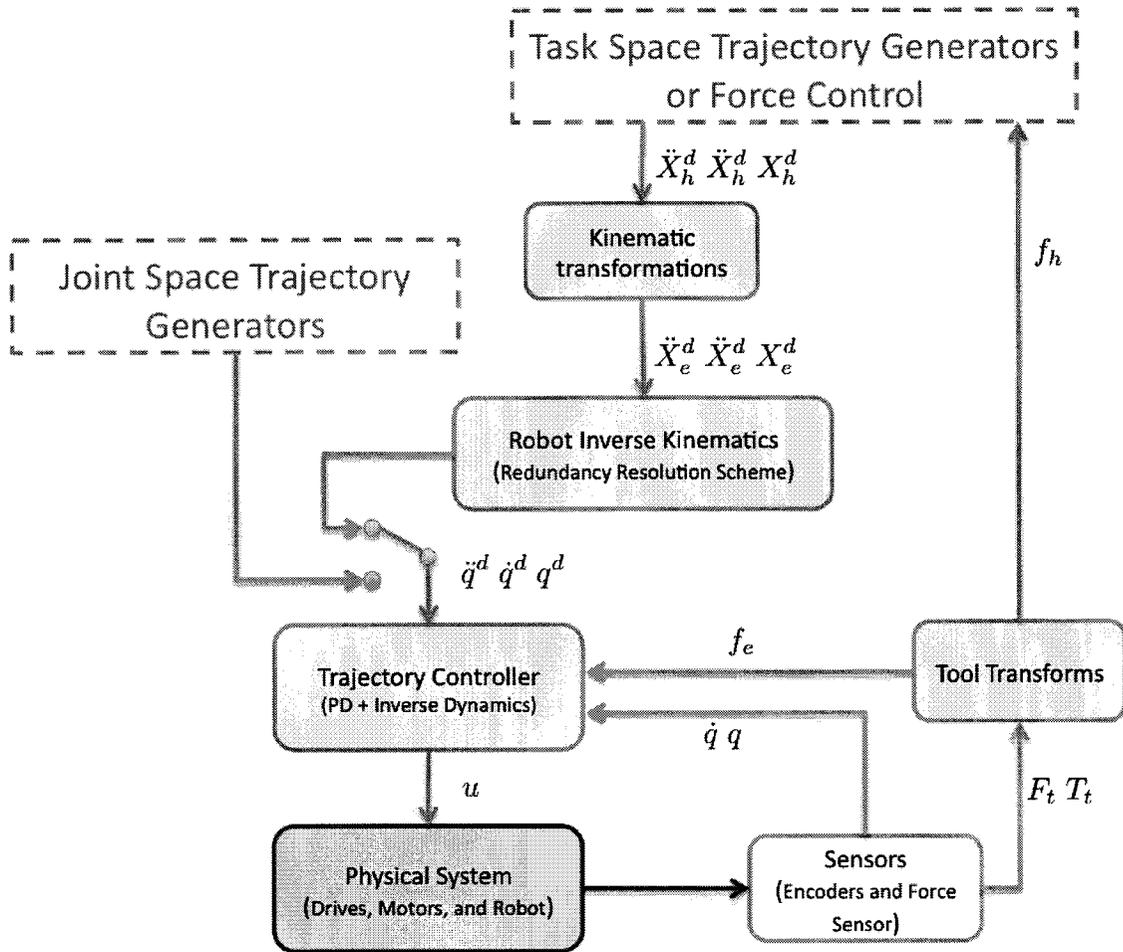


Figure 5.1: ViGRR Controls setup for experiment implementation.

the task space trajectory and admittance control, except the robot inverse kinematics are first used to transform the desired trajectories to joint space.

## 5.1 Joint Trajectory Tracking

Smooth step trajectories were applied to each of the joints simultaneously in order to assess the joint tracking performance of the controller. The 5th order trajectory (Section 4.1.1) was created between the start and end joint angles specified in Table 5.1. The time was set to 5 seconds.

**Table 5.1:** Joint step trajectory beginning and end values.

joint	Angle at time $t_0$ (deg)	Angle at time $t_f$ (deg)
$q_1$	20	110
$q_2$	80	-80
$q_3$	80	0
$q_3$	-60	50

### 5.1.1 Results

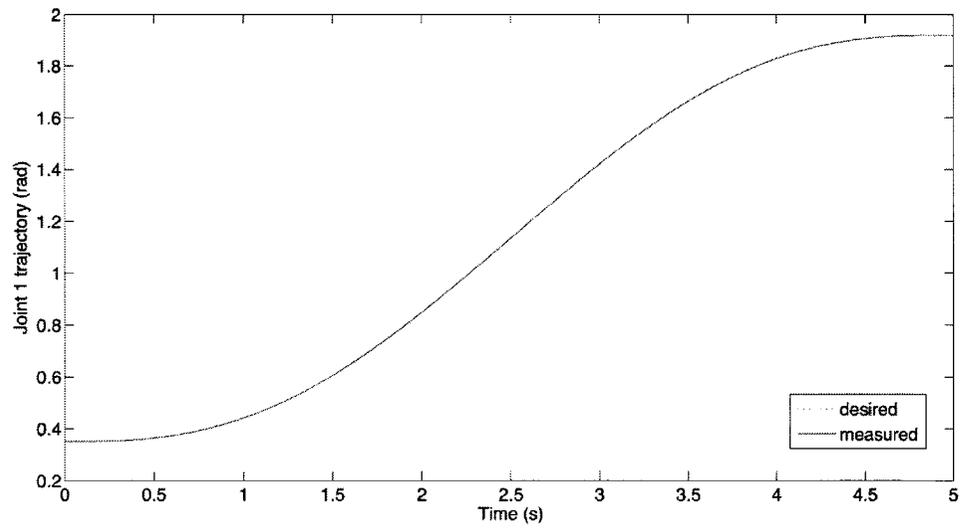
Figures 5.2 to 5.5 show the desired and measured joint trajectories. Figures 5.6 to 5.9 show the joint position errors over the trajectory.

### 5.1.2 Interpretation of the Results

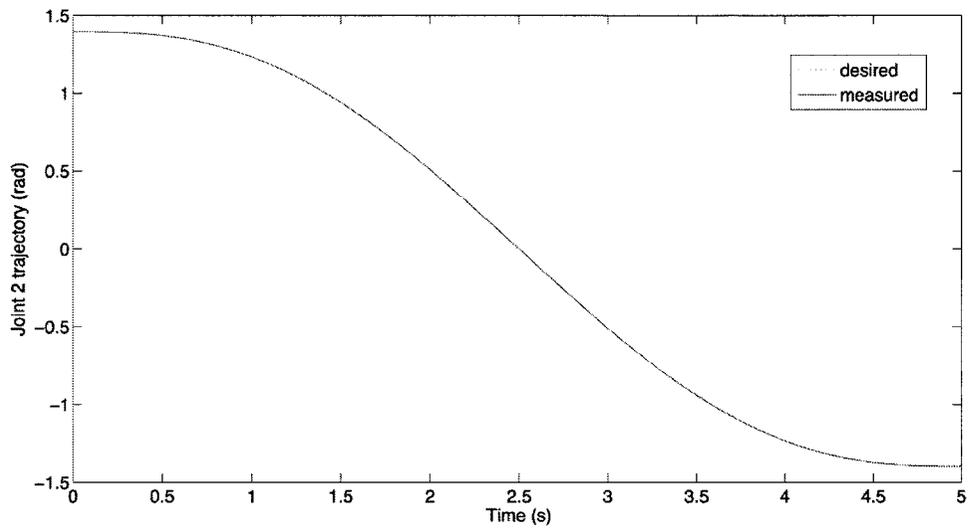
The joint position tracking error was on the order of  $10^{-3}$  rad for each joint over the trajectory. The error appeared to spike near the beginning and end of the trajectory when there was lower velocities and accelerations. This may be due to modelling errors. The calibration process was performed for a data set from motion that was very dynamic. Perhaps a slower trajectory added to the calibration process would better excite the gravitational terms and show better tracking results at low joint velocities. The error in the fourth joint showed oscillatory behaviour and a larger steady state error compared to the other joints. The oscillations may be due to modelling errors or perhaps a slight misalignment between the motor output shaft and harmonic drive.

## 5.2 End Effector Trajectory Tracking

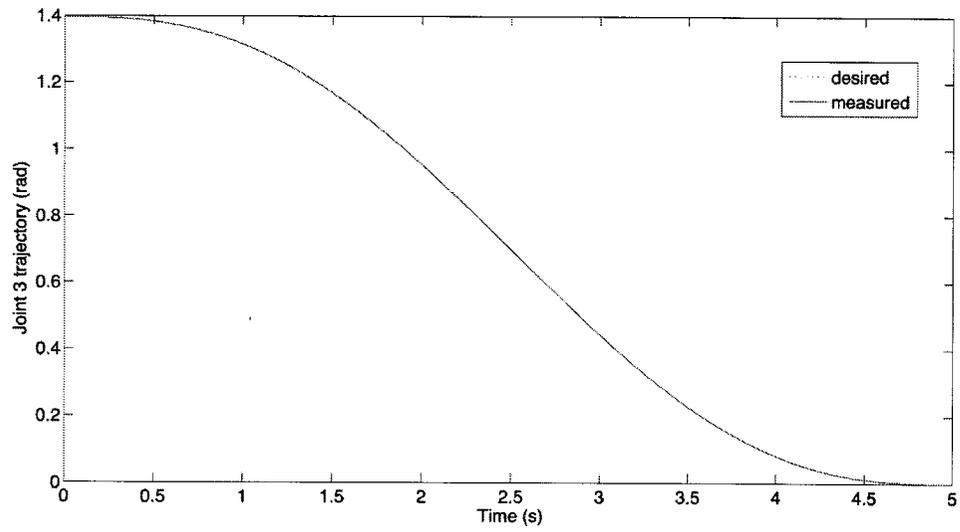
A circular end effector trajectory was applied with the inverse kinematics and redundancy resolution scheme. The purpose of this experiment was to assess the trajectory



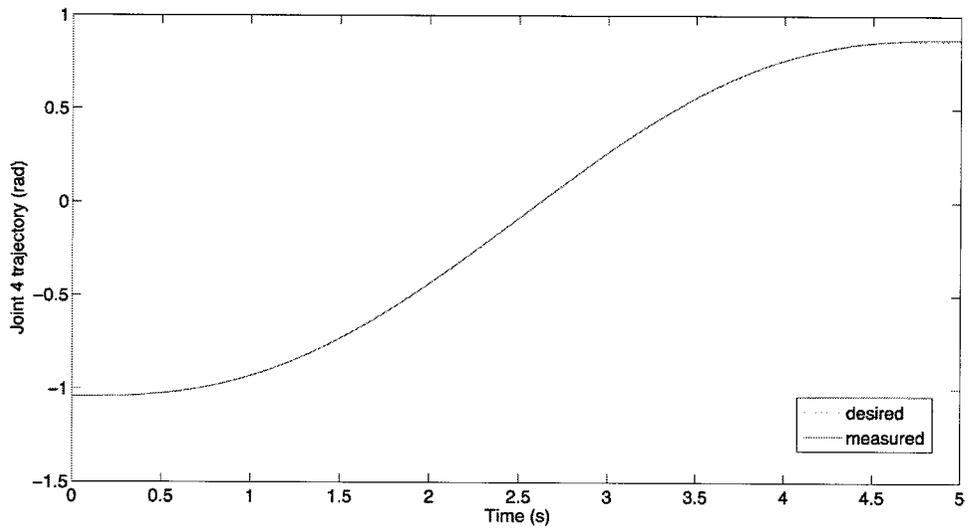
**Figure 5.2:** Position trajectory tracking for joint 1.



**Figure 5.3:** Position trajectory tracking for joint 2.



**Figure 5.4:** Position trajectory tracking for joint 3.



**Figure 5.5:** Position trajectory tracking for joint 4.

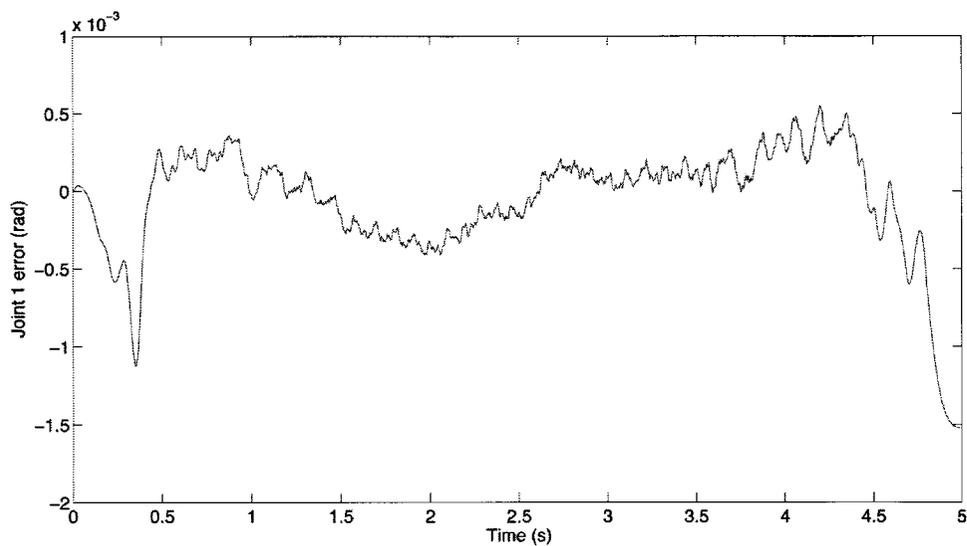


Figure 5.6: Tracking position error for joint 1 smooth step trajectory.

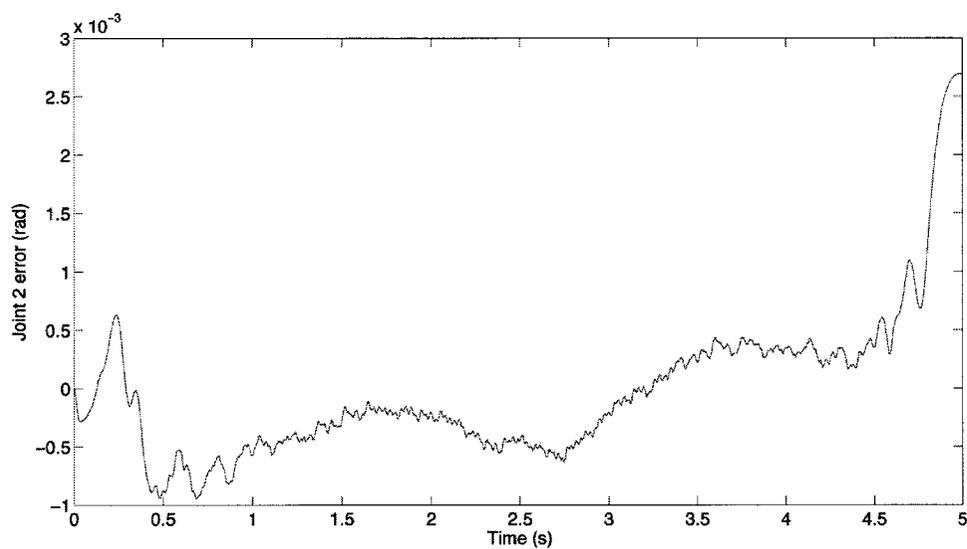
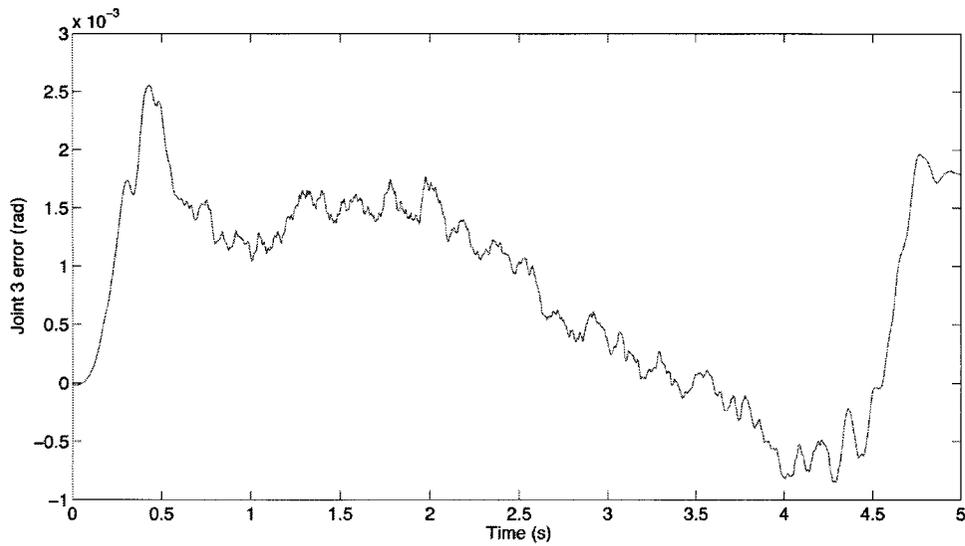
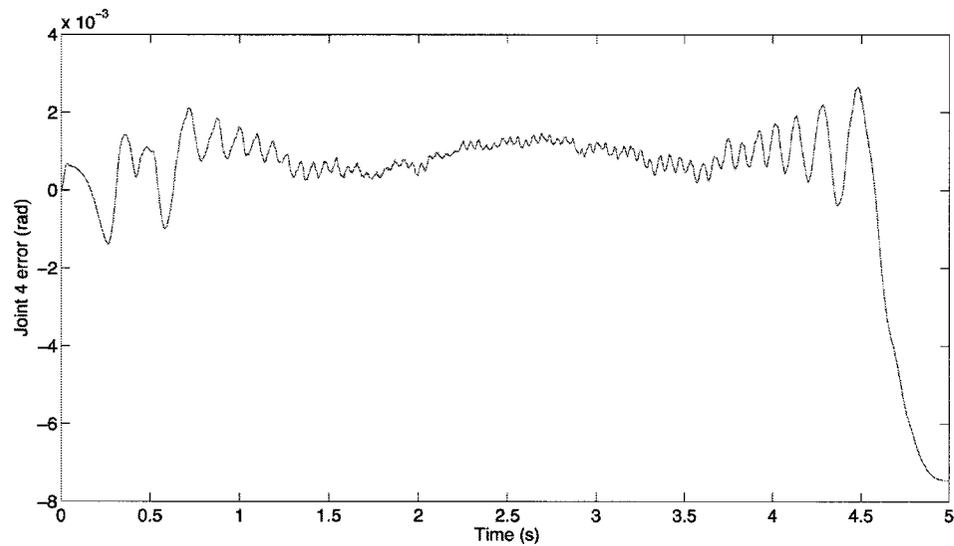


Figure 5.7: Tracking position error for joint 2 smooth step trajectory.



**Figure 5.8:** Tracking position error for joint 3 smooth step trajectory.



**Figure 5.9:** Tracking position error for joint 4 smooth step trajectory.

tracking performance of the controller in Cartesian space. A 5th order blend trajectory was first generated between the initial position  $X_{h_0} = [0.7, 0.45, 0]^T$  and the start of the circular trajectory over 5 seconds. The periodic circular trajectory generator then provided the desired end effector positions, velocities and accelerations as

a function of time (4.4). The specified end effector circle position and oscillating end effector angle parameters are provided in Table 5.2. A description of these parameters are provided in Section 4.1.2.

**Table 5.2:** Circular end effector trajectory parameters.

Parameter	Value	Units
$c_x$	0.65	(m)
$c_y$	0.45	(m)
$r_c$	0.15	(m)
$\omega_c$	$\frac{2\pi}{5}$	(rad)
$\psi_b$	0	(rad)
$\psi_a$	0.25	(rad)
$\psi_p$	0	(rad)

### 5.2.1 Results

Figure 5.10 shows an x-y plot of the desired and measured position trajectory with one circle completed. The starting point  $X_{h_0}$  and direction of motion are shown in the plot. From  $X_{h_0}$ , the trajectory then moves to the initial values of the circle trajectory ( $p_{s_0}, v_{s_0}, a_{s_0}$ ) and continues to perform one period of circular motion over 5 seconds. Figure 5.11 shows an x-y plot of the error during the test. Figures 5.12 to 5.14 show the desired and measured end effector trajectories. Figures 5.15 to 5.17 show the end effector position errors over the trajectory.

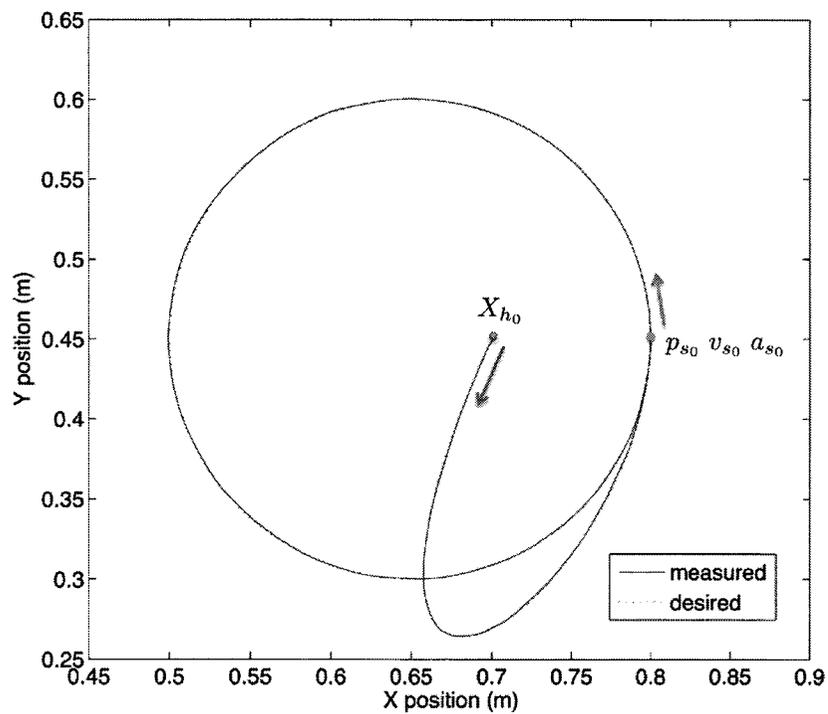


Figure 5.10: End effector X-Y plane position trajectory tracking.

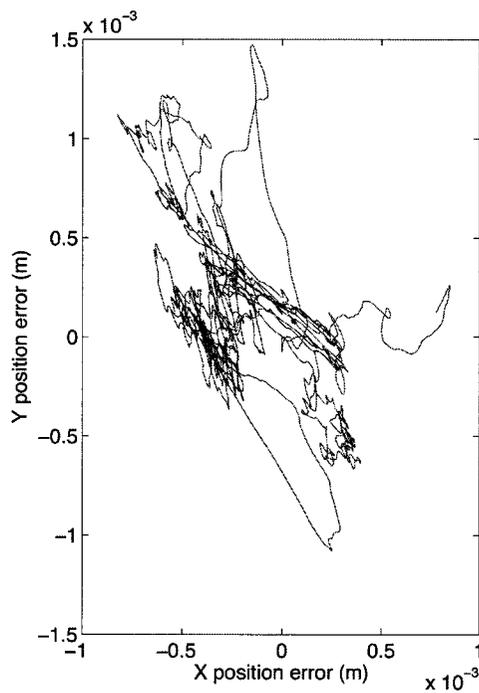


Figure 5.11: End effector X-Y plane position trajectory error.

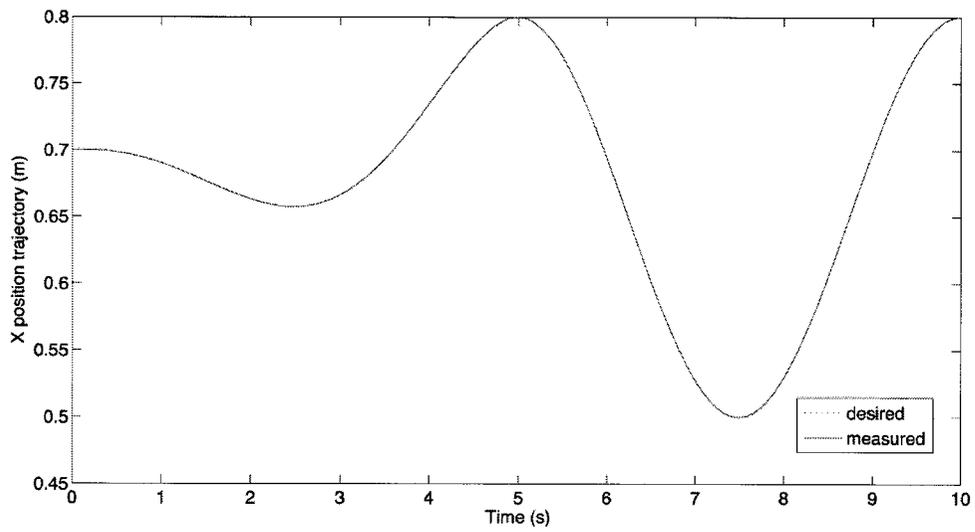


Figure 5.12: End effector X position trajectory tracking.

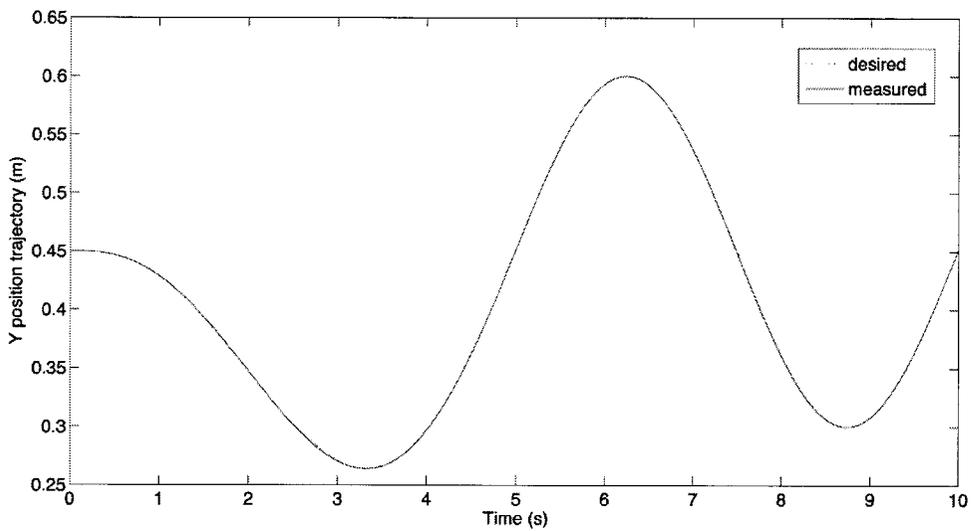
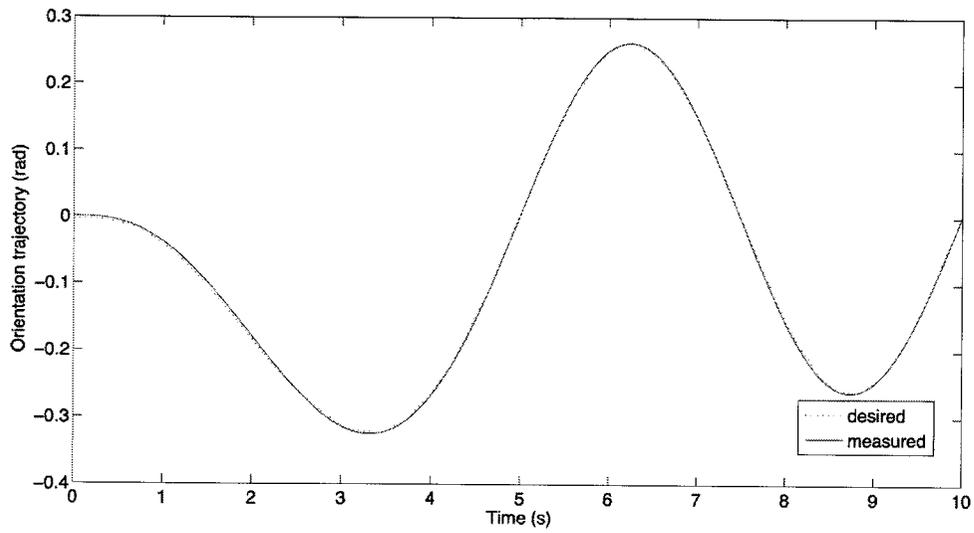
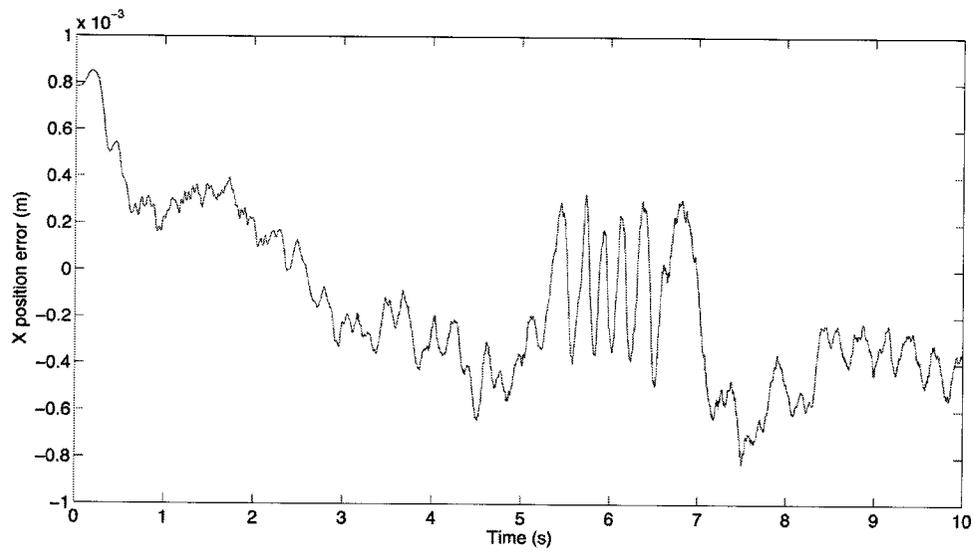


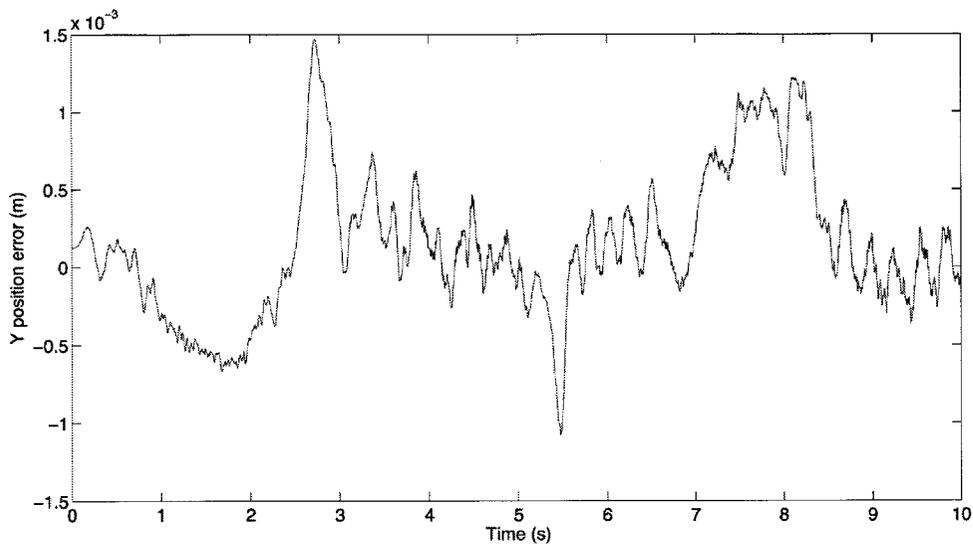
Figure 5.13: End effector Y position trajectory tracking.



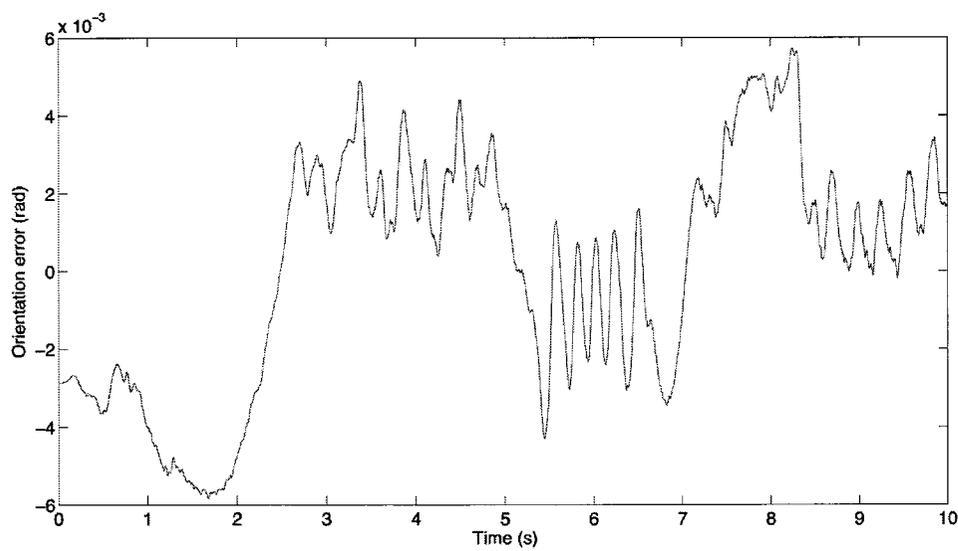
**Figure 5.14:** End effector orientation trajectory tracking.



**Figure 5.15:** Tracking X position error for end effector smooth step trajectory and circular path following.



**Figure 5.16:** Tracking Y position error for end effector smooth step trajectory and circular path following.



**Figure 5.17:** Tracking orientation error for end effector smooth step trajectory and circular path following.

### 5.2.2 Interpretation of Results

For this task space trajectory, the feedback control is still performed in the joint space and the errors in the task space trajectories may be attributed to the kinematic transformations between the end effector and joint space, as well as dynamic model uncertainties. When given only joint space trajectories, the parameter-regressor dynamic model is applied. When providing task space trajectories, the inverse kinematics require use of the linkage lengths. The accuracy of the basic kinematic calibration procedure is uncertain and errors in the calibrated link lengths may have contributed to the errors in the task space trajectory tracking and force control. It is also important to note that the measured end effector trajectory is based on the joint angles with forward kinematics of the robot and not external independent measurements. The results in this section are an assessment of the controller performance and are not a measure of the accuracy of the end effector position.

The position error for task space trajectory tracking is sub-millimetre for the x direction (Figure 5.15) and slightly higher at points in the y direction (Figure 5.16). This might be due to the configuration of the robot where the links are extended further in the x direction, so that errors in joints angles produce larger y direction errors compared to the x direction. Figure 5.16 also shows peaks near points in the trajectory with zero velocity in the y direction. The higher errors in the joint level control at lower velocities, as seen in the previous section, may be contributing to this error.

Errors in the end effector orientation (Figure 5.17) are on the order of  $10^{-3}$  rad. There appears to be some oscillations in the error over the trajectory. This may be attributed to the oscillations seen in the fourth joint trajectory tracking in the previous section, or perhaps is from errors in the estimated link lengths.

The performance assessment for both joint and task space tracking is limited in

that the step trajectory blends did not push the capabilities of ViGRR near its limits. A more systematic performance assessment can be provided in the future, where a number of trajectories are tested that demonstrate maximum velocities, accelerations, and payloads of the device.

### 5.3 Demonstration of Admittance Control With Circular Visual Feedback

The admittance force controller was applied with a user and visual feedback. The required task to be performed was to have the person using the robot follow a circular path by manipulating a hand grip attached to the end effector of the robot. The purpose of this experiment is to determine whether the user can follow a tracking task when the robot is in force-controlled mode. The results will motivate future experiments and tests for interaction using the lower extremities.

The admittance control presented in Section 4.3.2 is visualized as a flowchart in Figure 5.18 which includes the admittance controller in the red bounding box and a visual feedback component. In this control mode, interaction forces between the robot and user deviates the desired trajectory according to the impedance model (4.25). High stiffness and damping coefficients for the admittance Cartesian space dynamic model provided a rigid interface between the robot and human whereas soft impedances allowed a faster response and causes the robot to be less apparent to the user. The visual feedback trajectory  $X_h^v$  does not affect the controller and is only used to show the person manipulating the robot what to follow. The measured position is also fed into the 2D x-y plot visualization.

The test setup involves a monitor placed in front of the user while the robot is manipulated by hand with a hand grip attached to the end effector. Figure 5.19 shows

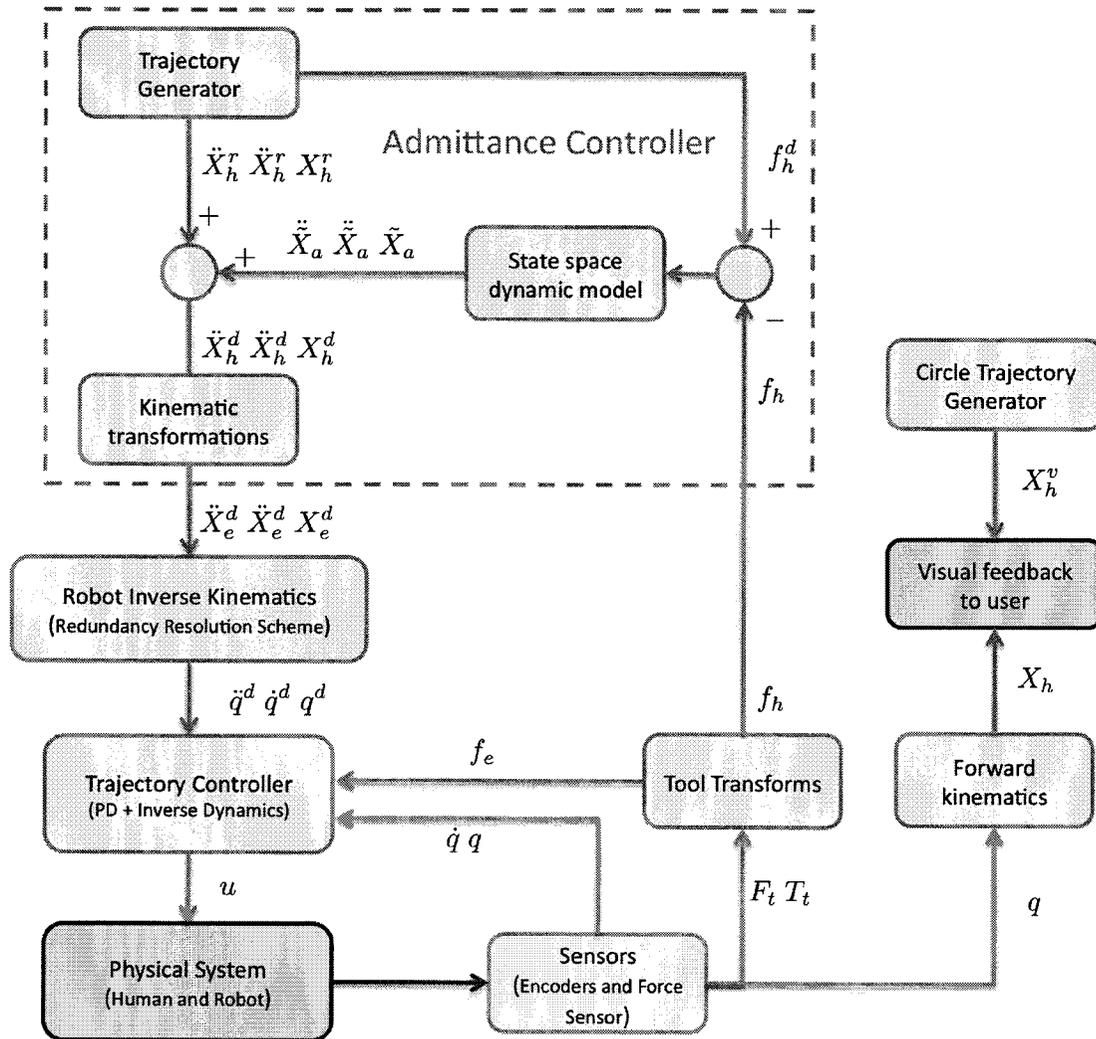
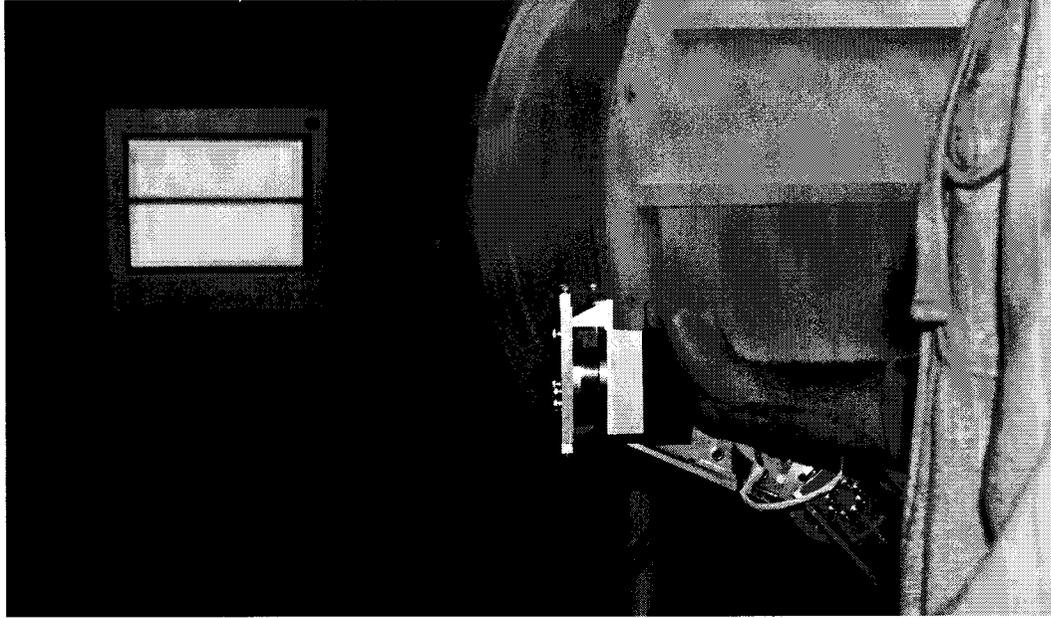


Figure 5.18: Admittance control with visual feedback flow chart.

a person holding the hand grip while watching the monitor and visual feedback. The visualization displays two arcs on a plot that are updated at 10 Hz with one second of past data plotted on the screen before disappearing. Figure 5.20 shows the x-y plot with the two arcs. One arc is the measured trace of the hand grip location and the other is the circular trajectory the person is trying to follow. The broken line shows the full circle trajectory, but only the arc is displayed for the user.

In this experiment, the impedance model was set to provide resistance to the



**Figure 5.19:** Test setup for circle trajectory following task.

motion with a specified viscous damping coefficient, zero stiffness, and a point mass. The reference trajectory is the initial position of the hand grip which corresponds to the centre of the circular trajectory. The reference orientation of the grip was set to zero degrees and the impedance model provided a damping and stiffness for resistance to rotating the end effector away from zero degrees. There was no desired grip orientation the user needed to follow. Table 5.3 lists the parameters for generating the circular visual feedback as the desired trajectory. Table 5.4 displays the controller parameters used in this experiment. Data was logged over a 60 second period.

**Table 5.3:** Parameters used to generate circular end effector position  $X_h^v$  for visual feedback.

Parameter	Value	Units
$c_x$	0.7	(m)
$c_y$	0.35	(m)
$r_c$	0.1	(m)
$\omega_c$	$\frac{2\pi}{5}$	(rad)

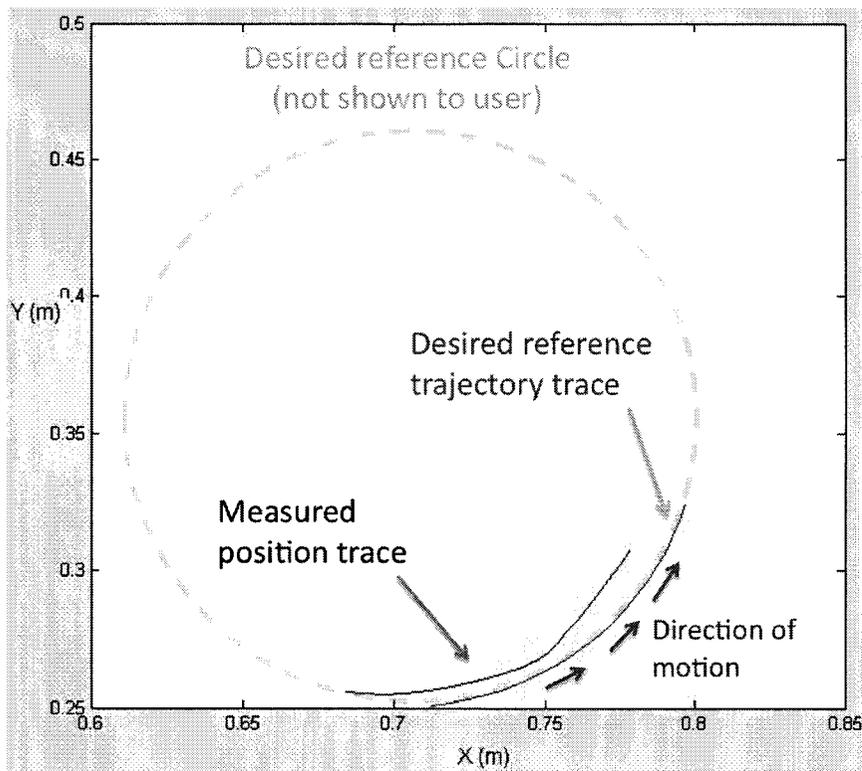


Figure 5.20: Visualization feedback for user.

### 5.3.1 Results

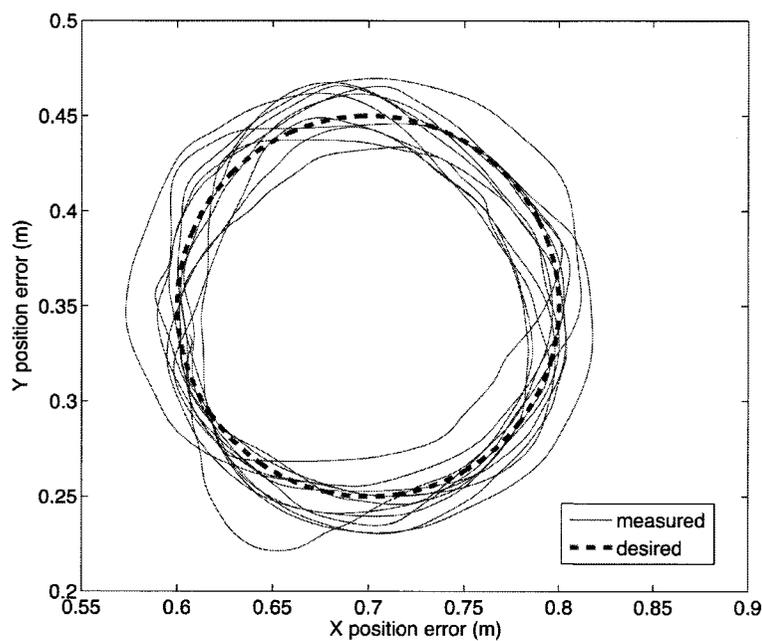
Figure 5.21 shows an x-y plot of the desired and measured hand grip positions. The desired circle provided as visual feedback is in blue and the measured grip position is in red. The desired positions refer to the visual feedback  $\hat{X}_h^v$  and the measured trajectory refers to the actual position of the grip  $X_h$ . Figure 5.22 shows an x-y plot of the error while the user was trying to follow the circle trajectory. Figures 5.23 and 5.24 show the desired and measured x and y hand grip positions. Figures 5.25 to 5.26 show the hand grip position errors.

### 5.3.2 Interpretation of the Results

Figure 5.21 shows that the user performed well in attempting to follow the trajectory without any assistance, drawing 12 circles over the course of 60 seconds. The user

**Table 5.4:** Admittance control experiment parameters.

Parameter	Value	Units
$M_I$	[1, 1, 0.1]	(kg, kg, Nm <sup>2</sup> )
$B_I$	[150, 150, 9]	(Ns/m, Ns/m, Ns/rad)
$K_I$	[0, 0, 30]	(N/m, N/m, Nm/rad)
$f_h^d$	[0, 0, 0]	(N, N, Nm)
$X_h^r$	[0.7, 0.35, 0]	(m, m, rad)
$\dot{X}_h^r$	[0, 0, 0]	(m/s, m/s, rad/s)
$\ddot{X}_h^r$	[0, 0, 0]	(m/s <sup>2</sup> , m/s <sup>2</sup> , rad/s <sup>2</sup> )

**Figure 5.21:** End effector X-Y plane position trajectory tracking.

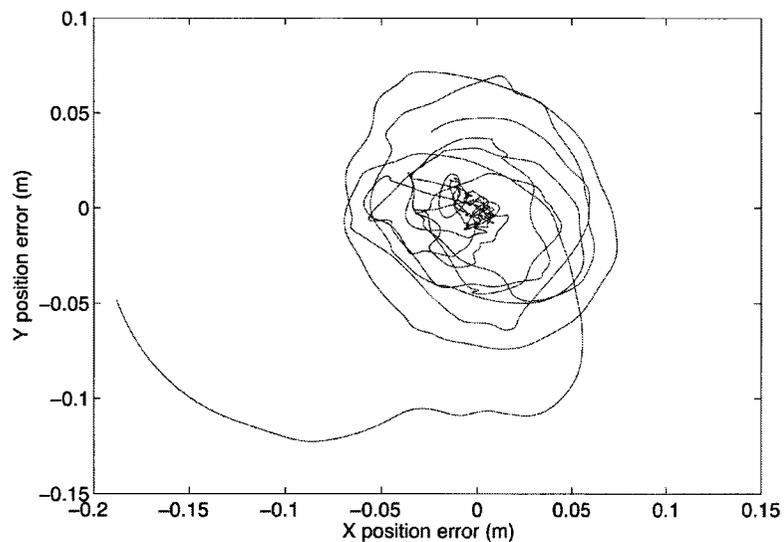


Figure 5.22: End effector X-Y plane position trajectory error.

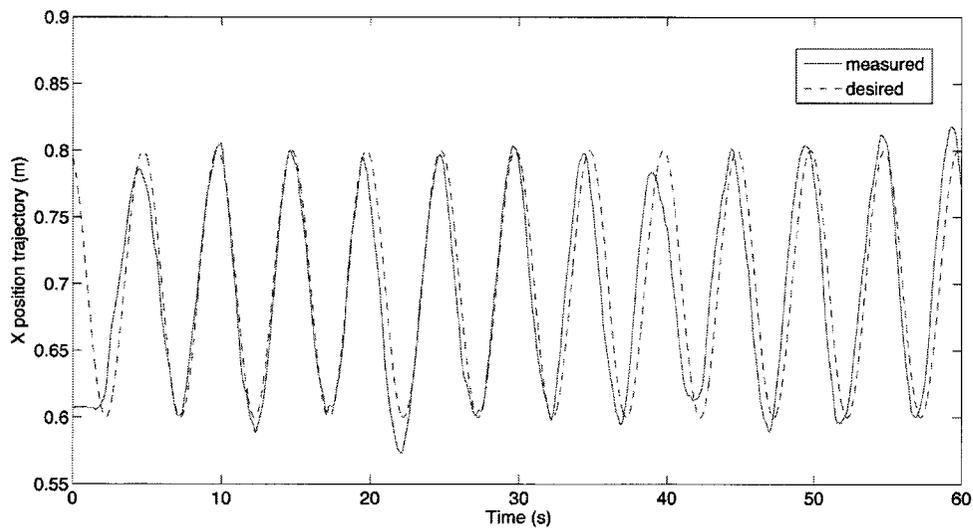


Figure 5.23: End effector X position trajectory tracking.

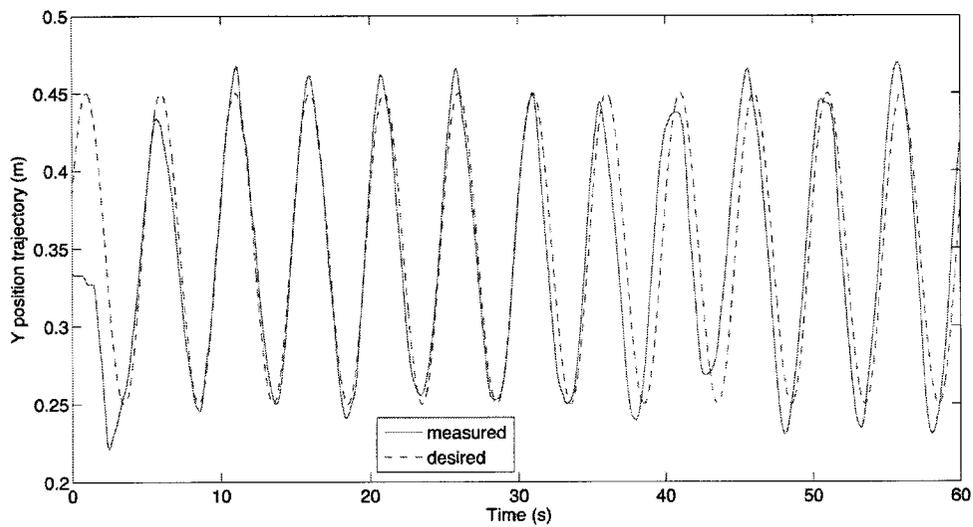


Figure 5.24: End effector Y position trajectory tracking.

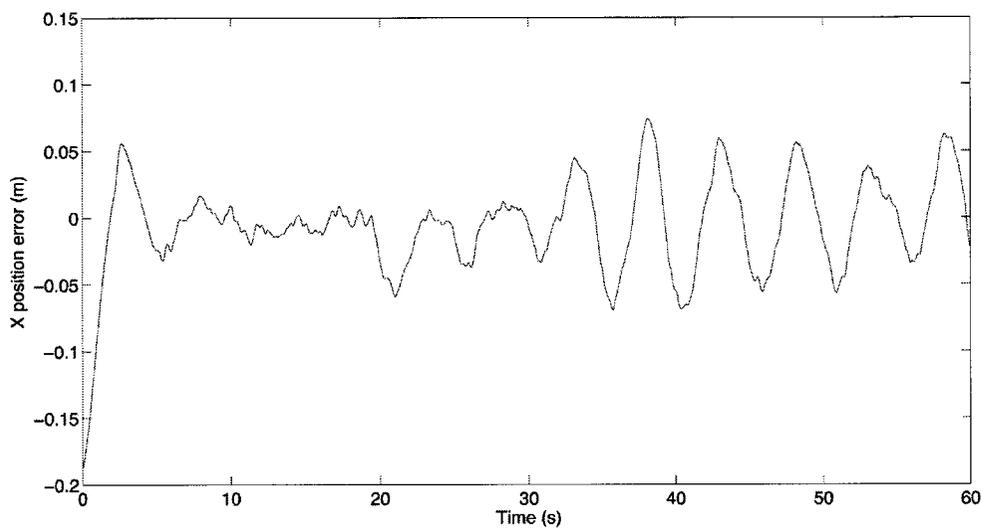
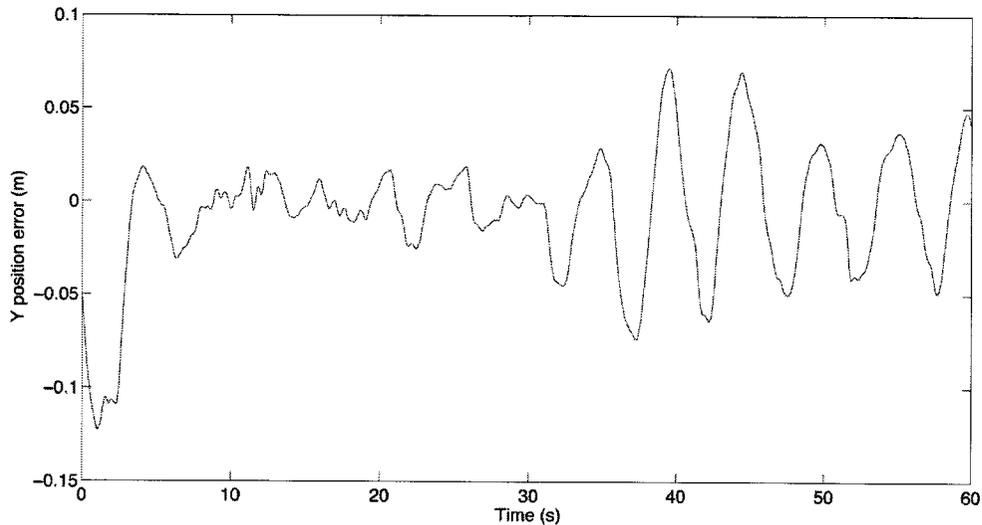


Figure 5.25: Tracking X position error for end effector smooth step trajectory and circular path following.



**Figure 5.26:** Tracking Y position error for end effector smooth step trajectory and circular path following.

appears to have a tendency to move the hand grip toward the bottom left of the circle, however the error in Figure 5.22 does not appear to have a significant bias in any one direction, except for the larger error at the beginning of the trajectory following task. Figure 5.23 and 5.24 show some undershoot and overshoot by the user but tracks the sinusoidal x and y position pattern well. In the second half of the trajectory, the user is anticipating the motion of the moving circle trajectory and Figures 5.25 and 5.26 show the errors having a sinusoidal pattern as the user moves out of phase with the desired circular trajectory.

In this experiment, the force feedback was successfully implemented to allow the user to track a desired path. The large viscous damping coefficients provided a challenging dynamic environment where the response of the motion was delayed from the input force as though the person was dragging a mass through a thick fluid. The person using the robot demonstrated the capability of ViGRR to provide force and visual feedback for performing a motor task with the user in the loop.

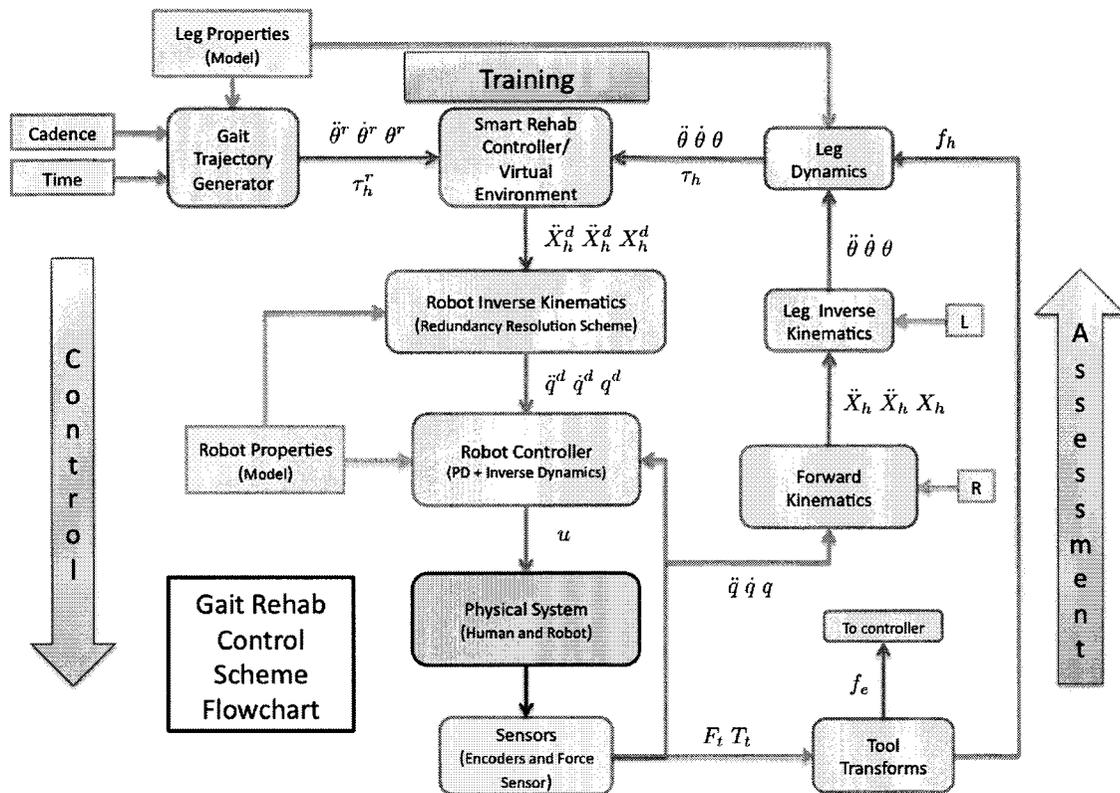
There are, however, limitations to this demonstrative experiment. There were not

multiple participants in the experiment, since the test was intended as a demonstration of the ViGRR capabilities and does not provide any conclusions as to whether the person performing the desired task improved their performance or involved motor learning in any way. The visual feedback was also lacking in that it provided only a simple plot with desired and measured trajectories visualized as moving lines. Another limitation is that the desired visualized trajectory was not part of the control loop and was not used for assistance. If the reference trajectory was set to the circle, the stiffness could be tuned to provide varying levels of assistance, preventing the person from deviated too far from the trajectory. This control mode may be implemented for future validation experiments with a person's leg.

The high damping was also, in part, necessary to filter oscillations caused by noise in the FT sensor, which is already heavily filtered. This experiment involved relatively small loads applied by hand compared to the intended design using a foot and mimicking ground reaction forces. The response of the system using the 3 Hz filter may be limiting for sensing ground reaction forces applied to the foot when the device is used for its intended purpose with a leg. Further experiments will be required to answer such questions and reducing noise in the FT sensor may be required as discussed in Section 4.4.2.

### **5.3.3 Application to ViGRR Rehabilitation Controller Concept**

Having performed the experiment and interpreted the results, the question remains: How does this relate to a lower extremity rehabilitation? The intended application of the ViGRR prototype is to demonstrate its rehabilitation-focused controls and capabilities and the initial concept for the rehabilitation-focused controller was introduced in Section 1.2.1 with Figure 1.6. A more detailed version of this flowchart is presented



**Figure 5.27:** The ViGRR detailed gait rehabilitation controller concept flowchart.

in Figure 5.27 with notation and processes based on the modelling and control of the ViGRR platform presented in this thesis. In this flowchart, the model parameters and inputs are represented as blocks with right angle corners and the processes are represented as blocks with rounded edges. In Figure 5.27, the controller is focused on gait rehabilitation, where the reference trajectory is based on average gait data, desired cadence, and the leg model of the user. This reference trajectory is analogous to the circle trajectory provided in the experiment as the tracking task for the user (Figure 5.18).

In the training portion of the controller, the gait rehabilitation concept may employ a virtual environment, for example, where the user's leg motion and forces in the real world are used as inputs to a dynamic model in a virtual world. The simulation in

the virtual world produces desired outputs for the robot to track and visual feedback of the user's performance compared to the desired task. In the circle hand following experiment, this virtual environment is analogous to the admittance controller with "simulated" task space dynamics that include damping, stiffness and inertia terms. The user's hand motion action is fed back to a visual display, as would be required with a simulated walking environment for the conceptual ViGRR controller. Visual feedback is applied here as a form of training and motivation for the user. However, the reference trajectory can conceivably be used to modify the virtual environment and thus the desired trajectory output to the robot as per an assistance or adaptive algorithm.

The controller portion of the control scheme tracks the desired output from the virtual environment or admittance control process. This component of the gait rehabilitation concept is essentially identical to that of the circle tracking task. Only the kinematic transformations between the human interaction frame and the end effector would be different.

In the assessment portion of the controller concept, modelling of the leg dynamics and kinematics is applied in Figure 5.18. In the admittance control for the circle tracking task, the user's effort is measured by the force/torque sensor and the input force drives the admittance controller. However, for assessing the user's performance, the applied forces are not important to the task at hand. Rather, it is the position of the hand grip that is of interest. Similarly, the measured interacting forces between the foot and the robot, or the foot motion is not necessarily relevant to assessing the person's performance of the task at hand. It is the joint torques and motion that is of interest. Choosing what variables to use in the control loop and what information to feed back to the user will be subject of interest for future study and depends on the type of controller and task being performed.

The circle tracking experiment is limited in that visual feedback is the only mechanism of helping the user perform the desired task. However, the controller contains the main components of the conceptual rehabilitation-based controls for ViGRR and demonstrates the capability of the device to perform force control with a user in the loop.

## 5.4 Summary

Three experiments were performed to assess the tracking performance and demonstrate the ability of ViGRR to provide a challenging motor task with visual feedback for the user. The performance characteristics of the robot are satisfactory, but an improved kinematic calibration is required to properly assess the actual end effector tracking performance of the robot. The admittance controller was shown to successfully provide a dynamic force feedback response to the user and visual feedback while they were manipulating ViGRR in an attempt to perform a motor task.

## Chapter 6

### Conclusions and Future Work

Gait rehabilitation robotics present appealing potential as a tool for clinical application with patients afflicted by neurological injuries or disorders. Principles of behaviour mediated plasticity can be facilitated by the application of robotics to provide high intensity exercise that is repetitive, targeted, challenging, and motivational. A number of robotic devices and control strategies for gait rehabilitation have been implemented to provide these potential advantages and ultimately improve outcomes for patients. However, devices largely based on BWSTT, such as the Lokomat, have not shown significant outcome improvements compared to traditional therapist driven strategies. Most gait rehabilitation devices attempt to enhance or replace therapist intervention while performing a treadmill-based exercise. The preliminary design and controller concept of ViGRR was initiated to focus on the principles of behaviour-initiated plasticity in a way that is more aligned with upper extremity rehabilitation robots. ViGRR was conceived to provide flexible control strategies for rehabilitation focused on subtasks using the lower extremities, thus improving outcomes for patients based on their specific needs and progression. The form factor of a stationary end effector design with supine or upright configurations will allow for ease of use, greater variety of exercises, and application to large range of patient capabilities. The objective of this thesis was to design and implement the ViGRR prototype for future

research in control strategies for gait rehabilitation and to motivate a full scale implementation. This goal was met with the ViGRR prototype realized as a planar 4 degree of freedom manipulator and its potential application as an interactive rehabilitation platform was demonstrated through a simple force controlled experiment.

## 6.1 Conclusions

The process of achieving the system realization involved modelling of the system, requirements analysis, iterative detailed design for the apparatus, controller implementation, calibration, and experimentation. An overview of the completed work in this process and conclusions drawn from the system implementation and experimentation are provided as follows:

### Leg and Robot Modelling

The ViGRR concept involves interaction with a person and their lower extremities. For use with the ViGRR prototype, the leg was modelled as a linkage system with revolute joints for the hip knee and ankle. The dynamic leg model was deemed acceptable for the design and simulation and was partially validated by comparisons with published average gait data showing similar trends in the joint moments and ground reaction forces (Figures 2.13 to 2.16). Validation of the model with a human occupant using the ViGRR prototype will be a part of future work in the project as described later in this chapter.

The ViGRR dynamic model was derived based on the Euler-Lagrange equations of motion and actuator drive train dynamics. The model was verified by comparison with the iterative Newton-Euler formulation of dynamics (Section 2.3.7) and implemented in the controller. A dynamic calibration was performed in Section 4.6, validating the robot model for use in simulation and control.

## **Requirements Analysis**

As a versatile rehabilitation device, the ViGRR prototype was designed to have a large reachable workspace allowing for various leg motions other than gait trajectories. The desired reachable workspace is presented in Figures 3.4 and 3.5. The loading requirements were based on gait trajectories due to the challenging trajectory and applied load involved in ambulation. The end effector requirements were generated using the workspace visualization tool in Section 3.1.3.

## **Detailed Design**

A simulation of the robot matching the desired end effector trajectories provided the basis for selecting appropriate drive train components. A nonlinear optimization routine was created to select various parameters of the robot such as the linkage lengths. These parameters are provided in Table 3.5. The selected drive train components are provided in Table 2.4 and the apparatus details are given in Section 3.4.

## **Controller implementation**

A model-based nonlinear controller was used for ViGRR. The model dynamics were reformulated as a parameter-regressor formulation and the controls were implemented in a Simulink model with QuaRC for hard real-time processing and interfacing with the DAQ. The software control modes presented in Section 4.4.3 were tested with the robot and a GUI was created for visualization and user interaction (Figures 4.8 and 4.7).

## **Calibration**

A basic kinematic calibration of the ViGRR prototype was performed with estimates

of the linkage lengths based on self motion of the robot and a measured end effector location. A kinematic calibration with more accurate external measurements of the end effector is recommended in future work. The basic calibration yielded the linkage lengths displayed in Table 4.4. A dynamic calibration was performed for each linkage separately as well as for the integrated robot. The final dynamic parameter from the calibration process was found to be (4.43) in Section 4.6.3.

## **Experimentation**

Joint space and Cartesian space tracking experiments were performed to assess the controller and system performance of ViGRR. The joint tracking results are presented in Figures 5.2 through 5.9 for joint tracking and Figures 5.10 through 5.17 for the Cartesian space tracking. The results showed good performance with sub-millimetre errors in Cartesian space position, which is adequate for the desired application of manipulating a person's leg. Further testing will be required to properly assess the robot performance capabilities in the presence of force disturbances and faster movements.

An experiment was performed where the user manipulated the end effector by hand in a force-controlled mode. The desired task was to have the person follow a reference circular trajectory from a display. The resulting motion of the hand was compared to the desired trajectory over a 60 second period in Figures 5.21 to 5.26. This experiment demonstrated the capability of the robot to provide a task oriented control framework (Fig 5.18) analogous to the ViGRR rehabilitation controller concept (Fig 5.27).

## **6.2 Future Work**

The ViGRR system was integrated and tested with satisfactory results. However, some limitations in its implementation are identified throughout the thesis that should

be addressed in future work on the ViGRR prototype. The hardware and apparatus modifications in future work are first presented so that the system can become fully functional with a person's leg interacting with ViGRR. Future directions using ViGRR in terms of research, clinical application and controls are then presented.

### **Apparatus and Systems Development**

The basic kinematic calibration is insufficient for precision control of the robot in the task space. A calibration with multiple configurations and end effector positions with known accuracy should be administered. This can be accomplished using a motion capture vision system or a coordinate measurement machine.

Another identified issue in the mechanical design is the noise in the FT sensor and low cutoff frequencies required in the filter. A potential solution proposed in Section 4.4.2 is to modify the location of the FT sensor on the end effector to lower the torque loading requirements on the sensor and allow for a lower calibration with less range but better signal to noise characteristics. An analysis of the required bandwidth of the FT sensor for a haptic controller with a person's leg should be conducted.

An accurate kinematic calibration and improved FT sensor characteristics can be used for new performance testing experiments that quantify the range of operation the robot can perform within specified accuracy limits. These experiments can also include payload analysis, perhaps with a dummy leg.

Refinements to the mechanical design will be required before having a user manipulate the robot with his or her leg. The robot stage is currently on a rigid table with mdf particle board and requires a redesign with a rigid platform and a seat and/or harness system for the user. Wiring management, foot plate design, and safety impact bars also need to be refined.

Finally, a user can use the device with his or her leg. A validation of the leg model can be performed with a motion tracking vision system analyzing the leg's motion

and comparing it to the inverse kinematics calculations from the robot sensors. More instrumentation may be required to provide a better assessment of the leg kinematics during operation of ViGRR. For the leg inverse dynamics, it is much more difficult to validate. Perhaps a detailed simulation with musculoskeletal models and motion capture data can be compared to the ViGRR leg model. Parameter estimation of the leg model properties may also be attempted using the ViGRR prototype.

If the ViGRR robot is to be used for real-time estimation of the leg kinematics in a dynamic simulation (or for the leg inverse dynamics calculation), knowledge of the leg joint velocities and accelerations are also required. Position sensors (ie. robot encoders) become very noisy from the effects of sensor quantization when differentiated to obtain accelerations. A nonlinear estimator such as a particle filter or Unscented Kalman filter may be implemented to estimate these accelerations.

### **Research in Controllers and Clinical Application**

A fully capable ViGRR prototype poses exciting opportunities for research in haptics, virtual environments, and rehabilitation-focused controllers. The capability of the robot to simulate a person's foot interacting with a floor, for example, can be investigated to determine if a varying centre of pressure on the user's foot can be felt. Research in collision models, stable haptic control, washout, etc. contribute to the development of ViGRR as a haptic display. 3D visualizations and potentially virtual reality headsets or 3D displays may also be investigated for improved immersion, biofeedback and motivation for the patient. These technologies can form the basis of rehabilitation-focused controller where the sandbox nature of a virtual environment provides opportunities for selective assistance, targeted subtasks, or other control schemes.

In order to move toward using ViGRR in a clinical rehabilitation context, strong

collaboration and integration with clinical personnel, researchers and patients is required. There are many individuals involved in the rehabilitation process of a patient, from family to friends, doctors to nurses, engineers to social workers, physiotherapists to podiatrists, everyone has a role in the patient's health and recovery. It must be determined how ViGRR can play a role in the context of this multidisciplinary team dedicated to the rehabilitation of the patient. It is known that ViGRR will operate at a rehabilitation centre and provide repetitive exercise in an effort to improve functional outcomes for the lower extremities. However, details relating to the type of patients and disorders, length of treatments, duration of exercises, types of exercises, patient motivation, assessment measurements, and interactions between the patient and therapist are very much contextual. With the flexibility of the ViGRR platform, these factors affecting the rehabilitation process and patient outcomes are daunting to tackle. However, as research in rehabilitation and robotics continues to mature, clinical trials will begin to demonstrate more effective therapy strategies for a greater variety of injuries or disorders. One can hope that as these new rehabilitation therapies emerge, ViGRR will deliver, build upon, modify, develop, and research new strategies that will elicit further insights into what is effective, how, and in what context.

The undertaking of the ViGRR project is ambitious, in that further development toward a full scale system, as outlined in Section 1.2, will involve collaboration between many people in multiple disciplines and areas of expertise with significant development time. Toward the full scale realization, the first prototype was designed and built as detailed in this thesis, and will provide a starting point for further development of the platform.

## List of References

- [1] H. Krebs, L. Dipietro, S. Levy-Tzedek, S. Fasoli, A. Rykman-Berland, J. Zipse, J. Fawcett, J. Stein, H. Poizner, A. Lo, B. Volpe, and N. Hogan, "A paradigm shift for rehabilitation robotics," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 4, pp. 61–70, 2008.
- [2] J. J. Daly and R. L. Ruff, "Construction of efficacious gait and upper limb functional interventions based on brain plasticity evidence and model-based measures for stroke patients.," *ScientificWorldJournal*, vol. 7, pp. 2031–2045, 2007.
- [3] B. H. Dobkin, "Strategies for stroke rehabilitation.," *Lancet Neurol*, vol. 3, pp. 528–536, Sep 2004.
- [4] A. L. Behrman, M. G. Bowden, and P. M. Nair, "Neuroplasticity after spinal cord injury and training: an emerging paradigm shift in rehabilitation and walking recovery.," *Phys Ther*, vol. 86, pp. 1406–1425, Oct 2006.
- [5] S. Hesse, H. Schmidt, C. Werner, and A. Bardeleben, "Upper and lower extremity robotic devices for rehabilitation and for studying motor control.," *Curr Opin Neurol*, vol. 16, pp. 705–710, Dec 2003.
- [6] L. Marchal-Crespo and D. J. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury.," *J Neuroeng Rehabil*, vol. 6, p. 20, 2009.
- [7] J. A. Kleim and T. A. Jones, "Principles of experience-dependent neural plasticity: implications for rehabilitation after brain damage.," *J Speech Lang Hear Res*, vol. 51, pp. S225–S239, Feb 2008.
- [8] L. W. Forrester, L. A. Wheaton, and A. R. Luft, "Exercise-mediated locomotor recovery and lower-limb neuroplasticity after stroke.," *J Rehabil Res Dev*, vol. 45, no. 2, pp. 205–220, 2008.

- [9] S. Hesse and C. Werner, "Connecting research to the needs of patients and clinicians," *Brain Res Bull*, vol. 78, pp. 26–34, Jul 2008.
- [10] A. W. Dromerick, P. S. Lum, and J. Hidler, "Activity-based therapies," *NeuroRx*, vol. 3, pp. 428–438, Oct 2006.
- [11] E. T. Wolbrecht, V. Chan, D. J. Reinkensmeyer, and J. E. Bobrow, "Optimizing compliant, model-based robotic assistance to promote neurorehabilitation," *IEEE Trans Neural Syst Rehabil Eng*, vol. 16, pp. 286–297, Jun 2008.
- [12] J. Liu, S. C. Cramer, and D. J. Reinkensmeyer, "Learning to perform a new movement with robotic assistance: comparison of haptic guidance and visual demonstration," *J Neuroeng Rehabil*, vol. 3, p. 20, 2006.
- [13] H. I. Krebs and N. Hogan, "Therapeutic robotics: A technology push," *Proc IEEE Inst Electr Electron Eng.*, vol. 94, no. 9, pp. 1727–1738, 2006.
- [14] N. Hogan, H. I. Krebs, B. Rohrer, J. J. Palazzolo, L. Dipietro, S. E. Fasoli, J. Stein, R. Hughes, W. R. Frontera, D. Lynch, and B. T. Volpe, "Motions or muscles? some behavioral factors underlying robotic assistance of motor recovery," *J Rehabil Res Dev*, vol. 43, no. 5, pp. 605–618, 2006.
- [15] J. Hidler, L. F. Hamm, A. Lichy, and S. L. Groah, "Automating activity-based interventions: The role of robotics," *J Rehabil Res Dev*, vol. 45, no. 2, pp. 337–344, 2008.
- [16] M. Lee, M. Rittenhouse, and H. A. Abdullah, "Design issues for therapeutic robot systems: Results from a survey of physiotherapists," *Journal of Intelligent and Robotic Systems*, vol. 42, pp. 239–252, Mar. 2005.
- [17] M. J. Johnson, "Recent trends in robot-assisted therapy environments to improve real-life functional performance after stroke," *J Neuroeng Rehabil*, vol. 3, p. 29, 2006.
- [18] A. C. Sy and N. T. Bugtai, "Towards the development of more human-like robotic therapy devices: A review," in *Proc. IEEE Workshop Health Care Management (WHCM)*, pp. 1–6, 2010.
- [19] G. Colombo, M. Wirz, and V. Dietz, "Driven gait orthosis for improvement of locomotor training in paraplegic patients," *Spinal Cord*, vol. 39, pp. 252–255, May 2001.

- [20] L. Bensoussan, J.-M. Viton, N. Barotsis, and A. Delarque, "Evaluation of patients with gait abnormalities in physical and rehabilitation medicine settings.," *J Rehabil Med*, vol. 40, pp. 497–507, Jul 2008.
- [21] C. Kirtley, *Clinical gait analysis : theory and practice*. New York: Elsevier, 2006.
- [22] D. H. Sutherland, "The evolution of clinical gait analysis part I: kinesiological emg.," *Gait Posture*, vol. 14, pp. 61–70, Jul 2001.
- [23] D. H. Sutherland, "The evolution of clinical gait analysis. part ii kinematics.," *Gait Posture*, vol. 16, pp. 159–179, Oct 2002.
- [24] D. H. Sutherland, "The evolution of clinical gait analysis part iii—kinetics and energy assessment.," *Gait Posture*, vol. 21, pp. 447–461, Jun 2005.
- [25] J. Hidler, "Robotic-assessment of walking in individuals with gait disorders.," *Conf Proc IEEE Eng Med Biol Soc*, vol. 7, pp. 4829–4831, 2004.
- [26] H. Huang, S. L. Wolf, and J. He, "Recent developments in biofeedback for neuromotor rehabilitation.," *J Neuroeng Rehabil*, vol. 3, p. 11, 2006.
- [27] B. R. Brewer, R. Klatzky, H. Markham, and Y. Matsuoka, "Investigation of goal change to optimize upper-extremity motor performance in a robotic environment.," *Dev Med Child Neurol*, vol. 51 Suppl 4, pp. 146–153, Oct 2009.
- [28] P. Culmer, A. Jackson, M. C. Levesley, J. Savage, R. Richardson, J. A. Cozens, and B. B. Bhakta, "An admittance control scheme for a robotic upper- limb stroke rehabilitation system," in *Proc. 27th Annual International Conference of the Engineering in Medicine and Biology Society IEEE-EMBS 2005*, pp. 5081–5084, 2005.
- [29] R. Riener, L. Lunenburger, S. Jezernik, M. Anderschitz, G. Colombo, and V. Dietz, "Patient-cooperative strategies for robot-aided treadmill training: first experimental results," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 380–394, 2005.
- [30] R. Colombo, F. Pisano, A. Mazzone, C. Delconte, S. Micera, M. C. Carrozza, P. Dario, and G. Minuco, "Design strategies to improve patient motivation during robot-aided rehabilitation.," *J Neuroeng Rehabil*, vol. 4, p. 3, 2007.
- [31] H. Huang, T. Ingalls, L. Olson, K. Ganley, T. Rikakis, and J. He, "Interactive multimodal biofeedback for task-oriented neural rehabilitation.," *Conf Proc IEEE Eng Med Biol Soc*, vol. 3, pp. 2547–2550, 2005.

- [32] K. Manal and S. J. Stanhope, "A novel method for displaying gait and clinical movement analysis data.," *Gait Posture*, vol. 20, pp. 222–226, Oct 2004.
- [33] L. M. Schutte, U. Narayanan, J. L. Stout, P. Selber, J. R. Gage, and M. H. Schwartz, "An index for quantifying deviations from normal gait.," *Gait Posture*, vol. 11, pp. 25–31, Feb 2000.
- [34] M. Visintin, H. Barbeau, N. Korner-Bitensky, and N. E. Mayo, "A new approach to retrain gait in stroke patients through body weight support and treadmill stimulation.," *Stroke*, vol. 29, pp. 1122–1128, Jun 1998.
- [35] N. D. Neckel, N. Blonien, D. Nichols, and J. Hidler, "Abnormal joint torque patterns exhibited by chronic stroke subjects while walking with a prescribed physiological gait pattern.," *J Neuroeng Rehabil*, vol. 5, p. 19, 2008.
- [36] L. Lunenburger, G. Colombo, R. Riener, and V. Dietz, "Clinical assessments performed during robotic rehabilitation by the gait training robot lokomat," in *Proc. 9th International Conference on Rehabilitation Robotics ICORR 2005*, pp. 345–348, 2005.
- [37] M. Wellner, M. Guidali, J. von Zitzewitz, and R. Riener, "Using a robotic gait orthosis as haptic display - a perception-based optimization approach," in *Proc. IEEE 10th International Conference on Rehabilitation Robotics ICORR 2007*, pp. 81–88, 2007.
- [38] M. Wellner, A. Schaufelberger, and R. Riener, "A study on sound feedback in a virtual environment for gait rehabilitation," in *Proc. Virtual Rehabilitation*, pp. 53–56, 2007.
- [39] A. Mayr, M. Kofler, E. Quirbach, H. Matzak, K. Frhlich, and L. Saltuari, "Prospective, blinded, randomized crossover study of gait rehabilitation in stroke patients using the lokomat gait orthosis.," *Neurorehabil Neural Repair*, vol. 21, no. 4, pp. 307–314, 2007.
- [40] J. M. Hidler and A. E. Wall, "Alterations in muscle activation patterns during robotic-assisted walking.," *Clin Biomech (Bristol, Avon)*, vol. 20, pp. 184–193, Feb 2005.
- [41] J. Hidler, W. Wisman, and N. Neckel, "Kinematic trajectories while walking within the lokomat robotic gait-orthosis.," *Clin Biomech (Bristol, Avon)*, vol. 23, pp. 1251–1259, Oct 2008.

- [42] J. Hidler, D. Nichols, M. Pelliccio, K. Brady, D. D. Campbell, J. H. Kahn, and T. G. Hornby, "Multicenter randomized clinical trial evaluating the effectiveness of the lokomat in subacute stroke.," *Neurorehabil Neural Repair*, vol. 23, pp. 5–13, Jan 2009.
- [43] J. F. Veneman, R. Ekkelenkamp, R. Kruidhof, F. C. V. D. Helm, and H. V. D. Kooij, "A series elastic- and bowden-cable-based actuation system for use as torque actuator in exoskeleton-type robots," *Int. J. Rob. Res.*, vol. 25, no. 3, pp. 261–281, 2006.
- [44] J. F. Veneman, R. Kruidhof, E. E. G. Hekman, R. Ekkelenkamp, E. H. F. V. Asseldonk, and H. van der Kooij, "Design and evaluation of the lopes exoskeleton robot for interactive gait rehabilitation.," *IEEE Trans Neural Syst Rehabil Eng*, vol. 15, pp. 379–386, Sep 2007.
- [45] R. Ekkelenkamp, J. Veneman, and H. van der Kooij, "Lopes: selective control of gait functions during the gait rehabilitation of cva patients," in *Proc. 9th International Conference on Rehabilitation Robotics ICORR 2005*, pp. 361–364, 2005.
- [46] R. Ekkelenkamp, P. Veltink, S. Stramigioli, and H. van der Kooij, "Evaluation of a virtual model control for the selective support of gait functions using an exoskeleton," in *Proc. IEEE 10th International Conference on Rehabilitation Robotics ICORR 2007*, pp. 693–699, 2007.
- [47] E. H. F. Van Asseldonk, R. Ekkelenkamp, J. F. Veneman, F. C. T. Van der Helm, and H. van der Kooij, "Selective control of a subtask of walking in a robotic gait trainer(lopes)," in *Proc. IEEE 10th International Conference on Rehabilitation Robotics ICORR 2007*, pp. 841–848, 2007.
- [48] F. E. Zajac, R. R. Neptune, and S. A. Kautz, "Biomechanics and muscle coordination of human walking. part i: introduction to concepts, power transfer, dynamics and simulations.," *Gait Posture*, vol. 16, pp. 215–232, Dec 2002.
- [49] F. E. Zajac, R. R. Neptune, and S. A. Kautz, "Biomechanics and muscle coordination of human walking: part ii: lessons from dynamical simulations and clinical implications.," *Gait Posture*, vol. 17, pp. 1–17, Feb 2003.
- [50] E. H. F. van Asseldonk, J. F. Veneman, R. Ekkelenkamp, J. H. Buurke, F. C. T. van der Helm, and H. van der Kooij, "The effects on kinematics and muscle activity of walking in a robotic gait trainer during zero-force control.," *IEEE Trans Neural Syst Rehabil Eng*, vol. 16, pp. 360–370, Aug 2008.

- [51] S. K. Banala, S. K. Agrawal, and J. P. Scholz, "Active leg exoskeleton (alex) for gait rehabilitation of motor-impaired patients," in *Proc. IEEE 10th International Conference on Rehabilitation Robotics ICORR 2007*, pp. 401–407, 2007.
- [52] S. K. Banala, S. K. Agrawal, S. H. Kim, and J. P. Scholz, "Novel gait adaptation and neuromotor training results using an active leg exoskeleton," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 216–225, 2010.
- [53] H. Schmidt, C. Werner, R. Bernhardt, S. Hesse, and J. Krger, "Gait rehabilitation machines based on programmable footplates.," *J Neuroeng Rehabil*, vol. 4, p. 2, 2007.
- [54] H. Schmidt, S. Hesse, and R. Bernhardt, "Safety concept for robotic gait trainers," in *Proc. 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEMBS '04*, vol. 1, pp. 2703–2706, 2004.
- [55] H. Schmidt, D. Sorowka, F. Piorko, N. Marhoul, and R. Bernhardt, "Control system for a robotic walking simulator," in *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, vol. 2, pp. 2055–2060 Vol.2, 2004.
- [56] H. Schmidt, M. Volkmar, C. Werner, I. Helmich, F. Piorko, J. Kruger, and S. Hesse, "Muscle activation patterns of healthy subjects during floor walking and stair climbing on an end-effector-based gait rehabilitation robot," in *Proc. IEEE 10th International Conference on Rehabilitation Robotics ICORR 2007*, pp. 1077–1084, 2007.
- [57] S. Hussein, H. Schmidt, S. Hesse, and J. Kruger, "Effect of different training modes on ground reaction forces during robot assisted floor walking and stair climbing," in *Proc. IEEE Int. Conf. Rehabilitation Robotics ICORR 2009*, pp. 845–850, 2009.
- [58] P. Metrailler, V. Blanchard, I. Perrin, R. Brodard, R. Frischknecht, C. Schmitt, J. Fournier, M. Bouri, and R. Clavel, "Improvement of rehabilitation possibilities with the motionmaker tm," in *Proc. First IEEE/RAS-EMBS Int. Conf. Biomedical Robotics and Biomechatronics BioRob 2006*, pp. 359–364, 2006.
- [59] Y. Allemand, Y. Stauffer, R. Clavel, and R. Brodard, "Design of a new lower extremity orthosis for overground gait training with the walktrainer," in *Proc. IEEE Int. Conf. Rehabilitation Robotics ICORR 2009*, pp. 550–555, 2009.

- [60] M. Bouri, Y. Stauffer, C. Schmitt, Y. Allemand, S. Gnemmi, R. Clavel, P. Metrailler, and R. Brodard, "The walktrainer: A robotic system for walking rehabilitation," in *Proc. IEEE International Conference on Robotics and Biomimetics ROBIO '06*, pp. 1616–1621, 2006.
- [61] Y. Stauffer, Y. Allemand, M. Bouri, J. Fournier, R. Clavel, P. Metrailler, R. Brodard, and F. Reynard, "The walktrainer—a new generation of walking reeducation device combining orthoses and muscle stimulation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 1, pp. 38–45, 2009.
- [62] J. J. Craig, *Introduction to Robotics*. Pearson Prentice Hall, 2005.
- [63] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley & Sons Inc., 2006.
- [64] L. Meirovitch, *Methods of Analytical Dynamics*. Mineola, New York: Dover Publications Inc., 1998.
- [65] A. D. Kuo and J. M. Donelan, "Dynamic principles of gait and their clinical implications," *Phys Ther*, vol. 90, pp. 157–174, Feb 2010.
- [66] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Wiley, 2005.
- [67] S. J. Piazza, "Muscle-driven forward dynamic simulations for the study of normal and pathological gait.," *J Neuroeng Rehabil*, vol. 3, p. 5, 2006.
- [68] D. G. Thelen and F. C. Anderson, "Using computed muscle control to generate forward dynamic simulations of human walking from experimental data.," *J Biomech*, vol. 39, no. 6, pp. 1107–1115, 2006.
- [69] D. A. Winter, *The Biomechanics and Motor Control of Human Gait*. University of Waterloo Press, 1991.
- [70] R. Riener and T. Edrich, "Identification of passive elastic joint moments in the lower extremities.," *J Biomech*, vol. 32, pp. 539–544, May 1999.
- [71] A. Silder, B. Whittington, B. Heiderscheit, and D. G. Thelen, "Identification of passive elastic joint moment-angle relationships in the lower extremity.," *J Biomech*, vol. 40, no. 12, pp. 2628–2635, 2007.
- [72] R. Drillis and R. Contini, "Body segment parameters," Tech. Rep. 1166 03, Office of Vocational Rehabilitation, Department of Health, Education and Welfare, 1966.

- [73] W. T. Dempster, “Space requirements of the seated operator,” Tech. Rep. WADC TR 55 159, United States Air Force, Wright Patterson Air Force Base, 1955.
- [74] C. E. Clauser, J. T. McConville, and J. W. Young, “Weight, volume, and center of mass of segments of the human body,” Tech. Rep. AMRL TR 69 70, Aerospace Medical Research Laboratory, August 1969.
- [75] M. A. McDowell, C. D. Fryar, C. L. Ogden, and K. M. Flegal, “Anthropometric reference data for children and adults: United states, 2003-2006,” National Health Statistics Report Number 10, U.S. Department of Health and Human Services, October 2008.
- [76] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. Springer-Verlag London Limited, 2009.
- [77] H. D. Taghirad and P. R. Belanger, “An experimental study on modelling and identification of harmonic drive systems,” in *Proc. 35th IEEE Decision and Control*, vol. 4, pp. 4725–4730, 1996.
- [78] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, 1989.
- [79] Danaher Motion, Radford, VA, *AKM Series Motors Selection Guide*, 2006.
- [80] Harmonic Drive Gearing, New York, *Cup Type Component Sets & Housed Units: CSF & CSG Series*, csf-csg rev 03-06 ed., 2006.
- [81] Danaher Motion, Radford, VA, *S200 High Performance Compact Brushless Servo Drives Reference Manual*, m-sm-200-01 revision c ed., 2008.
- [82] ATI Industrial Automation, Apex, NC, *Data Acquisition (DAQ) Six-Axis Force/Torque Sensor System Installation and Operation Manual*, 9610-05-1017 daq ed., October 2008.
- [83] Quanser Inc., Markham, Ontario, *Q8 Data Acquisition System User’s Guide*, version 1.0 ed., 2003.
- [84] Quanser Inc., Toronto, Canada, *Q8 High-Performance H.I.L. Control Board Product Information Sheet*, c1 - 1 - rev. d ed., 2008.
- [85] Texas Instruments, Dallas, Texas, *uA9637AC Dual Differential Line Receiver*, slls111b ed., May 1995.

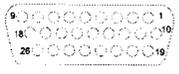
- [86] Infineon Technologies AG, *TLE4906L High Precision Hall-Effect Switch*, v 1.1 ed., October 2005.
- [87] ATI Industrial Automation, Apex, NC, *Six-Axis Force/Torque Transducer System Calibration Accuracy Report*, 9105-cal-report revision 03 ed., October 2009.

## Appendix A

# Quanser Q8 and S200 Terminal

## Definitions

Table A.1: Description of S200 J4 D-sub Command IO pin assignments in use for the rehabilitation robot

Pin Number	Description of J4 connector	 [81]
J4-1	Digital input ground	
J4-2	Enable motor digital input	
J4-3	Inhibit clockwise motion digital input	
J4-4	Inhibit counterclockwise motion digital input	
J4-5	Null analog command digital input (set CmdOffset)	
J4-17	Encoder index pulse output channel $Z$	
J4-18	Encoder index pulse output channel $\bar{Z}$	
J4-19	Encoder output channel $A$	
J4-20	Encoder output channel $\bar{A}$	
J4-21	Encoder output channel $B$	
J4-22	Encoder output channel $\bar{B}$	
J4-24	Analog command input +	
J4-25	Analog command input -	
J4-26	Analog command ground	

**Table A.2:** Q8 Data Acquisition Board Output Wiring Description

Output	Description
AO0	Analog output command for joint 1 motor base unit
AO1	Analog output command for joint 2 motor base unit
AO2	Analog output command for joint 3 motor base unit
AO3	Analog output command for joint 4 motor base unit
DO0	Digital output enabling the motor for joint 1
DO1	Digital output to inhibit clockwise motion for joint 1
DO1	Digital output to inhibit counterclockwise motion for joint 1
DO3	Digital output to null analog command offset CmdOffset for joint 1
DO4	Digital output enabling the motor for joint 2
DO5	Digital output to inhibit clockwise motion for joint 2
DO6	Digital output to inhibit counterclockwise motion for joint 2
DO7	Digital output to null analog command offset CmdOffset for joint 2
DO8	Digital output enabling the motor for joint 3
DO9	Digital output to inhibit clockwise motion for joint 3
DO10	Digital output to inhibit counterclockwise motion for joint 3
DO11	Digital output to null analog command offset CmdOffset for joint 3
DO12	Digital output enabling the motor for joint 4
DO13	Digital output to inhibit clockwise motion for joint 4
DO14	Digital output to inhibit counterclockwise motion for joint 4
DO15	Digital output to null analog command offset CmdOffset for joint 4
+5 V	Supply voltage for low power electronic components
GND	Supply ground

**Table A.3:** Q8 Data Acquisition Board Input Wiring Description

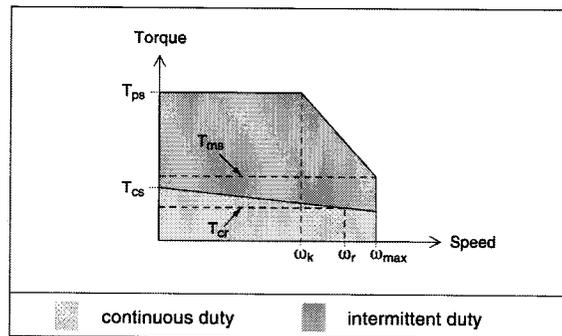
Input	Description
AI0	Analog input for FT sensor SG0
AI1	Analog input for FT sensor SG1
AI2	Analog input for FT sensor SG2
AI3	Analog input for FT sensor SG3
AI4	Analog input for FT sensor SG4
AI5	Analog input for FT sensor SG5
E0	Encoder counter input for joint 1
E1	Encoder counter input for joint 2
E2	Encoder counter input for joint 3
E3	Encoder counter input for joint 4
DI24	Digital input of limit switch for -ve motion of joint 4
DI25	Digital input of limit switch for +ve motion of joint 4
DI26	Digital input of limit switch for -ve motion of joint 3
DI27	Digital input of limit switch for +ve motion of joint 3
DI28	Digital input of limit switch for -ve motion of joint 2
DI29	Digital input of limit switch for +ve motion of joint 2
DI30	Digital input of limit switch for -ve motion of joint 1
DI31	Digital input of limit switch for +ve motion of joint 1
DI16	Digital input for emergency stop buttons
DI32	Digital input for DAQ fuse monitoring

GENERAL	
Interface	PCI
PCI Bus Width	32-bit
Bus Speed	33 MHz
ANALOG INPUTS	
Channels	8 SE
Resolution	14-bit
Input Range (fixed)	± 10V
Conversion Time & Sampling Frequency*	
1 or 2 channels simultaneous	5.2 µs / 192 kHz
All 8 channels simultaneous	17.8 µs / 56 kHz
* The values above are measurements of time to the completion time of the I/O specified. On-board hardware FIFO allows A/D conversions to be performed in the background by the hardware, thus permitting control calculations and A/D to be overlapped.	
Features	
Simultaneous Sample & Hold	
ANALOG OUTPUTS	
Channels	8
Resolution	12-bit
Slew Rate	2.5 V / µs
Output range:	
Bipolar:	± 5V, ± 10V
Unipolar:	0-10V
Conversion Time **	
1 channel	0.62 µs
2 channels simultaneous	0.62 µs
8 channels simultaneous	1.35 µs
** The values above are measurements of time to write to the D/A. On-board double-buffering allows D/A conversions to be performed in the background by the hardware, thus permitting control calculations and D/A to be overlapped.	
Features	
Individual mode and gain selectability	
Known state on power-up or reset	
Simultaneous output	
ENCODER INPUT COUNTERS	
Quadrature Encoder Inputs supported	8
Counter size	24-bit
Max. A and B Frequency in Quadrature	2 MHz
Max. Count Frequency in 4X Quadrature	8 MHz
Max. Count Frequency in Non-quadrature	15 MHz
Conversion Time	
Single Encoder Read	2.50 µs
4 Simultaneous Encoder Reads	2.50 µs
All 8 channels simultaneous	4.72 µs
Features	
Simultaneous Sampling	
Single-Ended Inputs	
Non-quadrature (count/direction) and 1X, 2X or 4X quadrature modes	
Extensive Index Pulse Signal Support	
Individually programmable count and index modes, and filter clocks	
TTL/CMOS compatible	
DIGITAL I/O	
Number of lines	32
Conversion Time	
Single Digital Output	0.18 µs
Multiple Digital Output (32 lines)	0.18 µs
Single Digital Input	0.68 µs
Multiple Digital Input (32 lines)	0.79 µs
Features	
Totem-pole outputs	
Known state on power-up or reset	
Individually Software Programmable	
Single access read/write of all channels	
GENERAL PURPOSE COUNTER-TIMERS	
Number of counter-timers	2
Counter-Timer Size	32-bit
Timer Resolution	30 ns
Features	
Configurable as PWM Outputs	
One Counter Configurable as Watchdog Timer	
One Counter Allows Hardware Gating	
Software enabling/disabling	
WATCHDOG TIMERS (reconfigured GP Counter-Timers)	
User programmable Watchdog Timers	1
Features	
Software enabling/disabling	
Automatic reset of Analog Outputs & Digital I/O	
PWM OUTPUTS (reconfigured GP Counter-Timers)	
User-programmable PWM outputs	2
EXTRA FEATURES	
Fused Encoder Section with LED status	
Fuse monitoring with watchdog and interrupt capability	
55 configurable interrupt sources	
External A/D Triggering Support	
SOFTWARE SUPPORT	
Real-Time Target Support ***	
Quanser QuaRC for Windows®, QuaRC for QNX	
The Mathworks xPC Target	
OPAL-RT RT-LAB	
National Instruments LabVIEW	
*** Software support provided by their respective vendors.	
OS Drivers	
Microsoft® Windows® XP and Windows Vista®	
QNX®	
Ardence RTX	
API	
C	
C++	
ActiveX	
.NET (VB, C#, C++ and others)	
LabVIEW	
MATLAB	
MATRIXx	

Figure A.1: Quanser Q8 specification sheet [84]

## Appendix B

# Danaher Motion AKM Motor Specifications



### Definitions

- $T_{ps}$  - Peak stall torque for system
- $T_{ms}$  - Peak torque at maximum speed
- $T_{cs}$  - Continuous torque at stall
- $T_{cr}$  - Continuous rated torque (torque at rated power)
- $\omega_{max}$  - Maximum speed
- $\omega_r$  - Rated speed (speed at rated power)
- $\omega_k$  - Speed at knee in peak envelope (intersection of system peak torque with voltage limit line)

Figure B.1: Motor torque-speed curve definitions [79]

**AKM2x - Up to 640 VDC**

See system data beginning on page 8 for typical torque/speed performance.

PARAMETER	Tol	SYMBOL	UNITS	AKM21			AKM22			AKM23			AKM24		
				C	F	G	C	E	G	D	F	G	D	F	
Max Rated DC Bus Voltage	Max	Vbus	Vdc	320	160	75	640	320	160	640	640	320	640	640	320
Continuous Torque (Stall) for $\Delta T$ winding = 100°C	Nom	Tcs	N-m lb-in	0.48 4.2	0.50 4.4	0.50 4.4	0.84 7.4	0.87 7.7	0.88 7.8	1.13 10.0	1.16 10.3	1.18 10.4	1.38 12.2	1.41 12.5	1.42 12.6
Continuous Current (Stall) for $\Delta T$ winding = 100°C	Nom	Ics	A <sub>rms</sub>	1.58	3.11	4.87	1.39	2.73	4.82	1.41	2.19	4.31	1.42	2.21	3.89
Continuous Torque (Stall) for $\Delta T$ winding = 60°C	Nom	Tcs	N-m lb-in	0.38 3.4	0.40 3.5	0.40 3.5	0.67 5.9	0.70 6.2	0.70 6.2	0.90 8.0	0.92 8.2	0.94 8.4	1.10 9.8	1.13 10.0	1.14 10.1
Max Mechanical Speed	Nom	Nmax	rpm	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000
Peak Torque	Nom	Tp	N-m lb-in	1.47 13.0	1.49 13.2	1.51 13.4	2.73 24.2	2.76 24.4	2.79 24.7	3.77 33.4	3.84 34.0	3.88 34.3	4.73 41.9	4.76 42.1	4.82 42.7
Peak Current	Nom	Ip	A <sub>rms</sub>	6.3	12.4	19.5	5.6	10.9	19.3	5.6	8.8	17.2	5.7	8.8	15.6
75VDC	Rated Torque (speed)	T <sub>rtid</sub>	N-m lb-in	-	0.48 4.2	0.46 4.1	-	0.85 7.5	0.83 7.4	-	-	1.15 10.2	-	-	1.39 12.3
	Rated Speed	N <sub>rtid</sub>	rpm	-	2000	4000	-	1000	2500	-	-	1500	-	-	1000
	Rated Power (speed)	P <sub>rtid</sub>	kW Hp	-	0.10 0.13	0.19 0.26	-	0.09 0.12	0.22 0.29	-	-	0.18 0.24	-	-	0.15 0.20
160VDC	Rated Torque (speed)	T <sub>rtid</sub>	N-m lb-in	0.46 4.0	0.41 3.7	-	0.83 7.3	0.81 7.1	0.74 6.5	1.11 9.8	1.12 9.9	1.07 9.5	-	1.36 12.0	1.33 11.8
	Rated Speed	N <sub>rtid</sub>	rpm	2500	7000	-	1000	3500	7000	1000	1500	4500	-	1500	3000
	Rated Power (speed)	P <sub>rtid</sub>	kW Hp	0.12 0.16	0.30 0.41	-	0.09 0.12	0.30 0.40	0.54 0.72	0.12 0.16	0.18 0.24	0.50 0.68	-	0.21 0.29	0.42 0.56
320VDC	Rated Torque (speed)	T <sub>rtid</sub>	N-m lb-in	0.39 3.4	-	-	0.78 6.9	0.70 6.2	-	1.08 9.6	1.03 9.1	0.94 8.3	1.32 11.7	1.29 11.4	1.12 9.9
	Rated Speed	N <sub>rtid</sub>	rpm	8000	-	-	3500	8000	-	2500	5000	8000	2000	4000	8000
	Rated Power (speed)	P <sub>rtid</sub>	kW Hp	0.32 0.43	-	-	0.29 0.38	0.59 0.79	-	0.28 0.38	0.54 0.72	0.79 1.06	0.28 0.37	0.54 0.72	0.94 1.26
560VDC	Rated Torque (speed)	T <sub>rtid</sub>	N-m lb-in	-	-	-	0.68 6.0	-	-	0.99 8.8	0.92 8.1	-	1.25 11.1	1.11 9.8	-
	Rated Speed	N <sub>rtid</sub>	rpm	-	-	-	8000	-	-	5500	8000	-	4500	8000	-
	Rated Power (speed)	P <sub>rtid</sub>	kW Hp	-	-	-	0.57 0.76	-	-	0.57 0.76	0.77 1.03	-	0.59 0.79	0.93 1.25	-
640VDC	Rated Torque (speed)	T <sub>rtid</sub>	N-m lb-in	-	-	-	0.68 6.0	-	-	0.95 8.4	0.92 8.1	-	1.22 10.8	1.11 9.8	-
	Rated Speed	N <sub>rtid</sub>	rpm	-	-	-	8000	-	-	7000	8000	-	5500	8000	-
	Rated Power (speed)	P <sub>rtid</sub>	kW Hp	-	-	-	0.57 0.76	-	-	0.70 0.93	0.77 1.03	-	0.70 0.94	0.93 1.25	-
Torque Constant	±10%	K <sub>t</sub>	N-m/A <sub>rms</sub> lb-in/A <sub>rms</sub>	0.30 2.7	0.16 1.4	0.10 0.9	0.61 5.4	0.32 2.8	0.18 1.6	0.80 7.1	0.52 4.6	0.27 2.4	0.97 8.6	0.63 5.6	0.36 3.2
Back EMF constant	±10%	K <sub>v</sub>	V/krpm	19.5	10.2	6.6	39	20.4	11.7	51.8	33.8	17.6	62.4	40.8	23.4
Resistance (line-line)	±10%	R <sub>m</sub>	mΩ	13.0	3.42	1.44	19.4	5.09	1.69	20.3	8.36	2.23	20.4	8.4	2.77
Inductance (line-line)		L	mH	19	5.2	2.18	35.5	9.7	3.19	40.7	17.3	4.68	43.8	18.7	6.16
Inertia (includes Resolver feedback)		J <sub>m</sub>	kg-cm <sup>2</sup> lb-in-s <sup>2</sup>		0.11 9.5E-05			0.16 1.4E-04			0.22 1.9E-04		0.27 2.4E-04		
Optional Brake Inertia (additional)		J <sub>m</sub>	kg-cm <sup>2</sup> lb-in-s <sup>2</sup>		0.012 1.1E-05			0.012 1.1E-05			0.012 1.1E-05		0.012 1.1E-05		
Weight		W	kg lb		0.82 1.8			1.1 2.4			1.38 3.0		1.66 3.7		
Static Friction		T <sub>f</sub>	N-m lb-in		0.002 0.02			0.005 0.04			0.007 0.06		0.01 0.09		
Viscous Damping		K <sub>dv</sub>	N-m/krpm lb-in/krpm		0.0046 0.04			0.0055 0.05			0.0055 0.06		0.0074 0.07		
Thermal Time Constant		TCT	minutes		8			9			10		11		
Thermal Resistance		R <sub>thw-a</sub>	°C/W		1.48			1.28			1.19		1.17		
Pole Pairs					3			3			3		3		
Heatsink Size					10x10x1/4" Aluminum Plate			10x10x1/4" Aluminum Plate			10x10x1/4" Aluminum Plate		10x10x1/4" Aluminum Plate		

Notes:

1. Motor winding temperature rise,  $\Delta T=100^\circ\text{C}$ , at  $40^\circ\text{C}$  ambient.
2. All data referenced to sinusoidal commutation.
3. Add parking brake if applicable for total inertia.
4. Motor with standard heatsink.
5. May be limited at some values of Vbus.
6. Measured at  $25^\circ\text{C}$ .
7. Brake motor option reduces continuous torque ratings by:  
AKM21 = 0.00      AKM22 = 0.01 N-m  
AKM23 = 0.02 N-m      AKM24 = 0.05 N-m
8. For non-Resolver feedback options:  
no continuous torque reduction.
9. Motors with non-Resolver feedback and Brake option, reduce continuous torque by:  
AKM21 = 0.00      AKM22 = 0.02 N-m  
AKM23 = 0.05 N-m      AKM24 = 0.12 N-m
10. For motors with optional shaft seal, reduce torque shown by 0.047 N-m (0.41lb-in), and increase T<sub>f</sub> by the same amount.

Figure B.2: AKM2x motor specifications [79]

**AKM4x - Up to 640 VDC**

See system data beginning on page B for typical torque/speed performance.

PARAMETER	Tol	SYMBOL	UNITS	AKM41			AKM42				AKM43			AKM44		
				C	E	H	C	E	G	J	E	G	K	E	G	J
Max Rated DC Bus Voltage	Max	Vbus	Vdc	640	640	320	640	640	640	320	640	640	320	640	640	640
Continuous Torque (Stall) for ΔT winding = 100°C	Nom	Tcs	N-m lb-in	1.95 17.3	2.02 17.9	2.06 18.2	3.35 29.6	3.42 30.3	3.53 31.2	3.56 31.5	4.70 41.6	4.80 42.5	4.90 43.4	5.76 51.0	5.88 52.0	6.00 53.1
Continuous Current (Stall) for ΔT winding = 100°C	Nom	Ics	A rms	1.46	2.85	5.60	1.40	2.74	4.80	8.40	2.78	4.87	8.60	2.9	5.0	8.8
Continuous Torque (Stall) for ΔT winding = 60°C	Nom	Tcs	N-m lb-in	1.56 13.8	1.62 14.3	1.65 14.6	2.68 23.7	2.74 24.2	2.82 25.0	2.85 25.2	3.76 33.3	3.84 34.0	3.92 34.7	4.61 40.8	4.70 41.6	4.80 42.5
Max Mechanical Speed	Nom	Nmax	rpm	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
Peak Torque	Nom	Tp	N-m lb-in	6.12 54.2	6.28 55.6	6.36 56.3	11.1 98.8	11.3 99.7	11.5 102	11.6 103	15.9 141	16.1 142	16.3 144	19.9 176	20.2 179	20.4 181
Peak Current	Nom	Ip	A rms	5.8	11.4	22.4	5.61	11.0	19.2	33.7	11.0	19.5	38.3	11.4	20.0	35.2
Rated Torque (speed)		Trd	N-m lb-in	-	-	1.99 17.6	-	-	-	-	-	-	-	-	-	-
Rated Speed		Nrd	rpm	-	-	1000	-	-	-	-	-	-	-	-	-	-
Rated Power (speed)		Prd	KW Hp	-	-	0.21 0.28	-	-	-	-	-	-	-	-	-	-
Rated Torque (speed)		Trd	N-m lb-in	-	1.94 17.2	1.86 16.5	-	-	-	3.03 26.8	-	-	4.08 36.1	-	-	-
Rated Speed		Nrd	rpm	-	1200	3000	-	-	3000	-	-	2500	-	-	-	-
Rated Power (speed)		Prd	KW Hp	-	0.24 0.33	0.58 0.78	-	-	0.95 1.28	-	-	1.07 1.43	-	-	-	-
Rated Torque (speed)		Trd	N-m lb-in	1.88 16.6	1.82 16.1	1.62 14.3	-	3.12 27.6	2.90 25.7	2.38 21.1	4.24 37.5	4.00 35.4	2.62 23.2	5.22 46.2	4.90 43.4	3.84 34.0
Rated Speed		Nrd	rpm	1200	3000	6000	-	1800	3500	6000	1500	2500	6000	1200	2000	4000
Rated Power (speed)		Prd	KW Hp	0.24 0.32	0.57 0.77	1.02 1.36	-	0.59 0.79	1.06 1.42	1.50 2.00	0.67 0.89	1.05 1.40	1.65 2.21	0.66 0.88	1.03 1.38	1.61 2.16
Rated Torque (speed)		Trd	N-m lb-in	1.77 15.7	1.58 14.0	-	3.10 27.4	2.81 24.9	2.35 20.8	-	3.92 34.7	3.01 26.6	-	4.80 42.5	3.76 33.3	2.75 24.3
Rated Speed		Nrd	rpm	3000	6000	-	1500	3500	6000	-	2500	5000	-	2000	4000	6000
Rated Power (speed)		Prd	KW Hp	0.56 0.75	0.99 1.33	-	0.49 0.65	1.03 1.38	1.48 1.98	-	1.03 1.38	1.58 2.11	-	1.01 1.35	1.57 2.11	1.73 2.32
Rated Torque (speed)		Trd	N-m lb-in	1.74 15.4	1.58 14.0	-	3.02 26.7	2.72 24.1	2.35 20.8	-	3.76 33.3	2.57 22.7	-	4.56 40.4	3.19 28.2	2.75 24.3
Rated Speed		Nrd	rpm	3500	6000	-	2000	4000	6000	-	3000	6000	-	2500	5000	6000
Rated Power (speed)		Prd	KW Hp	0.64 0.85	0.99 1.33	-	0.63 0.85	1.14 1.53	1.48 1.98	-	1.18 1.58	1.61 2.16	-	1.19 1.60	1.67 2.24	1.73 2.32
Torque Constant	±10%	Kt	N-m/A rms lb-in/A rms	1.34 11.9	0.71 6.3	0.37 3.3	2.40 21.2	1.26 11.2	0.74 6.5	0.43 3.8	1.72 15.2	0.99 8.8	0.52 4.6	2.04 18.1	1.19 10.5	0.69 6.1
Back EMF constant	±10%	Kb	V/A rms	86.3	45.6	23.7	154	80.9	47.5	27.5	111	63.9	33.2	132	76.6	44.2
Resistance (line-line)	±10%	Rm	Ω	21.7	5.7	1.51	27.52	7.22	2.38	0.80	8.04	2.61	0.70	8.08	2.65	0.88
Inductance (line-line)		L	mH	66.1	18.4	5.0	97.4	26.8	9.2	3.1	32.6	10.8	2.9	33.9	11.5	3.8
Inertia		Jm	kg-cm² lb-in²	0.81				1.5				2.1			2.7	
(includes Resolver feedback)		Jm	kg-cm² lb-in²		7.2E-04			1.3E-03				1.8E-03			2.4E-03	
Optional Brake Inertia		Jm	kg-cm² lb-in²		0.068			0.068				0.068			0.068	
(additional)		Jm	kg-cm² lb-in²		6.0E-05			6.0E-05				6.0E-05			6.0E-05	
Weight		W	kg lb	2.44 5.4				3.39 7.5			4.35 9.6			5.3 11.7		
Static Friction		Tf	N-m lb-in		0.014 0.12			0.026 0.23			0.038 0.34			0.05 0.44		
Viscous Damping		Kdv	N-m/krpm lb-in/krpm		0.009 0.08			0.013 0.12			0.017 0.15			0.021 0.19		
Thermal Time Constant		TCT	minutes		13			17			20			24		
Thermal Resistance		Rthw-a	°C/W		1.04			0.89			0.78			0.71		
Pole Pairs					5			5			5			5		
Heatsink Size				10x10x1/4" Aluminum Plate			10x10x1/4" Aluminum Plate				10x10x1/4" Aluminum Plate			10x10x1/4" Aluminum Plate		

- Notes:
- Motor winding temperature rise, ΔT=100°C, at 40°C ambient.
  - All data referenced to sinusoidal commutation.
  - Add parking brake if applicable for total inertia.
  - Motor with standard heatsink.
  - May be limited at some values of Vbus.
  - Measured at 25°C.
  - Brake motor option reduces continuous torque ratings by 0.12 N-m.
  - Non-Resolver feedback option reduces continuous ratings by:  
AKM41 = 0.1 N-m  
AKM43 = 0.2 N-m  
AKM42 = 0.1 N-m  
AKM44 = 0.3 N-m
  - Motors with non-Resolver feedback and Brake option, reduce continuous torque by:  
AKM41 = 0.22 N-m  
AKM43 = 0.55 N-m  
AKM42 = 0.36 N-m  
AKM44 = 0.76 N-m
  - For motors with optional shaft seal, reduce torque shown by 0.071 N-m (0.63lb-in), and increase Tf by the same amount.

Figure B.3: AKM4x motor specifications [79]

# Appendix C

## Harmonic Drive Unit Sizing Specifications

CSF Rating Table

Table 1

Size	Ratio	Rated Torque at 2000 T <sub>r</sub> rpm		Limit for Repeated Peak Torque		Limit for Average Torque		Limit for Momentary Peak Torque		Maximum Input Speed		Limit for Average Input Speed		Moment of Inertia	
		Nm	in-lb	Nm	in-lb	Nm	in-lb	Nm	in-lb	Oil	Grease	Oil	Grease	x10 <sup>-4</sup> kg·m <sup>2</sup>	x10 <sup>-6</sup> kgf·m·s <sup>2</sup>
		rpm		rpm		rpm		rpm		rpm		rpm			
32	30	54	478	100	885	75	664	200	1770						
	50	76	673	216	1912	108	956	382	3381						
	80	118	1044	304	2690	167	1478	568	5027	7000	4800	4600	3500	1.69	1.72
	100	137	1212	353	2947	216	1912	647	5726						
	120	137	1212	353	3124	216	1912	686	6071						
40	160	137	1212	372	3292	216	1912	686	6071						
	50	137	1212	402	3558	196	1735	686	6071						
	80	208	1823	619	4593	284	2513	980	8673						
	100	265	2345	568	5027	372	3292	1080	9558	5600	4000	3600	3000	4.50	4.59
	120	294	2602	617	5460	451	3991	1180	10443						
45	160	294	2602	647	5726	451	3991	1180	10443						
	50	176	1558	500	4425	265	2345	950	8408						
	80	313	2770	706	6248	390	3452	1270	11240						
	100	353	3124	755	6682	500	4425	1570	13895	5000	3800	3300	3000	6.68	6.86
	120	402	3558	823	7284	620	5487	1760	15576						
50	160	402	3558	862	7806	630	5576	1910	16904						
	50	245	2168	715	6328	350	3098	1430	12656						
	80	372	3292	941	8328	519	4593	1860	16461						
	100	470	4160	980	8673	666	5894	2060	18231	4500	3500	3000	2500	12.5	12.8
	120	529	4682	1080	9558	813	7196	2060	18231						
58	160	529	4682	1180	10443	843	7461	2450	21683						
	50	353	3124	1020	9027	520	4602	1960	17346						
	80	549	4859	1480	13098	770	6815	2450	21683						
	100	696	6160	1590	14072	1060	9381	3180	28143	4000	3000	2700	2200	27.3	27.9
	120	745	6593	1720	15222	1190	10532	3330	29471						
160	745	6593	1840	16284	1210	10709	3430	30356							

Figure C.1: CSF Harmonic Drive Unit Sizing Specifications [80]

Rating Table

Table 2

Size	Ratio	Rated Torque at 2000 Tr rpm		Limit for Repeated Peak Torque		Limit for Average Torque		Limit for Momentary Peak Torque		Maximum Input Speed rpm		Limit for Average Input Speed rpm		Moment of Inertia	
		Nm	in-lb	Nm	in-lb	Nm	in-lb	Nm	in-lb	Oil	Grease	Oil	Grease	x10 <sup>-4</sup> kg-m <sup>2</sup>	x10 <sup>-6</sup> kgf-m-s <sup>2</sup>
65	50	490	4337	1420	12567	720	6372	2830	25046	3500	2800	2400	1900	46.8	47.8
	80	745	6583	2110	187	1040	9204	3720	32922						
	100	951	8416	2300	20355	1520	13452	4750	42038						
	120	951	8416	2510	22214	1570	13895	4750	42038						
	160	951	8416	2630	23276	1570	13895	4750	42038						
80	50	872	7717	2440	21594	1260	11151	4870	43100	2900	2300	2200	1600	122	124
	80	1320	11682	3430	30356	1830	16196	6590	58322						
	100	1700	15045	4220	37347	2360	20886	7910	70004						
	120	1990	17612	4590	40622	3130	27701	7910	70004						
	160	1990	17612	4910	43454	3130	27701	7910	70004						
90	50	1180	10443	3530	31241	1720	15222	6660	58941	2700	2000	2100	1300	214	218
	80	1550	13718	3990	35312	2510	22214	7250	64163						
	100	2270	20090	5680	50269	3360	29736	9020	79827						
	120	2570	22745	6160	54516	4300	38055	9800	86730						
	160	2700	23895	6840	60534	4300	38055	11300	100005						
100	50	1580	13983	4450	39383	2280	20178	8900	78765	2500	1800	2000	1200	356	363
	80	2380	21063	6060	53631	3310	29294	11600	102660						
	100	2940	26019	7350	65048	4630	40976	14100	124785						
	120	3180	28143	7960	70446	5720	50622	15300	135405						
	160	3550	31418	9180	81243	5720	50622	15500	137175						

CSG Rating Table

Table 3

Size	Ratio	Rated Torque at 2000 Tr rpm		Limit for Repeated Peak Torque		Limit for Average Torque		Limit for Momentary Peak Torque		Maximum Input Speed rpm		Limit for Average Input Speed rpm		Moment of Inertia	
		Nm	in-lb	Nm	in-lb	Nm	in-lb	Nm	in-lb	Oil	Grease	Oil	Grease	x10 <sup>-4</sup> kg-m <sup>2</sup>	x10 <sup>-6</sup> kgf-m-s <sup>2</sup>
14	50	7.0	62	23	204	9	80	46	407	14000	8500	6500	3500	0.033	0.0034
	80	10	89	30	266	14	124	61	540						
	100	10	89	36	319	14	124	70	620						
17	50	21	186	44	390	34	301	91	805	10000	7300	6500	3500	0.079	0.081
	80	29	257	56	496	35	310	113	1000						
	100	31	274	70	620	51	451	143	1266						
	120	31	274	70	620	51	451	112	991						
	160	33	292	73	646	44	389	127	1124						
20	50	44	389	96	850	61	540	165	1460	10000	6500	6500	3500	0.193	0.197
	80	52	460	107	947	64	566	191	1690						
	100	52	460	113	1000	64	566	191	1690						
	120	52	460	120	1062	64	566	191	1690						
	160	51	451	127	1124	72	637	242	2142						
25	50	82	726	178	1575	113	1000	332	2938	7500	5600	5600	3500	0.413	0.421
	80	87	770	204	1805	140	1239	369	3266						
	100	87	770	217	1920	140	1239	395	3496						
	120	87	770	229	2027	140	1239	408	3611						
	160	99	876	281	2487	140	1239	497	4399						
32	50	153	1354	395	3496	217	1920	738	6531	7000	4800	4600	3500	1.69	1.72
	80	178	1575	433	3832	281	2487	841	7443						
	100	178	1575	459	4062	281	2487	892	7894						
	120	178	1575	484	4283	281	2487	892	7894						
	160	178	1575	523	4629	255	2257	892	7894						
40	50	268	2372	675	5974	389	3266	1270	11240	5600	4000	3600	5000	4.50	4.59
	80	345	3053	738	6531	484	4283	1400	12390						
	100	382	3381	802	7098	586	5186	1530	13541						
	120	382	3381	841	7443	586	5186	1530	13541						
	160	382	3381	841	7443	586	5186	1530	13541						

Figure C.2: CSG Harmonic Drive Unit Sizing Specifications [80]

## Appendix D

# Hall Effect Proximity Sensor

## Specifications

Parameter	Symbol	Limit Values			Unit	Conditions
		min.	typ.	max.		
Supply voltage	$V_S$	2.7	–	18	V	
Output voltage	$V_Q$	– 0.7	–	18	V	
Junction temperature	$T_j$	– 40	–	150	°C	
Output current	$I_Q$	0	–	20	mA	

Figure D.1: Hall effect sensor operating range [86]

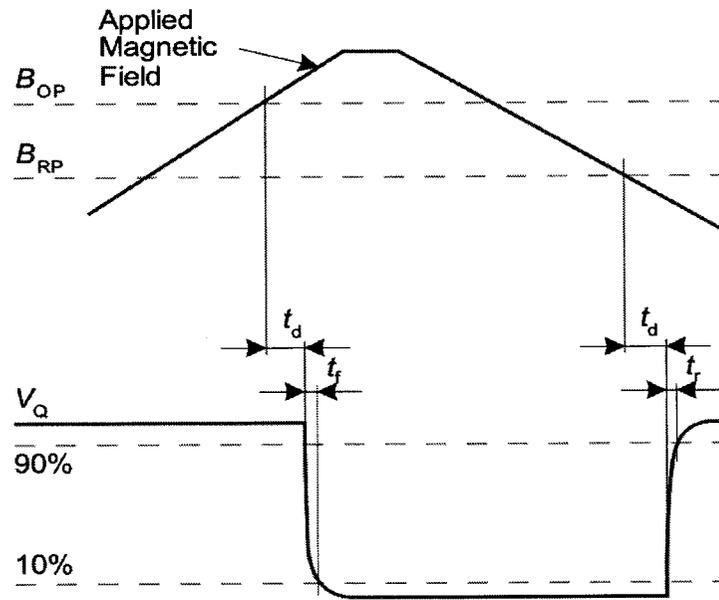


Figure D.2: Hall effect sensor timing diagram [86]

Parameter	Symbol	Limit Values			Unit	Conditions
		min.	typ.	max.		
Supply current	$I_S$	2	4	6	mA	$V_S = 2.7 \text{ V} \dots 18 \text{ V}$
Reverse current	$I_{SR}$	0	0.2	1	mA	$V_S = -18 \text{ V}$
Output saturation voltage	$V_{QSAT}$	–	0.3	0.6	V	$I_Q = 20 \text{ mA}$
Output leakage current	$I_{QLEAK}$	–	0.05	10	$\mu\text{A}$	for $V_Q = 18 \text{ V}$
Output fall time	$t_f$	–	0.02	1	$\mu\text{s}$	$R_L = 1.2 \text{ k}\Omega$ ; $C_L = 50 \text{ pF}$ see: Figure 3 “Timing Definition” on Page 12
Output rise time	$t_r$	–	0.4	1	$\mu\text{s}$	
Chopper frequency	$f_{OSC}$	–	320	–	kHz	
Switching frequency	$f_{SW}$	0	–	15 <sup>2)</sup>	kHz	
Delay time <sup>3)</sup>	$t_d$	–	13	–	$\mu\text{s}$	
Output jitter <sup>4)</sup>	$t_{QJ}$	–	1	–	$\mu\text{s}_{RMS}$	Typical value for square wave signal 1 kHz
Power-on time <sup>5)</sup>	$t_{PON}$	–	13	–	$\mu\text{s}$	$V_S \geq 2.7 \text{ V}$
Thermal resistance <sup>6)</sup>	$R_{thJA}$	–	100	–	K/W	SC59
		–	–	190		PG-SSO-3-2

1) over operating range, unless otherwise specified. Typical values correspond to  $V_S = 12 \text{ V}$  and  $T_A = 25^\circ\text{C}$

2) To operate the sensor at the max. switching frequency, the value of the magnetic signal amplitude must be 1.4 times higher than for static fields.

This is due to the -3 dB corner frequency of the low pass filter in the signal path.

3) Systematic delay between magnetic threshold reached and output switching

4) Jitter is the unpredictable deviation of the output switching delay

5) Time from applying  $V_S \geq 2.7 \text{ V}$  to the sensor until the output state is valid

6) Thermal resistance from junction to ambient

Figure D.3: Hall effect sensor electrical characteristics [86]

Parameter	Symbol	$T_j$ [ $^\circ\text{C}$ ]	Limit Values			Unit	Notes
			min.	typ.	max.		
Operate point	$B_{OP}$	– 40	6.7	10.3	13.9	mT	
		25	6.5	10.0	13.5		
		150	6.2	9.5	12.9		
Release point	$B_{RP}$	– 40	5.2	8.7	12.3	mT	
		25	5.0	8.5	12.0		
		150	4.7	8.1	11.4		

Figure D.4: Hall effect sensor magnetic characteristics [86]



## Appendix E

# Mini85 6-axis Force/Torque Sensor Calibration Accuracy Report

Full-Scale Loads						
Fx	Fy	Fz	Tx	Ty	Tz	
1900	1900	3800	80	80	80	

Applied Loads						
	Fx	Fy	Fz	Tx	Ty	Tz
1	0.000	556.028	0.000	-63.543	0.000	0.000
2	-556.028	0.000	0.000	0.000	-63.543	0.000
3	0.000	-556.028	0.000	63.543	0.000	0.000
4	556.028	0.000	0.000	0.000	63.543	0.000
5	0.000	889.644	0.000	-33.886	0.000	0.000
6	-889.644	0.000	0.000	0.000	-33.886	0.000
7	0.000	-889.644	0.000	33.886	0.000	0.000
8	889.644	0.000	0.000	0.000	33.886	0.000
9	0.000	622.751	0.000	-7.909	0.000	-71.318
10	0.000	622.751	0.000	-7.909	0.000	71.350
11	-622.751	0.000	0.000	0.000	-7.909	-71.296
12	-622.751	0.000	0.000	0.000	-7.909	71.312
13	0.000	-622.751	0.000	7.909	0.000	-71.350
14	0.000	-622.751	0.000	7.909	0.000	71.318
15	622.751	0.000	0.000	0.000	7.909	-71.312
16	622.751	0.000	0.000	0.000	7.909	71.296
17	0.000	0.000	889.644	-67.791	0.000	0.000
18	0.000	0.000	889.644	0.000	-67.791	0.000
19	0.000	0.000	889.644	67.791	0.000	0.000
20	0.000	0.000	889.644	0.000	67.791	0.000
21	0.000	0.000	1779.289	0.000	0.000	0.000
22	0.000	0.000	-1779.289	0.000	0.000	0.000
23	0.000	0.000	-889.644	67.791	0.000	0.000
24	0.000	0.000	-889.644	0.000	67.791	0.000
25	0.000	0.000	-889.644	-67.791	0.000	0.000
26	0.000	0.000	-889.644	0.000	-67.791	0.000

*Refer to page 1 for important information on regarding this report.*

Figure E.1: Mini85 SI-1900-80 list of applied loads during calibration [87]

Full-Scale Error						
	Fx	Fy	Fz	Tx	Ty	Tz
1	-0.43%	0.10%	-0.04%	0.20%	0.49%	-0.01%
2	0.18%	0.13%	-0.03%	0.11%	-0.37%	0.06%
3	-0.36%	-0.12%	-0.03%	-0.07%	0.61%	0.02%
4	0.30%	0.20%	-0.08%	0.18%	-0.65%	0.19%
5	-0.30%	0.21%	0.01%	0.11%	0.40%	0.00%
6	-0.12%	0.02%	0.01%	0.08%	-0.22%	0.12%
7	-0.27%	-0.23%	0.02%	0.00%	0.35%	0.23%
8	0.39%	0.08%	0.02%	0.05%	-0.22%	0.05%
9	-0.01%	-0.28%	0.02%	-0.18%	0.16%	0.37%
10	-0.18%	-0.11%	0.02%	-0.06%	0.06%	0.03%
11	0.32%	0.21%	0.01%	-0.02%	-0.20%	-0.06%
12	0.17%	-0.11%	0.01%	0.04%	-0.13%	0.10%
13	-0.32%	0.17%	0.01%	0.03%	0.04%	-0.04%
14	0.03%	0.29%	0.00%	0.13%	0.15%	0.07%
15	-0.18%	-0.05%	0.02%	0.13%	0.04%	-0.02%
16	-0.21%	-0.02%	0.02%	-0.14%	-0.04%	0.04%
17	-0.33%	0.08%	-0.12%	0.43%	0.70%	0.06%
18	0.09%	0.11%	-0.15%	0.17%	-0.21%	0.11%
19	-0.22%	0.09%	-0.12%	-0.01%	0.59%	-0.06%
20	0.13%	0.03%	-0.11%	0.08%	-0.69%	0.01%
21	0.01%	0.08%	0.07%	0.01%	-0.05%	-0.08%
22	-0.09%	0.03%	-0.11%	0.05%	0.10%	-0.02%
23	-0.39%	0.22%	-0.03%	0.44%	0.40%	-0.01%
24	0.11%	0.13%	-0.03%	0.23%	-0.01%	0.03%
25	-0.22%	-0.09%	-0.03%	-0.25%	0.38%	-0.02%
26	0.38%	0.14%	-0.03%	0.19%	-0.68%	0.01%

Figure E.2: Mini85 SI-1900-80 list of percent errors for applied loads during calibration [87]

## Appendix F

# ViGRR Regressor Matrices

The general ViGRR dynamics regressor matrix is

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}, \quad (\text{F.1})$$

where

$$\begin{aligned}
Y_1 = & [(\ddot{q}_1), \\
& (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3), \\
& (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4), \\
& (\ddot{q}_1 + \ddot{q}_2), \\
& c_{1234}, \\
& (2\ddot{q}_1c_3 + 2\ddot{q}_2c_3 + \ddot{q}_3c_3 - \dot{q}_3\dot{q}_1s_3 - \dot{q}_3\dot{q}_2s_3 - \dot{q}_3s_3(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)), \\
& c_{12}, \\
& v(2\ddot{q}_1c_4 + 2\ddot{q}_2c_4 + 2\ddot{q}_3c_4 + \ddot{q}_4c_4 - \dot{q}_4\dot{q}_1s_4 - \dot{q}_4\dot{q}_2s_4 - \dot{q}_4\dot{q}_3s_4 \\
& \quad - \dot{q}_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& c_1, \\
& c_{123}, \\
& (2\ddot{q}_1c_{234} + \ddot{q}_2c_{234} + \ddot{q}_3c_{234} + \ddot{q}_4c_{234} - \dot{q}_1s_{234}(\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - \dot{q}_2s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - \dot{q}_3s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - \dot{q}_4s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& (2\ddot{q}_1c_{34} + 2\ddot{q}_2c_{34} + \ddot{q}_3c_{34} + \ddot{q}_4c_{34} - \dot{q}_3s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - \dot{q}_4s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - \dot{q}_1s_{34}(\dot{q}_3 + \dot{q}_4) - \dot{q}_2s_{34}(\dot{q}_3 + \dot{q}_4)), \\
& (2\ddot{q}_1c_2 + \ddot{q}_2c_2 - \dot{q}_2\dot{q}_1s_2 - \dot{q}_2s_2(\dot{q}_1 + \dot{q}_2)), \\
& (2\ddot{q}_1c_{23} + \ddot{q}_2c_{23} + \ddot{q}_3c_{23} - \dot{q}_2s_{23}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\
& \quad - \dot{q}_3s_{23}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) - \dot{q}_1s_{23}(\dot{q}_2 + \dot{q}_3)), \\
& \dot{q}_1, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_1\dot{q}_1), \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0]
\end{aligned}$$

$$\begin{aligned}
Y_2 = & [0, \\
& (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3), \\
& (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4), \\
& (\ddot{q}_1 + \ddot{q}_2), \\
& c_{1234}, \\
& (2\ddot{q}_1 c_3 + 2\ddot{q}_2 c_3 + \ddot{q}_3 c_3 - \dot{q}_3 \dot{q}_1 s_3 - \dot{q}_3 \dot{q}_2 s_3 - \dot{q}_3 s_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)), \\
& c_{12}, \\
& (2\ddot{q}_1 c_4 + 2\ddot{q}_2 c_4 + 2\ddot{q}_3 c_4 + \ddot{q}_4 c_4 - \dot{q}_4 \dot{q}_1 s_4 - \dot{q}_4 \dot{q}_2 s_4 \\
& \quad - \dot{q}_4 \dot{q}_3 s_4 - \dot{q}_4 s_4 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& 0, \\
& c_{123}, \\
& (\ddot{q}_1 c_{234} + \dot{q}_1 \dot{q}_1 s_{234}), \\
& (2\ddot{q}_1 c_{34} + 2\ddot{q}_2 c_{34} + \ddot{q}_3 c_{34} + \ddot{q}_4 c_{34} - \dot{q}_3 s_{34} (\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - \dot{q}_4 s_{34} (\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - \dot{q}_1 s_{34} (\dot{q}_3 + \dot{q}_4) - \dot{q}_2 s_{34} (\dot{q}_3 + \dot{q}_4)), \\
& (\ddot{q}_1 c_2 + \dot{q}_1 \dot{q}_1 s_2), \\
& (\ddot{q}_1 c_{23} + \dot{q}_1 \dot{q}_1 s_{23}), \\
& 0, \\
& \dot{q}_2, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_2 \dot{q}_2), \\
& 0, \\
& 0, \\
& \ddot{q}_2, \\
& 0, \\
& 0]
\end{aligned}$$

$$\begin{aligned}
Y_3 = & [0, \\
& (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3), \\
& (\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4), \\
& 0, \\
& c_{1234}, \\
& (\ddot{q}_1 c_3 + \ddot{q}_2 c_3 + \dot{q}_1 s_3(\dot{q}_1 + \dot{q}_2) + \dot{q}_2 s_3(\dot{q}_1 + \dot{q}_2)), \\
& 0, \\
& (2\ddot{q}_1 c_4 + 2\ddot{q}_2 c_4 + 2\ddot{q}_3 c_4 + \ddot{q}_4 c_4 - \dot{q}_4 \dot{q}_1 s_4 - \dot{q}_4 \dot{q}_2 s_4, \\
& \quad - \dot{q}_4 \dot{q}_3 s_4 - \dot{q}_4 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& 0, \\
& c_{123}, \\
& (\ddot{q}_1 c_{234} + \dot{q}_1 \dot{q}_1 s_{234}), \\
& (\ddot{q}_1 c_{34} + \ddot{q}_2 c_{34} + \dot{q}_1 s_{34}(\dot{q}_1 + \dot{q}_2) + \dot{q}_2 s_{34}(\dot{q}_1 + \dot{q}_2)), \\
& 0, \\
& (\ddot{q}_1 c_{23} + \dot{q}_1 \dot{q}_1 s_{23}), \\
& 0, \\
& 0, \\
& \dot{q}_3, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_3 \dot{q}_3), \\
& 0, \\
& 0, \\
& \ddot{q}_3, \\
& 0]
\end{aligned}$$

$$\begin{aligned}
Y_4 = & [0, \\
& 0, \\
& (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3 + \ddot{q}_4), \\
& 0, \\
& c_{1234}, \\
& 0, \\
& 0, \\
& (\dot{q}_1 c_4 + \dot{q}_2 c_4 + \dot{q}_3 c_4 + \dot{q}_1 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\
& \quad + \dot{q}_2 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) + \dot{q}_3 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)), \\
& 0, \\
& 0, \\
& (\ddot{q}_1 c_{234} + \dot{q}_1 \dot{q}_1 s_{234}), \\
& (\ddot{q}_1 c_{34} + \ddot{q}_2 c_{34} + \dot{q}_1 s_{34}(\dot{q}_1 + \dot{q}_2) + \dot{q}_2 s_{34}(\dot{q}_1 + \dot{q}_2)), \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& \dot{q}_4, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_4 \dot{q}_4), \\
& 0, \\
& 0, \\
& \ddot{q}_4]
\end{aligned}$$

The ViGRR regressor matrix for nonlinear control is

$$Y(q, \dot{q}, a, v) = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} \quad (\text{F.2})$$

$$\begin{aligned}
Y_1 = & [(a_1), \\
& (a_1 + a_2 + a_3), \\
& (a_1 + a_2 + a_3 + a_4), \\
& (a_1 + a_2), \\
& c_{1234}, \\
& (2a_1c_3 + 2a_2c_3 + a_3c_3 - \dot{q}_3v_1s_3 - \dot{q}_3v_2s_3 - v_3s_3(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)), \\
& c_{12}, \\
& v(2a_1c_4 + 2a_2c_4 + 2a_3c_4 + a_4c_4 - \dot{q}_4v_1s_4 - \dot{q}_4v_2s_4 - \dot{q}_4v_3s_4 \\
& \quad - v_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& c_1, \\
& c_{123}, \\
& (2a_1c_{234} + a_2c_{234} + a_3c_{234} + a_4c_{234} - v_1s_{234}(\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - v_2s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - v_3s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - v_4s_{234}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& (2a_1c_{34} + 2a_2c_{34} + a_3c_{34} + a_4c_{34} - v_3s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - v_4s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - v_1s_{34}(\dot{q}_3 + \dot{q}_4) - v_2s_{34}(\dot{q}_3 + \dot{q}_4)), \\
& (2a_1c_2 + a_2c_2 - \dot{q}_2v_1s_2 - v_2s_2(\dot{q}_1 + \dot{q}_2)), \\
& (2a_1c_{23} + a_2c_{23} + a_3c_{23} - v_2s_{23}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\
& \quad - v_3s_{23}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) - v_1s_{23}(\dot{q}_2 + \dot{q}_3)), \\
& \dot{q}_1, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_1\dot{q}_1), \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0]
\end{aligned}$$

$$\begin{aligned}
Y_2 = & [0, \\
& (a_1 + a_2 + a_3), \\
& (a_1 + a_2 + a_3 + a_4), \\
& (a_1 + a_2), \\
& c_{1234}, \\
& (2a_1c_3 + 2a_2c_3 + a_3c_3 - \dot{q}_3v_1s_3 - \dot{q}_3v_2s_3 - v_3s_3(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)), \\
& c_{12}, \\
& (2a_1c_4 + 2a_2c_4 + 2a_3c_4 + a_4c_4 - \dot{q}_4v_1s_4 - \dot{q}_4v_2s_4 \\
& \quad - \dot{q}_4v_3s_4 - v_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& 0, \\
& c_{123}, \\
& (a_1c_{234} + \dot{q}_1v_1s_{234}), \\
& (2a_1c_{34} + 2a_2c_{34} + a_3c_{34} + a_4c_{34} - v_3s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) \\
& \quad - v_4s_{34}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4) - v_1s_{34}(\dot{q}_3 + \dot{q}_4) - v_2s_{34}(\dot{q}_3 + \dot{q}_4)), \\
& (a_1c_2 + \dot{q}_1v_1s_2), \\
& (a_1c_{23} + \dot{q}_1v_1s_{23}), \\
& 0, \\
& \dot{q}_2, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_2\dot{q}_2), \\
& 0, \\
& 0, \\
& a_2, \\
& 0, \\
& 0]
\end{aligned}$$

$$\begin{aligned}
Y_3 = & [0, \\
& (a_1 + a_2 + a_3), \\
& (a_1 + a_2 + a_3 + a_4), \\
& 0, \\
& c_{1234}, \\
& (a_1c_3 + a_2c_3 + v_1s_3(\dot{q}_1 + \dot{q}_2) + v_2s_3(\dot{q}_1 + \dot{q}_2)), \\
& 0, \\
& (2a_1c_4 + 2a_2c_4 + 2a_3c_4 + a_4c_4 - \dot{q}_4v_1s_4 - \dot{q}_4v_2s_4, \\
& \quad - \dot{q}_4v_3s_4 - v_4s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3 + \dot{q}_4)), \\
& 0, \\
& c_{123}, \\
& (a_1c_{234} + \dot{q}_1v_1s_{234}), \\
& (a_1c_{34} + a_2c_{34} + v_1s_{34}(\dot{q}_1 + \dot{q}_2) + v_2s_{34}(\dot{q}_1 + \dot{q}_2)), \\
& 0, \\
& (a_1c_{23} + \dot{q}_1v_1s_{23}), \\
& 0, \\
& 0, \\
& \dot{q}_3, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_3\dot{q}_3), \\
& 0, \\
& 0, \\
& a_3, \\
& 0]
\end{aligned}$$

$$\begin{aligned}
Y_4 = & [0, \\
& 0, \\
& (a_1 + a_2 + a_3 + a_4), \\
& 0, \\
& c_{1234}, \\
& 0, \\
& 0, \\
& (a_1 c_4 + a_2 c_4 + a_3 c_4 + v_1 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\
& \quad + v_2 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) + v_3 s_4(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)), \\
& 0, \\
& 0, \\
& (a_1 c_{234} + \dot{q}_1 v_1 s_{234}), \\
& (a_1 c_{34} + a_2 c_{34} + v_1 s_{34}(\dot{q}_1 + \dot{q}_2) + v_2 s_{34}(\dot{q}_1 + \dot{q}_2)), \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& 0, \\
& \dot{q}_4, \\
& 0, \\
& 0, \\
& 0, \\
& \tanh(\nu_4 \dot{q}_4), \\
& 0, \\
& 0, \\
& a_4]
\end{aligned}$$