

**IMPEDANCE CONTROL WITHOUT FORCE SENSORS WITH APPLICATION
IN HOMECARE ROBOTICS**

by

Yanjun Wang

M. A. Sc., Shanghai Jiao Tong University, 2009

B. A. Sc., Nanjing University of Aeronautics & Astronautics, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(MECHANICAL ENGINEERING)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

November, 2014

© Yanjun Wang, 2014

Abstract

This thesis addresses the problem of interaction control between robot manipulator and the manipulated object in a homecare project. This project aims to use homecare robots at the elderly or disabled people's home to provide necessary aid and assistance. The robot manipulator is to be operated in autonomous mode or teleoperation mode. The possible first aid or assistance requires direct interaction between the remote side robot manipulator and the human body.

To guarantee the compliant interaction between the manipulator and the human body, impedance control was applied. In impedance control, neither the force nor the actual motion of the manipulator is controlled. The dynamic relationship between the interaction force and the resulting motion is controlled so that the interaction force will be monitored and kept at an acceptable range.

To shape the mechanical impedance to any desired value as we wish, the remote side interaction force sensing is required. The interaction force could be sensed by a force sensor. Force sensors have a lot of inherent limitations such as narrow bandwidth, sensing noise, and high cost. To avoid a force sensor due to its limitations, sliding mode observers will be applied to estimate the interaction force. The estimated interaction force will be used in the impedance control algorithms. The observer and controller framework will be formulated and the solvability will be discussed thoroughly. In addition, the proposed approach will be compared with some available approaches to show its advantages over others.

Bilateral impedance control will be applied in a teleoperation system. The master side impedance controller is to ensure the robust stability of the teleoperation system. The remote slave side impedance controller is used so that the interaction force will be monitored and kept at some acceptable range. Desired impedance parameters selection will be discussed considering the compromise between robust stability and performance. Also, in order to deal with the uncertainties in operator and environment dynamics, a robust performance guaranteed controller synthesis approach will be proposed. Gain-

scheduling control could guarantee the stability and the robust performance under those uncertainties.

Preface

The entire work presented in this thesis was conducted at the Industrial Automation Laboratory of the University of British Columbia, Vancouver campus under the supervision of Dr. Clarence W. de Silva. I was responsible for plant modeling, control algorithm design and experimentation with the guidance and advice of my supervisor, Dr. Clarence W. de Silva.

A version of Chapter 2 has been submitted for publication to a journal and is under review: Yanjun Wang and Clarence W. de Silva (2014). “Dynamic Model Identification of a Simplified Two-Link Manipulator with Robust Differentiators”.

A version of Chapter 2 has been published: Yanjun Wang and Clarence W. de Silva (2014). “An Experimental Setup for Dynamic Control of Robot Manipulators.,” *Proceedings of International Conference on Computer Science & Education*, Vancouver, Canada, 2014.

A version of Chapter 3 has been submitted to a journal and is under review: Yanjun Wang and Clarence W. de Silva (2014). “Adaptive Sliding Mode Observer based Interaction Force Estimation in Robotics”.

A version of Chapter 4 has been submitted to a journal and is under review: Yanjun Wang and Clarence W. de Silva (2014). “Robust Impedance Control with Adaptive Interaction Force Observation”.

A version of Chapter 5 has been published: Yanjun Wang and Clarence W. de Silva (2012). “Bilateral Impedance Control in Haptic based Teleoperation with Sliding Mode Control,” *Proceedings of CSME International Congress*, Winnipeg, Canada, 2012.

A version of Chapter 5 has been published: Yanjun Wang and Clarence W. de Silva (2012). “Bilateral Teleoperation System Control with Guaranteed Robust Performance under Human Operator and Environment Uncertainty,” *Proceedings of CSME International Congress*, Winnipeg, Canada, 2012.

A version of Chapter 5 has been submitted to a journal and is under review: Yanjun Wang and Clarence W. de Silva (2014). “Bilateral Impedance Control in Teleoperation with Task-Dependent Impedance Parameters and Interaction Force Estimation”.

In all the listed publications (published or submitted), I proposed the control algorithms and finished the numerical simulations or experiments with direct advice, guidance and supervision of my supervisor. I wrote the manuscripts, which were edited and refined by Dr. Clarence W. de Silva.

This work has been supported by research grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canada Foundation for Innovation (CFI), the British Columbia Knowledge Development Fund (BCKDF), and the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation held by C.W. de Silva.

Table of Contents

Abstract	ii
Preface	iv
Table of Contents.....	vi
List of Tables	x
List of Figures	xi
List of Symbols.....	xvii
List of Abbreviations	xix
Acknowledgements	xxi
Chapter 1: Introduction	1
1.1 Background and Motivation.....	1
1.2 Problem Formulation.....	2
1.3 Related Work.....	6
1.3.1 Sliding Mode Systems	6
1.3.2 Sliding Mode Observers	12
1.3.3 Robust Differentiators based on Sliding Mode Theory	13
1.3.4 Robot Dynamics and Identification	15
1.3.5 Interaction Control	19
1.3.6 Force Control without Force Sensors	25
1.3.7 Teleoperation	30
1.4 Contributions and Organizations of the Thesis	31
Chapter 2: Modeling and Identification of Robot Dynamics	34
2.1 Introduction	35

2.2	Development of the Experimental Setup.....	38
2.2.1	Hardware of xPC Target System	39
2.2.2	Software of xPC Target System.....	41
2.2.3	WAM Simplification	44
2.2.4	Fixing Joints 1 and 3 by Software.....	48
2.3	Dynamic Model Derivation.....	49
2.3.1	Joint Space Dynamic Model	49
2.3.2	Cartesian Space Dynamic Model.....	54
2.3.3	Problem with Dynamic Model based on CAD Models	56
2.4	Dynamic Model Identification	63
2.4.1	Reference Trajectory Selection.....	63
2.4.2	Velocity and Acceleration Reconstruction	66
2.4.3	Offline Identification of Dynamic Model	72
2.4.4	Online Dynamic Model Identification	76
2.4.5	Offline Dynamic Model Identification with Neural Network based Compensator	82
2.4.6	Comparison with Conventional Differentiator based Identification.....	86
2.5	Summary	90
Chapter 3: Observers-based Estimation of Interaction Force		91
3.1	Interaction Force Estimation	91
3.1.1	Cartesian Space Reference Trajectory Selection	91
3.1.2	Interaction Force Calculation Using Identified Dynamic Model.....	98
3.1.3	Interaction Force Estimation Using Second-order Sliding Mode Observer (SOSMO).....	103
3.1.4	Adaptive High-order Sliding Mode Observer (AHOSMO) based Interaction Force Estimation.....	110
3.2	Summary	116

Chapter 4: Impedance Control without Direct Force Sensing	117
4.1 Introduction	118
4.1.1 Impedance Control in Interaction Applications	118
4.1.2 Incomplete Impedance Control without Force Sensors	120
4.2 Impedance Control Using AHOSMO-estimated Interaction Force	121
4.2.1 Problem Formulation	121
4.2.2 Cartesian Space Reference Trajectory Selection	122
4.2.3 Interaction Control through Position Control	124
4.2.4 Accuracy of Impedance Control	127
4.2.5 Inverse Dynamics-based Impedance Controller	129
4.2.6 First-order Sliding Mode Controller-based Impedance Control	133
4.3 Summary	137
Chapter 5: Applications in Bilateral Teleoperation	138
5.1 Bilateral Impedance Control with Task-dependent Impedance Parameters	139
5.1.1 System Modeling	140
5.1.2 Selection of Desired Impedance Parameters.....	143
5.1.3 Transparency Analysis of the Teleoperation System.....	146
5.1.4 Numerical Simulation Results	147
5.1.5 Conclusion & Discussion.....	148
5.2 Robust Performance by Gain-scheduling Control.....	148
5.2.1 Preliminaries for Gain-scheduling Control of LPV Systems.....	149
5.2.2 Human Operator Modelling	152
5.2.3 Environment Modeling	153
5.2.4 Modeling of Master Slave System	153
5.2.5 Numerical Simulation Results	157
5.2.6 Summary	159

Chapter 6: Conclusions and Future Work.....	160
6.1 Conclusions	160
6.2 Contributions	162
6.3 Significance of the Work.....	163
6.4 Limitations and Suggested Future Work.....	163
Bibliography.....	166

List of Tables

Table 2.1: DH parameters of WAM.....	36
--------------------------------------	----

List of Figures

Figure 1.1: Schematic representation of the developed system.....	3
Figure 1.2: Variable structure system	6
Figure 1.3: The saturation function	9
Figure 1.4: Task of drilling a hole in the wood piece	20
Figure 1.5: Application of hybrid position/force control.....	22
Figure 1.6: Motion-based impedance control (admittance control)	24
Figure 1.7: Torque-based impedance control	25
Figure 1.8: Disturbance observer in robot dynamics control	28
Figure 1.9: Schematic representation of a teleoperation system	30
Figure 2.1: Whole Arm Manipulator (WAM) from Barrett Technology	35
Figure 2.2: Schematic representation of WAM	35
Figure 2.3: Hardware communications in WAM	37
Figure 2.4: xPC Target-based experimental setup.....	40
Figure 2.5: CAN initialization block	41
Figure 2.6: Joint position report request	42
Figure 2.7: Reporting joint positions	42
Figure 2.8: WAM mode check request.....	43
Figure 2.9: WAM mode report	43
Figure 2.10: WAM torque command sending switch.....	43
Figure 2.11: WAM torque command sending	44
Figure 2.12: Flowchart of the software system	44
Figure 2.13: Schematic representation of simplified WAM	45
Figure 2.14: Simplified two-link planar manipulator	46
Figure 2.15: Coordinate frames of simplified two-link planar manipulator.....	46
Figure 2.16: Simplified WAM system (joint 1)	47
Figure 2.17: Simplified WAM system (joint 3)	48
Figure 2.18: Free body diagram of link i	49
Figure 2.19: Joint space position reference trajectories.....	57

Figure 2.20: Joint space velocity reference trajectories.....	58
Figure 2.21: Joint space acceleration reference trajectories	58
Figure 2.22: WAM starting time snapping	59
Figure 2.23: Internal block diagram representing the control algorithm.....	60
Figure 2.24: Global block diagram representing the control algorithm	60
Figure 2.25: Actual joint space position trajectories	61
Figure 2.26: Actual joint space velocity trajectories	61
Figure 2.27: Actual joint space acceleration trajectories.....	61
Figure 2.28: Joint torque trajectories	62
Figure 2.29: Position reference trajectories for each joint.....	65
Figure 2.30: Velocity reference trajectories for each joint	66
Figure 2.31: Actual joint position trajectories	70
Figure 2.32: Actual joint velocity trajectories based on conventional differentiator	70
Figure 2.33: Actual joint acceleration trajectories based on conventional differentiator	70
Figure 2.34: Actual joint velocity trajectories based on robust differentiator	71
Figure 2.35: Actual joint acceleration trajectories based on robust differentiator	71
Figure 2.36: Validation scheme of the offline identified dynamic parameter	72
Figure 2.37: Reference trajectories for joint 1 and 2	73
Figure 2.38: Trajectory tracking performance for joints 1 and 2 (offline)	73
Figure 2.39: Comparison between the predicted torque and the actual torque (offline) ..	74
Figure 2.40: Difference between joint 1 predicted torque and actual torque (offline)	74
Figure 2.41: Comparison between joint 2 predicted torque and actual torque (offline) ..	75
Figure 2.42: Difference between the predicted torque and the actual torque (offline).....	75
Figure 2.43: Friction compensation based on online parameter estimation	77
Figure 2.44: Trajectory tracking performance for joint 1 and 2 (online)	78
Figure 2.45: Online estimation of F_{c1}	78
Figure 2.46: Online estimation of F_{v1}	79
Figure 2.47: Online estimation of F_{c2}	79
Figure 2.48: Online estimation of F_{v2}	80
Figure 2.49: Scheme of the online dynamic parameter validation process	80
Figure 2.50: Trajectory tracking with online friction compensation	81
Figure 2.51: Joint 1 actual torque vs. predicted torque with online identification	81

Figure 2.52: Joint 1 torque prediction error with online dynamic parameter identification	81
Figure 2.53: Joint 2 actual torque vs. predicted torque with online identification	82
Figure 2.54: Joint 2 torque prediction error with online dynamic parameter identification	82
Figure 2.55: Neural Network-based torque compensator	83
Figure 2.56: Scheme of the validation process of offline dynamic parameters with Neural Network-based compensator	84
Figure 2.57: Joint 1 actual torque vs. predicted torque with offline identification (with torque compensator)	84
Figure 2.58: Joint 1 torque prediction error (with torque compensator)	85
Figure 2.59: Joint 2 actual torque vs. predicted torque with offline identification (with torque compensator)	85
Figure 2.60: Joint 2 torque prediction error (with torque compensator)	85
Figure 2.61: Joint acceleration trajectories after low-pass filtering	87
Figure 2.62: Joint torque trajectories after low-pass filtering.....	87
Figure 2.63: Joint space trajectory tracking (with dynamic parameters identified using filters).....	88
Figure 2.64: Joint 1 actual torque vs. predicted torque (with dynamic parameters identified using filters).....	88
Figure 2.65: Joint 1 torque prediction error (with dynamic parameters identified using filters).....	89
Figure 2.66: Joint 2 actual torque vs. predicted torque (with dynamic parameters identified using filters).....	89
Figure 2.67: Joint 2 torque prediction error (with dynamic parameters identified using filters).....	89
Figure 3.1: Interaction force estimation validation experimental setup	92
Figure 3.2: Coordinate system of the force sensor	92
Figure 3.3: Two degree-of-freedom WAM under external force	93
Figure 3.4: Reference trajectories of joint 1 and joint 2 (positions).....	96
Figure 3.5: Reference trajectories of joint 1 and joint 2 (velocities)	96
Figure 3.6: Cartesian space position reference trajectories	97

Figure 3.7: The validation scheme of the interaction force estimation	97
Figure 3.8: Actual joint space position trajectories (with inverse dynamics using identified dynamic parameters and direct differentiation).....	98
Figure 3.9: Actual Cartesian space trajectories (with inverse dynamics using identified dynamic parameters and direct differentiation).....	99
Figure 3.10: Estimated force vs. measured force (x direction with inverse dynamics using identified dynamic parameters and direct differentiation).....	99
Figure 3.11: Force estimation error (x direction with inverse dynamics using identified dynamic parameters and direct differentiation).....	99
Figure 3.12: Estimated force vs. measured force (z direction with inverse dynamics using identified dynamic parameters and direct differentiation).....	100
Figure 3.13: Force estimation error (z direction with inverse dynamics using identified dynamic parameters and direct differentiation).....	100
Figure 3.14: Actual joint space position trajectories (with inverse dynamics using identified dynamic parameters and robust differentiator)	101
Figure 3.15: Actual Cartesian space position trajectories (with inverse dynamics using identified dynamic parameters and robust differentiator)	101
Figure 3.16: Estimated force vs. measured force (x direction with inverse dynamics using identified dynamic parameters and robust differentiator)	102
Figure 3.17: Force estimation error (x direction with inverse dynamics using identified dynamic parameters and robust differentiator).....	102
Figure 3.18: Estimated force vs. measured force (z direction with inverse dynamics using identified dynamic parameters and robust differentiator)	102
Figure 3.19: Force estimation error (z direction with inverse dynamics using identified dynamic parameters and robust differentiator).....	103
Figure 3.20: Estimated force vs. measured force (x direction with SOSMO, $T_i=0.01$)..	106
Figure 3.21: Force estimation error (x direction with SOSMO, $T_i=0.01$)	107
Figure 3.22: Estimated force vs. measured force (z direction with SOSMO, $T_i=0.01$)..	107
Figure 3.23: Force estimation error (z direction with SOSMO, $T_i=0.01$).....	107
Figure 3.24: Estimated force vs. measured force (x direction with SOSMO, $T_i=0.1$)....	108
Figure 3.25: Force estimation error (x direction with SOSMO, $T_i=0.1$)	108
Figure 3.26: Estimated force vs. measured force (z direction with SOSMO, $T_i=0.1$)....	108

Figure 3.27: Force estimation error (z direction with SOSMO, $T_i=0.1$).....	109
Figure 3.28: Estimated force vs. measured force (x direction with SOSMO, $T_i=0.05$)..	109
Figure 3.29: Force estimation error (x direction with SOSMO, $T_i=0.05$)	109
Figure 3.30: Estimated force vs. measured force (z direction with SOSMO, $T_i=0.05$)..	110
Figure 3.31: Force estimation error (z direction with SOSMO, $T_i=0.05$).....	110
Figure 3.32: Estimated force vs. measured force (x direction with AHOSMO)	114
Figure 3.33: Force estimation error (x direction with AHOSMO)	115
Figure 3.34: Estimated force vs. measured force (z direction with AHOSMO).....	115
Figure 3.35: Force estimation error (z direction with AHOSMO)	115
Figure 4.1: Original mechanical impedance of a mass block.....	118
Figure 4.2: Actively controlled mechanical impedance	119
Figure 4.3: Impedance control in robot interaction applications.....	119
Figure 4.4: Cartesian space reference trajectories (for interaction control)	124
Figure 4.5: Joint space reference trajectories (for interaction control).....	124
Figure 4.6: Interaction force under pure position control (x direction)	126
Figure 4.7: Interaction force under pure position control (z direction)	126
Figure 4.8: Cartesian space trajectories (with inverse dynamics-based impedance control algorithms).....	130
Figure 4.9: Joint space trajectories (with inverse dynamics-based impedance control algorithms).....	130
Figure 4.10: Interaction force (x direction with inverse dynamics-based impedance control algorithms).....	131
Figure 4.11: Interaction force (z direction with inverse dynamics-based impedance control algorithms).....	131
Figure 4.12: Joint torque trajectories (with inverse dynamics-based impedance control algorithms).....	132
Figure 4.13: Impedance control accuracy (with inverse dynamics-based impedance control algorithms).....	132
Figure 4.14: Interaction Force (x direction with SMC-based impedance control algorithms).....	136
Figure 4.15: Joint torque trajectories (with SMC-based impedance control algorithms)	136

Figure 4.16: Impedance control accuracy (with SMC-based impedance control algorithms).....	136
Figure 5.1: Task-dependent bilateral impedance control scheme	140
Figure 5.2: Network representation of a teleoperation system	144
Figure 5.3: Transformation between power variables and scattering variables	144
Figure 5.4: Network representation of a teleoperation system in scattering domain	145
Figure 5.5: Potentially unstable in scattering domain	145
Figure 5.6: Potentially unstable in scattering domain (overlap).....	146
Figure 5.7: Absolutely stable in scattering domain	146
Figure 5.8: Force tracking performance	147
Figure 5.9: Force tracking performance (local zoomed view)	147
Figure 5.10: Position tracking performance	147
Figure 5.11: Position tracking performance (local zoomed view)	148
Figure 5.12: Linear fractional transformation representation of a general system.....	150
Figure 5.13: Linear fractional transformation representation of a teleoperation system	154
Figure 5.14: Linear fractional transformation representation of a teleoperation system with filters.....	156
Figure 5.15: Signal for environment disturbance with identification error	158
Figure 5.16: Position tracking error.....	158
Figure 5.17: Force tracking error.....	159

List of Symbols

B_d	Desired Damping Matrix
C	Joint Space Coriolis/Centrifugal Matrix
C_C	Cartesian Space Coriolis/Centrifugal Matrix
\bar{C}	Nominal Joint Space Coriolis/Centrifugal Matrix
F_C	Joint Space Coulomb Friction Coefficient Matrix
F_{CC}	Cartesian Space Coulomb Friction Coefficient Matrix
F_e	Interaction Force
\hat{F}_e	Estimated Interaction Force
F_V	Joint Space Viscous Friction Coefficient Matrix
F_{VC}	Cartesian Space Viscous Friction Coefficient Matrix
G	Joint Space Gravity Matrix
G_C	Cartesian Space Gravity Matrix
\bar{G}	Nominal Joint Space Gravity Matrix
J	Manipulator Jacobian Matrix
K_d	Desired Stiffness Matrix
M	Joint Space Inertia Matrix

\bar{M}	Nominal Joint Space Inertia Matrix
M_c	Cartesian Space Inertia Matrix
M_d	Desired Inertia Matrix
q	Joint Position
s	Sliding Variable
T_c	Finite Convergence Time
T_s	Sampling Period of the Computer Control System
u_{eq}	Equivalent Control Input of a Sliding System
Y	Regressor Matrix
Π	Base Dynamic Parameters
$\Delta\tau$	Torque Output of the Neural Network based Compensator
τ	Joint Torque

List of Abbreviations

AHOSMO	Adaptive High Order Sliding Mode Observer
BIAS	Bounded Impedance Absolute Stability
CAD	Computer Aided Design
CAN	Controller Area Network
DH	Denavit-Hartenberg
DIDIM	Direct and Inverse Dynamic Identification Models
DOB	Disturbance Observer
DOF	Degree of Freedom
EKF	Extended Kalman Filter
EMG	Electromyography
FIR	Finite Impulse Response
GA	Genetic Algorithm
HOSM	High Order Sliding Mode
HOSMO	High Order Sliding Mode Observer
IDE	Integrated Development Environment
KFDOB	Kalman Filter-based Disturbance Observer
KFSO	Kalman Filter-based State Observer
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying

LSE	Least Squares Estimation
MGAID	Modified Genetic Algorithm Identification Method
PD	Proportional Derivative
PID	Proportional Integral Derivative
PSO	Particle Swam Optimization
RAC	Robust Acceleration Control
RF/TOB	Robust Force/Torque Observer
RHS	Right Hand Side
RLSE	Recursive Least Squares Estimation
ROS	Robot Operating System
SISO	Single Input Single Output
SOSM	Second Order Sliding Mode
SOSMO	Second Order Sliding Mode Observer
SRIV	Simple Refined Instrumental Variable
STA	Super Twisting Algorithm
TCP/IP	Transmission Control Protocol
TDE	Time Delay Estimation
TI	Time Invariant
UDP	User Datagram Protocol
VSS	Variable Structure System
WAM TM	Whole Arm Manipulator

Acknowledgements

First, I wish to express my sincere gratitude to my supervisor, Dr. Clarence W. de Silva for suggesting the topic of the research and for providing advice, supervision and careful guidance in my subsequent activities of research. I still remember the first time I contacted Dr. de Silva when I was applying for PhD studies. He replied me with patience and respect, and he revised my research proposal very carefully, even its format. His serious attitude toward study and his understanding about mechanical engineering and electrical engineering greatly impressed me. Also, I am grateful to him for providing all the equipment, hardware, and other resources for carrying out my research in the Industrial Automation Laboratory. I wish him all the best and he will be the professor I respect forever.

I would also like to thank Dr. Ying Wang (a former student of Dr. de Silva) who recommends me to Dr. de Silva, and it is he who helped me to obtain this valuable opportunity. Our laboratory colleague Dr. Roland Haoxiang Lang (a student of Dr. de Silva and the present Laboratory Manager) also helped me and I will always remember those happy days we spent together. Another former student and Laboratory Manager of Dr. de Silva, Dr. Muhammad Tahir Khan is acknowledged as well. I still remember the jokes we all played when we were sitting in the same office in the lab.

Thanks you all, my colleagues in the Industrial Automation Laboratory (IAL), Mr. Yunfei Zhang, Ms. Yu Du, Mr. Muhammad Tufail Khan, Mr. Shan Xiao, Ms. Lili Meng, Mr. Hani Balkhair, Ms. Pegah Maghsoud, Mr. Min Xia, Mr. Shujun Gao, Mr. Zuo Chen, Mr. Teng Li., Mr. Tongxin Shu for your friendship and help.

This work has been supported by research grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canada Foundation for Innovation (CFI), the British Columbia Knowledge Development Fund (BCKDF), and the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation held by C.W. de Silva.

I would also like to express my deepest gratitude to my father Jincheng Wang and my mother Jinbao Cai for their unconditional love and support. Zhijie Yang, my wife, you are the one in my life, Thank you.

Chapter 1: Introduction

Robotics research is a highly multidisciplinary area, which benefits from developments in computer science, control theory, mechanics, electrical engineering, and cognitive science. The increasing computing power of processors and the lower cost have made the implementation of advanced control algorithms feasible. The advancement in sensing technology makes it possible that the robots perceive their environment more comprehensively. Thanks to the rather exhaustive and accurate description of the environment through use of different types of sensors, the robots can become more intelligent and are able to operate in unknown and unstructured environments. The improved performance and reduced cost make it possible to equip robots not just for industrial situations, such as automotive production lines, but also for such human-interactive applications as homecare and rescue applications. Robots are increasingly used in service applications due to the high cost and other limitations of labor. These robots could provide higher-quality service while reducing the cost to the users. To complement the existing research activities in this area, the project of homecare robotics was launched at the Industrial Automation Laboratory, University of British Columbia. It focuses on home automation and concentrates on the application: robot-automated service providers (caregivers) at home for people with physical and cognitive impairments.

As part of this project, this thesis aims to investigate the problem of compliant interaction between the robot end-effector and the assisted elder or disabled person, leading to the development of a robust, low-cost and compliant approach of interaction control. The proposed control algorithms are expected to realize soft interaction between the human body and a homecare robot manipulator, which guarantees that no excessive force is exerted on the human body, for safety reasons.

1.1 Background and Motivation

The problem of population ageing is a global one due to the rising life expectancy and declining birth rate. Initially, this has been a problem primarily in countries that are economically more developed. But recently, developing countries such as China and India are also affected by this problem. According to the prediction of Statistics Canada,

the percentage of the senior population will reach 25% of the total population by 2050. The social and economic effects of population ageing are enormous. The elderly need some care and physical assistance when necessary, in their daily life. The total cost of providing care to people of this group is tremendous, considering the large size of the ageing population.

Another group of people that will rapidly increase its size are those with physical and cognitive impairments. The Canadian government spends about \$9 billion on disability related matters. The cost of basic care for a disabled person at home is about \$10,000/month, and this is a huge burden on the Government.

The problem described above may be resolved by the introduction of a homecare robotics system, which consists of groups of robotic devices to work together in a coordinated and efficient manner and carry out a common task of providing assistance to elderly and/or disabled people in the home setting. The needed technologies of robotics, network communication, and control are sufficiently mature and are available at reasonable cost. Due to the lack of necessary assistive technologies and specialized end-effector devices, and also due to insufficient efforts in bringing the pertinent technology teams together, the development of affordable and reliable systems has eluded the application area in the past. The research in homecare robotics is a new but rapidly developing field of service robotics, especially in Japan.

Through our established laboratory facilities we plan to develop affordable and effective service robots for the elderly or the disabled people with physical and cognitive impairments in a home setting. The developed system comprises several sets of robot manipulators having mobile bases, and can provide assistance in daily activities, medical assistance, surveillance, and so on. They can work independently or collaboratively in a team, based on the complexity of the task. The designed homecare robotics system allows the care-receiver to stay in their familiar home environment where the assistance is provided, which is a merit of the technologies.

1.2 Problem Formulation

The primary objective of the present work is to develop a robust and low cost manipulation system which has abilities to safely interact with the elderly and disabled in a home environment. Possible application scenarios include first aid and assistance in

changing clothes. A schematic representation of the proposed homecare robotic system is shown in Figure 1.1.

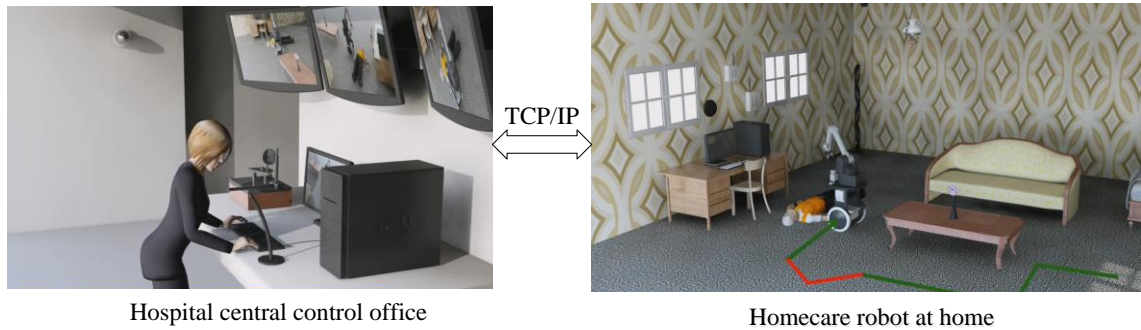


Figure 1.1: Schematic representation of the developed system

The system consists of two parts, the local homecare system at the care-receiver's home and the hospital control room. Besides the homecare robotic manipulator mounted on a mobile base, the system in the home environment includes sensors for monitoring and surveillance of the home environment, such as cameras, microphone, gas leakage sensor and so on. Sensors for monitoring the health condition of the user, such as the heart rate sensor and blood pressure, will be able to provide a complete description of the care-receiver's health status to the medical professionals in the control room. Of course, the sensors that are needed by the robot to carry out its activities (encoders, cameras, tactile sensors, etc.) are needed as well.

The robotic subsystem in the hospital control room is made up of a Phantom Premium 1.5 6DOF, which is a haptic master device that can operate the homecare robot remotely by sending out the hand movement command of the operator while providing force feedback. Also, for direct communication with the care-receiver, this subsystem has audio devices, including microphones and speakers. The care-receiver at home can maintain bidirectional communication with the operator in the hospital control room. The communication between the home side and the hospital control room is facilitated through the Internet or a dedicated Ethernet.

As noted before, the homecare robot system is able to operate in two different modes. The first is the autonomous mode, which is the research thrust of several members of the Industrial Automation Laboratory [1],[2]. In this mode, the care-receiver is assisted by the homecare robot system by itself without the involvement of the hospital control room. In addition to routine care, it can involve medical care (e.g., dispensing of prescription medication) without involving a medical professional. These may include

easy tasks that the homecare robots are able to carry out, such as pressing the power button of a household appliance. When the task is too complex and critical nature for autonomously carrying out by a homecare robot, the second mode of operation of the homecare robot is triggered. This second is the teleoperation mode, in which the robot is semi-autonomous, and needs the intervention of a professional in the remote control room. In this situation, the homecare system with the help of various sensors will notify this problem to the remote control room and will seek assistance.

In this thesis, the research contributions are applicable in both modes. When the robot manipulator interacts with human body irrespective of whether it is in the autonomous mode or the teleoperation mode, safety of operation should come first. In the teleoperation mode, the position command from the operator (master) in the hospital control room is transmitted to the homecare robot. The transmitted information from the operator is the motion command for the slave homecare robot system to follow. When the homecare robot is in contact with the human body, the interaction force and its rate of change should be controlled delicately in order not to have excessive force on the human body. In the meantime, the contact force should be feedback to the master manipulator in the hospital control room. The feedback force will be displayed by the master haptic device to the operators so that they are actually “feeling” the contact information of the remote side. Accordingly, they can send out a new command through the haptic device based on their experience. When the master haptic device and the homecare robot are formulated into a close-loop system, we face several challenges. Only four main challenges are discussed here and the corresponding solutions are proposed in the subsequent chapters of this thesis. They are all key issues in realizing an effective and practical homecare robotic system.

The first challenge is the sensing of the interaction force. In constrained motion control of a robot, interaction force is crucial information that describes the state of interaction between the robot manipulator and the contact object. The interaction force feedback can be included in the control algorithms for regulation or tracking purposes. Conventional force sensors are costly and have considerable noise in the sensed force data [3]. Furthermore, mounting a force sensor at the manipulator-object is quite difficult. Also, a suitable force sensor may not be readily available in the laboratory facilities. Sensor failure or inaccurate force sensing can deteriorate the stability and

performance of the whole system. Due to such reasons, it may be required to estimate the interaction force indirectly.

The second challenge is the compliant interaction between the homecare robot and the human body no matter whether the robot is in autonomous mode or in teleoperation mode. Conventional industrial robots are heavy and have large inertia, while in the present project, the homecare robots can be designed to be light-weight and compact. Even when the inertia is low in this kind of robots, at high accelerations, the resulting forces may still be a threat to the safety of the user during interaction. This is more serious since the user is a senior or disabled, who is generally more vulnerable [4].

The impedance of the robot should be actively tuned to accommodate the application. Uncertainty in the dynamic models is unavoidable due to the time-varying nature of the dynamics and necessary approximation of the robot manipulator. Robust impedance shaping should be achieved using nominal robot dynamic models. Conventionally, in impedance control the end-effector velocity is required. It is calculated based on the differential kinematic relationship between the Cartesian-space velocity and the joint space velocity.

The revolute joints of robot manipulator are equipped with encoders to sense the motion of each joint. In the homecare system of the present work, the 4 degree of freedom (DOF) the serial-link robot manipulator called WAM (Whole Arm Manipulator) manufactured by Barrett Technology [5] is used. The incremental optical encoder in each joint can sense the joint position accurately. Joint velocity information is required for the impedance control implementation as mentioned in the previous challenge. However, if we reconstruct the velocity by directly differentiating the sensed joint position information with respect to time, we will get an extremely noisy signal [6]. Filters could be applied when we know the bandwidth of the velocity signal, but not in the case of white noise. Reconstruction of the joint velocity for impedance control implementation purpose is the third challenge in this research.

In bilateral teleoperation, the master side is usually impedance controlled to reflect the contact force to the operator, while the slave side is impedance controlled to guarantee the compliant behavior of the robot manipulator. It is commonly assumed that the models of the master and slave sides are linear time invariant [7]. Generally, the robot manipulator dynamics is coupled, nonlinear and configuration dependent. Under ideal

impedance control, in each Cartesian degree of freedom, the manipulator can be linearized and decoupled. Thus, robust impedance control is critical for extending the available research results on teleoperation.

In this thesis, algorithms are developed to address these key challenges. The proposed approaches for impedance control and bilateral teleoperation can work robustly and effectively under uncertainty and in the absence of force sensor.

1.3 Related Work

1.3.1 Sliding Mode Systems

It is unavoidable that all practical systems are under some kind of disturbance and uncertainty. Controlling plants to have some desired properties in the presence of uncertainty remains an active research field. Sliding mode control is an important class of such controllers and it originated primarily by scholars in the former Soviet Union in the late 1950s. Since its introduction, sliding mode technique has found many applications in the design of controllers [8]–[11], observers [12]–[15] and differentiators [16].

Sliding mode system is a special case of variable structure system (VSS), where the system dynamics switches when crossing a manifold called sliding surface [17]. The sliding surface is selected so that the desired system performance is achieved once the system state is constrained onto this manifold. Usually, this desired performance could not be achieved in either subsystem, which is an important feature of sliding mode systems.

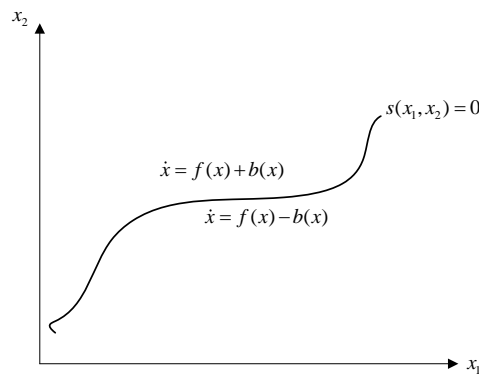


Figure 1.2: Variable structure system

Figure 1.2 shows a sliding mode system which may be represented by

$$\dot{x} = f(x) + b(x) \cdot \text{sign}(s) \quad (1.1)$$

The Lipschitz property of the right hand side (RHS) of the differential equation guarantees the existence and uniqueness of the solution. However, when the right hand side of the differential equation is discontinuous, the existence and uniqueness of the solution cannot be guaranteed. The vector field at $s=0$ is not determined since the function $sign(s)$ at $s=0$ is not defined.

One approach to analyze differential equations with discontinuous right hand side is by using the concept of Filippov inclusion [18]. The value of the derivatives on the discontinuous sliding surface may be determined by

$$f_0 = \text{conv}(f(x)+b(x), f(x)-b(x)) \cap \dot{s} \quad (1.2)$$

An alternative way to find the vector field on the sliding surface is by the use of the equivalent control concept. This concept is the basis for state observers. Ideally, once the system states are constrained onto the sliding surface, then $s=0$ should be valid. For a conventional sliding mode system, $\dot{s}=0$ should also be valid in the ideal sliding mode. Based on $\dot{s}=0$, we can calculate the control input u , which is named the equivalent control input and denoted by u_{eq} . The equivalent control input corresponds to the slow dynamics of the system [19],[20].

The design of a sliding mode controller consists of two steps. The first step is to select the sliding surface, which has the desired system dynamics once the system states are constrained onto this surface. The second step is to design the controller to drive the system states onto this sliding surface.

Taking the example in [18], consider a disturbed single-input-single-output (SISO) time invariant (TI) system described by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_1, x_2, t) + u \end{aligned} \quad (1.3)$$

where x_1, x_2 are the system states, u is the control input, and $|f(x_1, x_2, t)| \leq L > 0$ is the bounded disturbance. Select the sliding surface defined by

$$s = s(x_1, x_2) = x_2 + kx_1 \quad (k > 0) \quad (1.4)$$

We can get the first derivative of this sliding surface as

$$\dot{s} = \dot{x}_2 + k\dot{x}_1 = kx_2 + f(x_1, x_2, t) + u \quad (1.5)$$

The Lyapunov function is given by

$$V = \frac{1}{2}s^2 \quad (1.6)$$

Using this, the control input that ensures the finite-time convergence to the sliding surface can be designed.

The derivative of this Lyapunov function (1.6) is calculated as

$$\dot{V} = s\dot{s} = s[kx_2 + f(x_1, x_2, t) + u] \quad (1.7)$$

Select the control input as

$$u = -kx_2 - \rho \text{sign}(s) \quad (1.8)$$

$$\text{sign}(s) = \begin{cases} 1 & s > 0 \\ [-1, 1] & s = 0 \\ -1 & s < 0 \end{cases} \quad (1.9)$$

The derivative of the Lyapunov function (1.6) may be represented as

$$\begin{aligned} \dot{V} &= s[f(x_1, x_2, t) - \rho \text{sign}(s)] \\ &= sf(x_1, x_2, t) - s\rho \text{sign}(s) \\ &\leq |s|L - |s|\rho \end{aligned} \quad (1.10)$$

Select the parameter ρ as

$$\rho = L + \frac{\alpha}{\sqrt{2}} \quad (1.11)$$

Then the derivatives of the Lyapunov function can be further simplified as

$$\dot{V} \leq -\alpha V^{1/2} \quad (1.12)$$

Based on the comparison lemma, integrating both sides of equation (1.12) over time interval $[0, t]$, we have

$$V^{1/2}(t) \leq -\frac{1}{2}\alpha t + V^{1/2}(0) \quad (1.13)$$

The finite-convergence time of the system trajectory onto the manifold may be determined as

$$T_c = \frac{2V^{1/2}(0)}{\alpha} \quad (1.14)$$

Note that the parameter α should be tuned properly here. If we want small convergence time, we can have large α . However, we can see from equation (1.9) that

the control input is discontinuous at $s = 0$. The magnitude of this discontinuous input is proportional to α . A too big value for α will lead to undesirable chattering.

Chattering is due to fact that the control input is a discontinuous function. In a robotic manipulator, this control input is the torque command to be sent to the joint actuators in each joint. In the ideal case, the bandwidth of the actuator is infinite so that the joint torque could be switched without time delay. As shown in the Lyapunov-based analysis, the convergence onto the sliding surface will happen in finite time. However, in practice the bandwidth of the actuator is limited. It cannot complete a discontinuous switch in zero time. The intuitive explanation for the cause of chattering may be given using the following example. Assume that at time t_1 the actuator is commanded to have a positive joint torque, which takes the joint actuator Δt_1 to respond to this command and deliver the joint torque. However, the torque command during this Δt_1 has changed to a negative value. After the actuator responds to the positive torque command, it has to deliver a negative torque. This process repeats over and over, which makes the torque delivered by the actuator to chatter.

Chattering is very harmful to mechanical systems, which increases the wear and tear, vibration, noise. Also, it may induce the neglected high-order dynamics in modeling of the system. It has to be eliminated or at least attenuated in practice.

One approach usually used in the past is to replace the discontinuous control input with an approximate high-slope saturation continuous function, as shown in Figure 1.3. This type of approach is usually called boundary layer-based sliding mode control [21],[22].

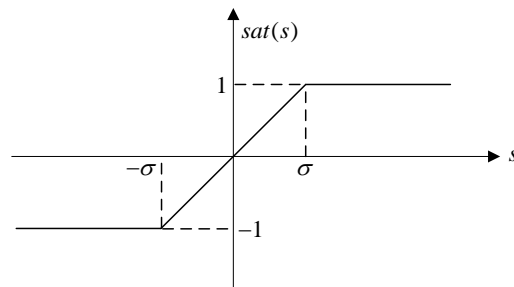


Figure 1.3: The saturation function

However, in some situations the system trajectory will converge to this boundary layer instead of the exact $s = 0$. The desired system dynamics, which is only achievable when $s = 0$, will not be available when the system states converge to this boundary. The

accuracy in this case will be deteriorated. As we can see, the boundary layer size σ is also a tuning parameter here. Too small boundary layer will still lead to chattering. On the other side, when we increase the size of this boundary layer, the accuracy will get worse. In practice, this parameter should be tuned based on extensive simulations before the right value is chosen in experiments.

An alternative approach to attenuate or eliminate the chattering problem is to use a high-order sliding mode system [23],[24]. High-order sliding mode (HOSM) system is an active research field, which started in the 1990s. In HOSM, the derivatives of the control input are discontinuous functions, which make the control input continuous after integral.

Conventional sliding systems are first-order sliding systems. In this type of sliding mode systems, the control input appears in the first-order derivatives of the sliding surface. This is observed in equation (1.8), where the control input appears explicitly in the first-order derivative of the sliding surface.

In r^{th} order sliding mode systems, the control input appears first in the r^{th} order sliding surface dynamics. A formal definition is given here since some subsequent chapters of the current work will be based on HOSM systems. This formal definition is from [18], and the details are found there.

A discontinuous differential equation $\dot{x} = f(x)$ with a smooth output function $s = s(x)$, which is understood in the Filippov sense. If

(1) time derivatives $s, \dot{s} \dots s^{(r-1)}$ are continuous functions of x

(2) the set $s = \dot{s} = \dots = s^{(r-1)} = 0$ is a nonempty integral set

(3) the Filippov set of admissible velocities at the r -sliding point of

$s = \dot{s} = \dots = s^{(r-1)} = 0$ contains more than one vector

the system is an r^{th} -order sliding mode system.

HOSM tries to not only drive the system state to the sliding surface, but also the high-order derivatives of the sliding variable s to zero. This further constraint of the high-order derivatives on the sliding surface will attenuate or even eliminate the chattering significantly, which is a very desirable feature of HOSM systems.

For the second-order sliding mode (SOSM) systems, there are various well-established approaches for sliding mode controller design, such as twisting algorithm [25], suboptimal algorithm [26], control algorithm with prescribed convergence law [27],

and quasi-continuous control algorithm [28]. The twisting controller is given here as an example to show the features of SOSMs [18].

The system dynamics may be described by

$$\dot{x} = a(x, t) + b(x, t)u \quad (1.15)$$

where $x \in R^n$ is the system state, and $u \in R$ is the control input. The output of this system which is the desired sliding surface is $s = s(x, t)$. Since it is a second-order sliding system, the control input should explicitly appear in the second-order derivative of the sliding surface. Suppose that it can be described by the representation

$$\ddot{s}(x, t) = h(x, t) + g(x, t)u \quad (1.16)$$

where $h(x, t)$ and $g(x, t)$ are some unknown smooth functions. Suppose that $h(x, t)$ and $g(x, t)$ satisfy the conditions

$$0 \leq K_m \leq g \leq K_M, \quad |h| \leq C \quad (1.17)$$

Then the twisting is given as

$$u = -[r_1 \text{sign}(s) + r_2 \text{sign}(\dot{s})], \quad r_1 > r_2 > 0 \quad (1.18)$$

The parameters r_1 and r_2 are selected to satisfy the inequality

$$\begin{aligned} (r_1 + r_2)K_m - C &> (r_1 - r_2)K_M + C \\ (r_1 - r_2)K_m &> C \end{aligned} \quad (1.19)$$

In this manner, finite time convergence to the manifold $s = \dot{s} = 0$ is proved in [18].

Controllers or observers designed in this way can drive the sliding variable as well as its first derivative to zero in finite time. The proofs of the convergence of those SOSMs are mostly based on the non-smooth Lyapunov functions [29] or the geometrical majorant curves [18],[30]. Sliding mode controllers with the sliding order higher than two are also proposed, such as nested sliding mode controllers [31]. The convergence of those HOSMs are only proved using homogeneity theory [32],[33], and no systematic way to obtain a Lyapunov function to show the convergence is available up to now.

While the first advantage of HOSMs is the attenuation of chattering, the second advantage is the accuracy in discrete control system implementations. It has been proved that for a sliding mode system with sliding order r and discrete sampling period T_s , the precision may be limited to T_s^r [34],[35].

It is easy to notice that the implementation of the twisting algorithms require the information of \dot{s} , even though only the sign of it is required. In discrete implementations this may be achieved by monitoring the change of s in two consecutive time steps [36]. An alternative approach is to use the sliding mode-based robust differentiators. The introduction of robust differentiators is given in the following subsections.

A special type of algorithm that has the feature of second sliding mode for relative degree one system is called super-twisting algorithm (STA) [37]. In this algorithm, the finite-time convergence of the sliding variable and its first derivative to zero are guaranteed. The control input will only be a function of the sliding variable itself, and no information about the derivative of the sliding variable is required. The STA is given as

$$\begin{cases} u = -\lambda_1 |s| \text{sign}(s) + w \\ \dot{w} = -\lambda_0 \text{sign}(s) \end{cases} \quad (1.20)$$

By proper selection of λ_0 and λ_1 , $s \rightarrow 0$ and $\dot{s} \rightarrow 0$ will be guaranteed in finite time. This special algorithm outperforms the conventional sliding mode controllers with regard to the smoothness of the control input.

1.3.2 Sliding Mode Observers

In state feedback control algorithms, some states are required but might not be available. For example, joint velocity may be required in impedance control algorithms. However, in the absence of tachometers this information is not available. Observers are used to estimate the state of plant and are widely used since it was firstly introduced by Luenberger in [38]. If the model of the plant is known, the Luenberger observer is able to reconstruct the states. This may be illustrated by the observable SISO system given by

$$\begin{aligned} \dot{x} &= Ax + B(u + d) \\ y &= Cx \end{aligned} \quad (1.21)$$

where $x \in R^n$ is the system state, $u \in R$ is the control input, $d \in R$ is the disturbance such as noise in the control input channel, and $y \in R$ is the output. In some cases some components of x are not available while they are required by some controllers. Luenberger observer may be designed as

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \quad (1.22)$$

Define the state observation error e as

$$e = x - \hat{x} \quad (1.23)$$

Then, the estimation error dynamics may be calculated as

$$\dot{e} = (A - KC)e + Bd \quad (1.24)$$

In the absence of input channel disturbance $d(t) = 0$, by proper selection of K , the poles of

$$\dot{e} = (A - KC)e \quad (1.25)$$

may be placed to guarantee that the estimation error asymptotically converges to zero. However, if the input channel disturbance $d(t) \neq 0$, the property that estimation error e asymptotically converges to zero may not be obtained. Robust state observers under plant model uncertainty should be designed in this case.

Sliding mode observers can guarantee the robust state estimation in the presence of model uncertainties. They replace the linear output feedback term in equation (1.22) with a discontinuous term $K \cdot \text{sign}(y - \hat{y})$. It has been proved that by proper choice of K , the estimated states will converge to actual ones in the presence of system uncertainty [15],[39],[40].

A class of sliding mode observers that can not only estimate the states of the plants, but also the unknown inputs of the plant was proposed in [14], [23], [40]–[42]. It is based on the STA. It was applied in mechanical system to identify the unknown dynamic parameters or external disturbances [43]–[45]. The limitation of this algorithm is its difficulty in selecting the gains to guarantee the convergence. Those gains are based on the bound of the uncertainty. An adaptive version of this algorithm may be developed to guarantee the convergence with bounded but unknown disturbances [46].

1.3.3 Robust Differentiators based on Sliding Mode Theory

$f(t)$ is a bounded real signal contaminated by measurement noise. For example, it is the joint position signal of robot manipulators. We are required to calculate the derivative of this signal (joint velocity). Conventionally, we can perform spectrum analysis of this signal to obtain details of its spectrum. Filters such as low-pass filter and band-pass filter may be applied to calculate the derivative of this signal with measurement noise. However, this should be done based on the assumption that the bandwidth of the system is known. Proper selection of the cut-off frequency of the filters should be done to ensure that the derived information is not distorted. Also, important information should not be

lost after using a filter. If that is not the case, we call the derivative information not “exact”.

Model-based filters, such as the Kalman filter may be applied to have an optimal estimation of signals with noise [47]. However, when the dynamic model of the system is inaccurate, the estimation based on Kalman filters could be inaccurate as well. Robust differentiator under system model uncertainty may be applied.

Robust differentiator based on sliding mode theory and independent of the dynamics of the original system was proposed in [16]. It is based on HOSM and the finite time convergence of the differentiator to original signal’s high-order derivatives is shown in terms of homogeneity theory [32],[33]. The general robust differentiator is given in Theorem 1 from [18].

Theorem 1: Assume that the input signal $f(t)$ is a function defined on $[0, \infty)$ consisting of a bounded Lebesgue-measurable noise without known features and base signal $f_0(t)$. The k -th derivative has a known Lipschitz constant $L > 0$. The real-time robust estimation of $\dot{f}_0(t)$, $\ddot{f}_0(t)$, \dots , $f_0^{(k)}(t)$ which are exact in the absence of measurement noise is given by robust differentiator defined as

$$\begin{aligned}
\dot{z}_0 &= v_0 \\
v_0 &= -\lambda_k L^{1/(k+1)} |z_0 - f(t)|^{k/(k+1)} \text{sign}(z_0 - f(t)) + z_1 \\
\dot{z}_1 &= v_1 \\
v_1 &= -\lambda_{k-1} L^{1/k} |z_1 - v_0|^{k/(k+1)} \text{sign}(z_1 - v_0) + z_2 \\
&\vdots \\
\dot{z}_{k-1} &= v_{k-1} \\
v_{k-1} &= -\lambda_1 L^{1/2} |z_{k-1} - v_{k-2}|^{k/(k+1)} \text{sign}(z_{k-1} - v_{k-2}) + z_k \\
\dot{z}_k &= -\lambda_0 L \text{sign}(z_k - v_{k-1})
\end{aligned} \tag{1.26}$$

If the parameters $\lambda_0, \lambda_1, \dots, \lambda_k > 0$ are properly chosen, then the derivatives of signal $f_0(t)$ are given as

$$z_0 = f_0(t), \dots, z_i = v_{i-1} = f_0^{(i)}(t), \quad i = 1, 2, \dots, k \tag{1.27}$$

It has been proved that the accuracy of the HOSM differentiators coincides with the Kolmogorov estimation. If the magnitude of the measurement noise is bounded by ξ , the accuracy of this robust differentiator reconstructed derivatives of the original signals is given by the inequality

$$\left| z_i(t) - f_0^{(i)}(t) \right| = O(\xi^{\zeta^{(k+1-i)/(k+1)}}) \quad (1.28)$$

This type of differentiator has been used in many applications, such as image edge detection [48] and robot dynamic model identification [49].

The problem with conventional robust exact differentiators is the proper selection of the switching gains λ_i , ($i = 0, 1, 2, \dots, k$). The choice of these gains is based on the Lipschitz constant of the original noisy signal [32],[50]. However, in practice, the characteristic of the measurement noise is not predictable, which makes the selection of those gains extremely difficult. Adaptive versions of this robust differentiator have been proposed to guarantee the convergence without understanding the measurement noise [48],[49].

1.3.4 Robot Dynamics and Identification

Robot manipulators are usually equipped with encoders to sense the motion in each joint. Sometimes the joint motion can be sensed indirectly. For example, the motion of a joint actuator, such as electric motor, hydraulic device, or pneumatic device is sensed. In a geared system, the joint motion may be obtained by multiplying the actuator motion by the transmission ratio.

The dynamics of a robot manipulator is the mathematical relationship between the sensed motion (and/or their corresponding derivatives, i.e., velocity and acceleration) and the applied actuation wrench. Accurate dynamic model of a robot manipulator is the prerequisite of analysis, validation and control of the robot manipulator. Advanced robot controllers are mostly based on robot dynamic model and the performance of the controllers depends on the accuracy of the model.

Based on the type of motion variables and the actuation wrenches used to describe this relationship, robot dynamics may be grouped into joint space dynamics and Cartesian space dynamics. Joint space dynamics aims to determine the relationship between the joint space rotation angle (for revolute joint) or displacement (for prismatic joint) and the joint actuation torque or actuation force. Cartesian space dynamics seeks to find a direct relationship between the applied generalized force and the displacement of the end effector in the Cartesian space. The joint space dynamics may be obtained in a straightforward manner by using the classical energy-based Euler-Lagrange formulation, or the computationally more efficient recursive Newton-Euler formulation. The Cartesian

space dynamics need some trivial kinematic transformation between the joint space coordinate and the Cartesian space coordinate. However, this transformation may sometimes lead to representation singularity. This problem can be avoided in the trajectory planning phase.

Accurate modeling of robot manipulator dynamics based on ideal physical formulation is a challenging task due to the structural complexity, especially when the uncertain joint friction component is taken into consideration. Several endeavors have been devoted into this area in the past to identify the dynamics as accurately as possible.

In [51] , the dynamics of a haptic device was identified by dismantling various components of the manipulator. The inertia parameters were measured. The problem with this type of problem is that they cannot include a complicated dynamic terms such as Coulomb friction into consideration, while such terms are the troublesome ones in the dynamics of manipulators. Some robot manufacturers provide approximate dynamic parameters based on the Computer Aided Design (CAD) models during the design process. The CAD models used to determine the dynamic parameters have the some limitations as the one mentioned before. They usually ignore the complex friction component in the dynamic model. Furthermore, they usually make assumptions on the geometrical dimensions and density of the materials.

Robot dynamic model identification by experimentation is an effective way to obtain accurate dynamic models. The complex friction components can be included into the dynamic model to be identified. Most of the dynamic model identification algorithms of robot manipulators take advantage of the property of linearity in parameters in the dynamic model. After including a carefully selected linear model for the friction component, the dynamic model of a manipulator may be described by the equation

$$\tau_i = Y_i(q, \dot{q}, \ddot{q})\Pi \quad (1.29)$$

where τ_i is the actuation torque of each joint, $Y_i(q, \dot{q}, \ddot{q})$ is the regressor matrix of the manipulator, which depends on the manipulator configuration (joint position q , joint velocity \dot{q} , joint acceleration \ddot{q}), and Π is the base dynamic parameter of the manipulator whose components are functions of the robot inertia tensors and the kinematic parameters. Equation (1.29) is also used in adaptive motion control of robot manipulators, and this representation may be formulated in a systematic way.

Using m (usually much bigger than the number of DOF of the manipulator) experimental tests, we have the set of measurements $W = [\tau_1 \ \tau_2 \ \cdots \ \tau_m]^T$, $Y = [Y_1 \ Y_2 \ \cdots \ Y_m]^T$. The ordinary least squares estimation (LSE) may be used to obtain an estimation of the unknown constant, given as

$$\Pi = (Y^T Y)^{-1} Y^T W = Y^+ W \quad (1.30)$$

where Y^+ is sometimes called the information matrix and plays an important role in tuning the accuracy of the identification result. Several robot dynamic parameter identification schemes have followed this routine, which differs from each other in the choices of analyzing the experimental data, ways to generate the input excitation torque, or how to deal with the measurement noise during experimentation.

Desired joint space trajectories are selected as the reference trajectories for the manipulator to track. These trajectories cannot be selected arbitrarily and should satisfy some conditions. First, system bandwidth has to be taken into consideration. The reference trajectories should be selected so as to be able to be tracked by the manipulator. Second, these reference trajectories should be selected to have a high level of system excitation, which is a prerequisite of any system identification algorithms. Usually the reference trajectory for each joint is selected as a combination of several sinusoids with various frequencies and amplitudes. Before real experimental implementation, these trajectories should be simulated based on the manipulator kinematics so that the workspace of the manipulator will not be violated.

In [52], the dynamic parameters of a PhantomTM haptic device was identified based on the linear in parameter property and LSE. The classical friction model, which includes terms for Coulomb friction and viscous friction for each joint was used in the dynamic model. The augmented dynamic model still meets the linear in parameters property. LSE was used to identify the dynamic parameters off line after several Position-Derivative (PD) based trajectory tracking experiments. Combinations of sinusoids with various frequencies and amplitudes were used as the reference trajectory for each joint. The joint velocities were estimated from joint angles by using a filter. Joint accelerations were reconstructed by using a strictly stable low-pass filter to avoid differentiating the velocities which will amplify the noise. The identified parameters were validated through experiments. It showed that it is able to predict joint torque at over 95% accuracy and

have an inertia matrix confirmed to be positive-definite within the device workspace. A limitation in this work is that the noise deteriorated joint encoder readings makes the information matrix noisy as well.

The ordinary LSE is known to be sensitive to noise. The measurement noise in the joint position, velocity, and acceleration makes the estimation biased. In [53], Simple Refined Instrumental Variable (SRIV) approach is applied to deal with the noisy observation matrix problem. In theory, this approach could give rise to statistically optimal estimation. The implementation of this approach is simple, and no noise model is required. What is more, this method is consistent even if the noise is colored. The acceleration calculation is avoided in the SRIV method. In addition, SRIV is less sensitive to the choice of the cut-off frequency compared to LS. Simulation and experimental results have been given to show the effectiveness of this approach.

Both LSE- and SRIV-based dynamic parameter identification algorithms require measurements of joint torques and joint positions. The joint velocities can be reconstructed through band-pass filtering of the joint position at a small sampling period. As mentioned above, the reconstruction of the joint velocities and accelerations are the main sources of the information matrix noise. This noise will lead to biased dynamic parameter identification. A new off-line dynamic parameter identification method named Direct and Inverse Dynamic Identification Models (DIDIM) which only requires the joint torque measurements is proposed in [54]. DIDIM is a close-loop output error feedback algorithm, which does not require joint position measurements. This method uses a simulated robot model with some selected initial dynamic parameters. Using the same structure for the control law, and the same reference trajectory for both the actual and simulated robot, the optimal dynamic parameters that minimize the error between the actual and simulated torque can be found after several iterations. The simplicity and advantage of this method are indicated by the dynamic parameter identification experiment carried out using an industrial robot.

While the above dynamic identification approaches depend on LSE or SRIV, some evolutionary computing algorithms also find their applications in this area. As pointed out in [55], LSE can have solutions such as local minima which are not optimal. Particle Swam Optimization (PSO) as an optimization method was used in [55] to estimate the inertia parameters of an industrial robot. It showed superior performance than the LSE

based identification algorithms. Another evolutionary computing algorithms, Genetic Algorithm (GA), is applied in [56] to identify the system model of a mechatronic system with unknown structure. Modified Genetic Algorithm Identification Method (MGAID) is used in [57] to identify the dynamic parameters of a two-link pneumatic artificial muscle manipulator.

All the above dynamic parameters identification algorithms attempt to complete the identification process offline. The experimental data is analyzed using the approaches mentioned above after several complete trajectory tracking experiments. Online dynamic parameter identification can be helpful to make the controller more adaptive to changing operating conditions. Changing payload of the robot is a very common situation in constraint motion control problem. However, this change requires that the dynamic model should be identified again using off-line dynamic identification algorithms. The online dynamic parameter identification schemes can be used to make the controller more robust and adaptive to these changes, which eventually makes the robots perform better.

In [58], an on-line identification algorithm based on batch adaptive control is proposed. The tracking error decreases because of the feedforward compensation from the dynamic model identified in the previous step. After several cycles of adaption, the real dynamic parameters can be obtained in finite time. Initial guess of the dynamic parameters is not critical here. An advantage of this identification algorithm is that it is implemented on line, which is quite appealing in practice.

1.3.5 Interaction Control

In our homecare project as described in the previous sections, the interplay of the manipulators and the human body has three phases. The first one is when the manipulator is moving in free space when no interaction between them is present. After the manipulator is commanded adequately close to the object, there comes the second phase, the transition phase. In this process, the interaction force will increase from zero to a limited and acceptable value. After steady contact with the object, the motion of the robot manipulator is fully constrained and the manipulator is in the third phase, the constrained motion.

Motion tracking in free space is a classical robot control problem and is dealt with from different perspectives. Simple Proportional-Integral-Derivative (PID) control algorithms may be applied directly in each joint while the nonlinearities in the dynamic model are considered as disturbances. Control algorithms in this category are typically decentralized control. The PID controller falls into this category can be applied to guarantee satisfactory trajectory tracking performance in the presence of disturbances. Due to its simplicity, most of industrial robots use this method for motion control. When stringent accuracy is required, model based control algorithms, such as inverse dynamics-based control, tries to cancel out the nonlinearities of dynamic models by feedback. After linearization, linear system design methodologies may be applied to design the controller for the nonlinear system. Model-based control algorithms are typically centralized control, while PID controllers ignoring the nonlinearities are called independent/decentralized control. Generally, model based control algorithms have better performance than the simple decentralized ones, even though they require dynamic parameters (at least nominal ones) and have more computing loads.

Smooth transition between free motion and constrained motion is critical in many applications. This is a new but important research area. Several control strategies with the aid of proximity sensors are proposed to guarantee the smoothness during this transition. The details are found in [59].

After the robot manipulator's interaction with an object, the robot control problem reaches the stage of constrained motion control, or interaction control, which is the main topic of the present work. The interaction force should be monitored and controlled to be within some acceptable range for safety in the homecare system. For this, a proper compliant behavior of the robot manipulator is required.

The task of drilling a hole in a piece of wood is shown in Figure 1.4. This is taken as an example here to introduce the concept of impedance control.

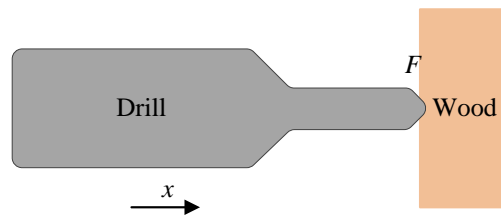


Figure 1.4: Task of drilling a hole in the wood piece

In this task, the interaction force should be monitored so that it will not damage the drilling tool, as well as the work piece. One intuitive way to control the interaction force is to depend entirely on the motion control strategy. Then, the magnitude of the interaction force is controlled by accurately controlling the movement of the drill without a force sensor to measure the interaction force. In order to avoid an excessive interaction force, the mechanical property of the wood, such as the stiffness should be known in advance. In addition, the motion control system in the drill need to be adequately fast and accurate, without jerk or overshoot. However, in practice, these two requirements are extremely difficult to meet. While the accuracy of the motion control system could be tuned accurately to some level, the precise description of the model of the interacted objects is unavailable in most applications. Even in well-planned industrial applications, such as a robot system for production line, there will be variations in the mechanical properties of the interacted components.

Interaction force provides a direct and effective description of the state of interaction. Due to the limitations of the above-mentioned force control algorithms in practice, new interaction control methodologies with the interaction force directly measured and included into the control loop have been investigated in the past decades.

Based on whether the interaction force is directly regulated or not, interaction force control schemes may be grouped into two categories, the direct force control and indirect force control.

In direct force control, in order to track the desired force as closely as possible, a force feedback loop with integral action on the force error is included in the position control loop. Because of the integral action, the interaction force can be tracked or regulated with zero steady state error. The classical hybrid position-force control algorithm falls into this category. Consider again the previously mentioned hole-drilling problem, but this time we are required to exert a desired normal force in the x direction, while moving along the y direction to change the location of the drill as shown in Figure 1.5. In this application, the robot manipulator task space is decomposed into two subspaces, the free space and the constrained space. In the x direction, direct force control schemes may be applied to regulate the interaction force, while in the y direction, classical motion control schemes may be directly used to change the location of the drill tip. However, in practice, this approach generally cannot provide satisfactory

performance because of its limitations. The task space cannot be decomposed into these two subspaces precisely. In addition, in the y direction, there will be an interaction force such as friction force, or elastic force when the drill tool is moving along this direction, because of the misalignment of the work piece.

Parallel force/position control is another direct force control algorithm. In this algorithm, a motion controller and a force controller are used simultaneously. However, the force controller has higher priority than the motion controller. Whenever there is conflict between these two controllers, the motion controller should yield to tolerate the position tracking error in order to regulate the force. However, as pointed out in [60], this algorithm requires accurate knowledge of the environmental dynamics in order to determine the reference velocity and the desired force level.

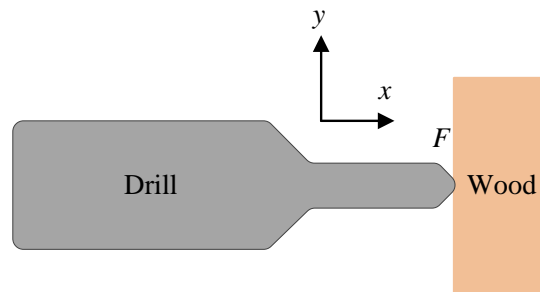


Figure 1.5: Application of hybrid position/force control

For indirect force control, the closure of a force feedback loop is usually not required. Controlling the interaction force by using a dynamic relationship between the interaction force and the motion would be adequate. If this dynamic relationship is described by stiffness, then this type of indirect force control method is called stiffness control. Stiffness control can be implemented either passively or actively. When the robot end effector is equipped with a compliant device, such as an elastic spring, then the interaction force between the end effector and the objects is determined by the stiffness of the spring. Compliant interaction may be achieved by selecting the stiffness properly. However, this parameter cannot be changed on line; thus, it is termed passive stiffness control. If on line tuning of this parameter is required, actively stiffness control should be applied, which could be implemented by static gravity compensation plus PD-based position control.

Another type of indirect force control algorithm is impedance control, which is also a subject of the present work. Mechanical impedance is defined as the quotient of the force over the resultant velocity in the Laplace domain. In impedance control, not only the

static property (stiffness) of the manipulator should be tuned according to the specific task, but also the dynamic properties, such as inertia and damping. In this sense, stiffness control mentioned above is just one special case of impedance control. The objective of impedance control is to use a proper control algorithm to change the mechanical impedance of the robot manipulator when viewed from the interacted object. Under impedance control, the manipulator dynamics can be described by

$$M_d(\ddot{X} - \ddot{X}_d) + B_d(\dot{X} - \dot{X}_d) + K_d(X - X_d) = F_e \quad (1.31)$$

where X is the actual position vector of the robot end effector, X_d is the desired position vector of the robot end effector, F_e is the interaction force between the robot end-effector and the interacted object, and M_d , B_d and K_d are the desired inertia, damping, and stiffness respectively.

In the Cartesian space, the robot manipulator has some inherent impedance if no specific feedback is used to shape it arbitrarily. However, this inherent impedance depends not only on the dynamic parameters of the manipulator, but also the manipulator configuration. Thus, the inherent impedance of the manipulator may be time-varying. If the inherent impedance is too big or too small in some configurations, sometimes it may be inappropriate for the interaction task. In this case, a feedback control system should be applied to actively change the mechanical impedance of the manipulator.

The interaction force during contact may be controlled indirectly by selecting proper desired impedance parameters. By selecting bigger impedance parameters, the rate of change of the interaction force can be made bigger. Also, the transition impact force between the free space and the constrained motion will be larger. Guidelines for selecting desired impedance parameters are application specific.

Controlling the impedance of robot manipulators is sometimes called impedance shaping. In the linear case, the impedance control problem may be solved by feedback control in a straightforward way. However, the dynamic models of robot manipulators are nonlinear and coupled, which complicate the impedance shaping problem. The situation becomes even worse when un-modeled dynamics, inaccurate dynamic parameters, and external disturbances are present. Robust and adaptive control algorithms may be used to shape the impedance in the presence of these disturbances and uncertainties.

Based on the ways of implementation, impedance control may be realized in a motion control-based approach, or a torque control-based approach. Motion-based approach is usually used in industrial robot manipulators where direct torque command to joint actuators is not allowed for safety concerns. We define a new position variable X_r , called reference trajectory, which is calculated based on the desired impedance equation

$$M_d(\ddot{X}_r - \ddot{X}_d) + B_d(\dot{X}_r - \dot{X}_d) + K_d(X_r - X_d) = F_e \quad (1.32)$$

The variable X_r denotes the reference trajectory for the motion control system of the manipulator. If the inner loop position system tracks this reference trajectory, then $X = X_r$ and the desired impedance shaping in equation (1.31) will be realized. This is a reasonable assumption, since in most robotic systems that only allow the user to command position or velocity, the bandwidth of this inner loop is usually much bigger than that of the outer force control loop. Because of this bandwidth difference, the motion tracking time is much shorter than the outer loop impedance shaping time. A survey on robust motion control system under uncertainty is found in [61]. An overview of adaptive motion control algorithms under uncertainty is found in [62]. A block diagram for this type of impedance control implementation is shown in Figure 1.6. Because this type of impedance control is realized by the motion control system, it is sometimes called admittance control.

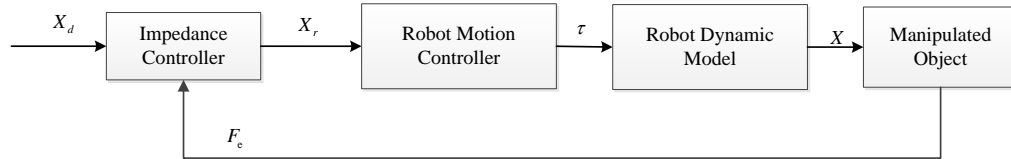


Figure 1.6: Motion-based impedance control (admittance control)

Torque-based impedance control is the algorithm that directly commands the control torque to the joint actuators to realize impedance control. A block scheme for this impedance control implementation is shown in Figure 1.7. This implementation is proved to be less robustness than the directly motion based impedance control implementation. However, it is easier to prove the stability and tracking performance of the torque based implementation approach.

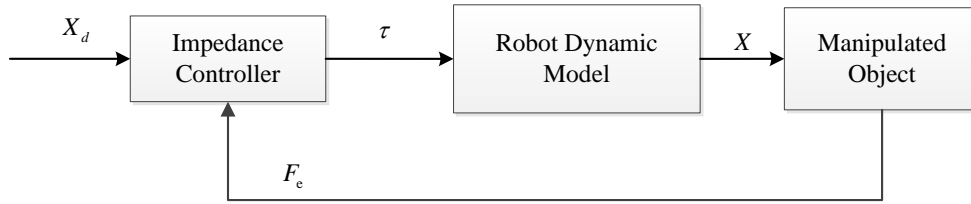


Figure 1.7: Torque-based impedance control

In the presence of dynamic parameter uncertainty, torque-based impedance control should be applied. Robust impedance control algorithms may be used to calculate the torque command. In [7],[63],[64] sliding mode control as a robust control algorithm is applied to shape the desired impedance in the presence of parameter uncertainty. The desired impedance model is defined as the sliding surface of the sliding control system. Control laws derived by the Lyapunov direct method are proved to be able to drive the system states onto the sliding surface. Due to the finite-time convergence property of sliding mode systems, the system state will stay in the sliding surface after it reaches this surface, in the ideal case. In this way, the desired impedance shaping is realized.

1.3.6 Force Control without Force Sensors

It is clear that in order to shape the desired impedance during interaction, the interaction force should be sensed. The interaction force sensing could be implemented by mounting a force sensor at the end-effector of the manipulator. This method is rather challenging and prone to error, and is not usually used in practice. Instead, conventionally the wrist force sensor mounted between the end effector and last link of the robot manipulator is applied. In this case, the interaction force is sensed indirectly since the sensed interaction force is the resultant force of the external interaction force, gravity and the dynamic force of the end effector. In order to determine the “real” interaction force between the end effector and the object with which it is interacting, the static gravity force and dynamic force should be calculated and compensated for. With an independent acceleration sensor at the end effector, sensor fusion technology was applied to determine this interaction force in [65]. Interaction force measured in this manner can be inaccurate even after compensation.

There are some other problems with the use of a force sensor in robot interaction control. It is usually the case that force measurement by a force sensor has significant noise [66]. Due to the high frequency nature of the sensor noise, proper filters should be

applied, such as a low-pass filter. The cut-off frequency of the filter should be higher than the bandwidth of the interaction force. However, there needs to be a compromise between the accuracy and robustness when selecting this cut-off frequency.

Another problem in using a force sensor is the limited bandwidth of the force sensor. The force sensors usually use strain gauges to detect the interaction force under loading. To sense the interaction force using strain gauges, this structure should have some flexibility. However, such a flexible device between the last link and the end-effector reduces the overall stiffness of the structure. Hence, the system bandwidth will decrease. Because of the reduced bandwidth of force sensing, the force control systems might lose stability when using the sensed force. Some strategies may be applied to stabilize the force control system. A commonly used method is to add a viscous feedback, which acts as damping to stabilize the force control system. However, in this case the system will become sluggish. Besides the above mentioned problems with force sensors in practice, they also need calibration because of its self-variance property under temperature or working condition change. Furthermore, mounting a force sensor at the interface of the robot end effector and the manipulated object (human body) is rather challenging.

In order to avoid problems related to force sensors in force control, much effort has been devoted into the development of force control systems without a force sensor, or sensorless force control. In sensorless force control, the interaction force is determined indirectly by using software or other hardware other than a force sensor [67]–[69].

In [70], a model based force estimation scheme is proposed. It uses an equivalent motion control system formulated based on the Luenberger observer. The essential idea is to consider the observer error dynamics as a damped spring-mass system driven by the environmental force. It was proved that the interaction force was proportional to the motion tracking error. Using this observer, the constant environmental forces could be estimated. Also, the noise-free joint velocity could be obtained by using the observed contact force information. However, only up to ramp style force functions could be tracked. The limitation of this method is that it cannot be generalized to other types of interaction force functions.

In [71], the interaction force was estimated from the torque measurements and accurate robot dynamic model using two different approaches. The equivalence of the two approaches was proved. The first approach used the linearity in parameters property

of robot dynamics to formulate the robot dynamic equations in a regressor form. It used a low-pass filter to obtain a filtered robot dynamic model to eliminate the need to measure the acceleration. Also, the inversion of the inertia matrix was avoided. Recursive least square estimation (RLSE) with forgetting factors was applied to obtain an online estimation of the interaction force. It was shown that a tradeoff between the smoothness of the estimated force information and the time lag in the force calculation was needed. The second approach was based on generalized momentum disturbance observer. Also, the acceleration and inverse of the inertia matrix were not required. However, compared to the first approach, the estimated interaction force information was not smooth. This is due to the fact that the recursive estimation scheme has the effect of smoothing the measurement noise in the joint torque. The joint torque measurement calculated based on the motor current is noisy in practice. Satisfactory performance may be achieved using both approaches if the dynamic model of the robot manipulator is accurate. This assumption serves to be the limitation of the work. In addition, it was assumed that the interaction force changed slowly to simplify the calculation, and this might not be practical. No physical force sensor is used to verify the accuracy of the estimated force information.

Disturbance observer (DOB) is an easy to implement robustness enhancement approach, which was proposed in the past decades and used in many fields. It is used to compensate for modeling uncertainties and external disturbance in robotic dynamic control, as shown in Figure 1.8. The disturbance comes from several sources, such as robot parameter uncertainty and external disturbances. When the robot comes into contact with the environment, the dynamic equation will change due to the external wrench. The disturbance observer will detect this and estimate the external disturbance. The difference between the output of the disturbance observer in free space and after contact will be the wrench resulting from the external contact force. In this way, the external force information can be estimated and used in force control algorithms such as impedance control.

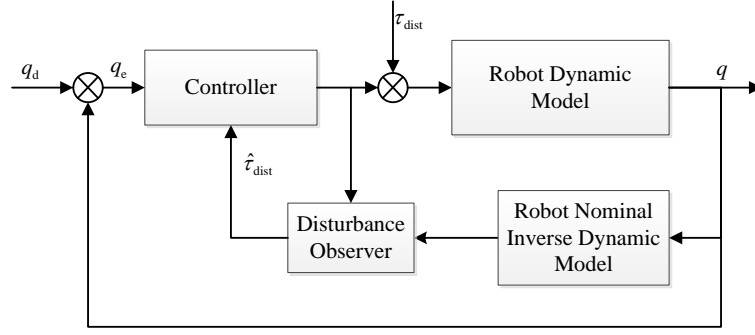


Figure 1.8: Disturbance observer in robot dynamics control

In [72]–[75], sensorless torque control in multi-degree of freedom robot manipulators is proposed based on an disturbance observer. Two disturbance observers are used in [72]. The first disturbance observer is used to compensate for un-modeled dynamics and external interaction force to achieve robust acceleration control (RAC), which is done by estimating the disturbance and compensating for it. The use of this disturbance observer will make the inner loop motion control system robust. In view of the disadvantages of using a force sensor in a force control system, a second disturbance observer called Robust Force/Torque Observer (RFOB/RTOB) is used to estimate the interaction force between the end effector and the environment. However, in that work, an accurate dynamic model of the robot is assumed. Even though some identification algorithms are proposed, dynamic parameters might be time-varying due to wear and tear.

In order to reduce the force estimation error due to the dynamic parameter mismatch, an online identification and compensation of the force estimating error method is investigated in [76]. In this work a single-DOF robot manipulator is used for bilateral teleoperation control implementations. The interaction force is estimated using similar method as in [72]. In addition, to compensate for the dynamic parameter error, online identification is proposed. The RLS algorithm with discontinuous projection mapping and conditional updating with forgetting factors can guarantee that the identified parameters are within some physical meaningful range. What is more, this algorithm can stop the online parameter updating when the manipulator contacts the environment to avoid parameter burst. Only one-DOF manipulator is considered in that work. Multi-degree of freedom case was considered in [77], where bilateral control was needed in microsurgery. The RLS algorithm with forgetting factor was used to obtain the dynamic parameter in free space. The estimated dynamic parameters were stored for contact force

estimation error correction during the contact phase. However, only a simulation result was given to show the effectiveness of the proposed method in one-DOF system.

Time delay estimation is a way to estimate the nonlinear and coupled dynamics of robot dynamics [78]–[80]. The advantage of this dynamic estimation algorithm is that it does not require the dynamic model of the robot, not even a nominal dynamic model. These papers point out that the previous disturbance observer-based estimation method suffers from dynamic parameter mismatch and the arduous identification work of the dynamic parameters. Time delay estimation (TDE) scheme has been used in [78] in two identical trajectory tracking problems. The first trajectory is in free space, while the second trajectory is in constrained motion. The output of the TDE is due to the presence of the external contact force. The external contact force is determined in this way. The limitation of the work is that the identity of the two trajectories is extremely difficult to guarantee. In addition, TDE by nature will introduce latency into the system and affect the stability of the system.

No complicated identification of the parameters is required by using DOB to estimate the interaction force. However, as pointed out in [81], even though the disturbance observer is simple to design and easy to implement, it is not suitable for industrial applications due to white Gaussian noise. The main drawback of using DOB to estimate the interaction force is its sensitivity to noise during velocity calculation. In order to attenuate this sensitivity, the estimated external force information is filtered by a low-pass filter. The cut-off frequency of this filter should be selected properly. However, as pointed out in [81], in the presence of white noise, the simple low-pass filter could not work effectively. A Kalman filter was used to complement the state observer and disturbance observer. Two observers based on Kalman filter, the Kalman-filter-based state observer (KFSO) and Kalman-filter-based DOB (KFDOB) were proposed to estimate the equivalent disturbance and external force. Taking the external reaction/action force as the state, the position, velocity and external force of a linear system could be estimated under noisy measurements. However, in this paper, the nominal dynamic model was used for both observers. As pointed out in [76], whenever there is dynamic parameter mismatch, there will be significant error in interaction force estimation. Also, a force sensor is required to have fast and accurate velocity estimation.

Joint velocity estimation under noisy measurement is an important research topic since determining the velocity information by differentiating will lead to noisy results [82]–[85]. Model-based joint velocity methods have prevailed in the past [86]–[92]. Simultaneous joint velocity and external interaction force estimation is important considering the limitations of the joint encoder readings and the need of interaction force information in an interaction control problem.

1.3.7 Teleoperation

There are situations where the human operator cannot be present to directly handle the task, such as nuclear waste management and deep sea exploration. Teleoperation is a technology aiming to help people to operate a manipulator from a remote location.

A teleoperation system has five key components. The first component is the human operator, who is located at the control site. The human operator could be an expert in a hospital control room or an engineer of a nuclear electrical power plant. Using the available feedback from the remote site, the human operator could make decisions and send out commands by manipulating a master device, which is the second component in a tele-operation system. The master device, which is also at the local site of the operator, is in its basic form somewhat like a joystick. The operator directly interacts with the master device to send out their command, while feeling the feedback interaction force from the remote task location, if the device is haptic. The third component of a teleoperation system is the communication channel, which is typically a TCP/IP or UDP transmission interface. It is used for information exchange between the remote site and the local site. The next component of a teleoperation system is the remote manipulator, also called slave device. It is located at the remote site and directly interacts with the manipulated object. It receives commands from the master device and attempts to execute them. Once it comes into contact with the last component of a teleoperation system, the motion of the manipulator is constrained and the interaction force should be monitored. The last component in the system is the teleoperated object, which is also called the task environment. The overall teleoperation system is shown in Figure 1.9.

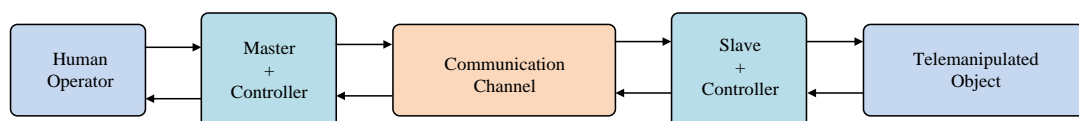


Figure 1.9: Schematic representation of a teleoperation system

Teleoperation systems have many applications, such as microsurgery, deep sea exploration, and space exploration. The primary problem with a teleoperation system is the stability under the unavoidable communication latency, which is in the range of 10ms to 4s. A number of solutions have been proposed to solve this problem, and they are found in the survey [93],[94]. Most of them lie in the framework of passivity concept or its variants.

Besides the stability under time delay of a teleoperation system, the performance of a teleoperation system is another concern. Traditionally, the performance of a teleoperation system may be measured by the match between the position/force of the master side and the position/force of the slave side. However, as pointed out in [95]–[101], in soft environments, such as human tissue, the discrimination of the impedance parameter change is more critical than just position and force match between the two sides. The sensitivity to the environmental impedance change under the stability constraint due to time delay has been formulated as an optimal control problem [100]–[102]. A feasible solution could be found by using standard linear and nonlinear optimization algorithms.

Since the slave manipulator is at the remote site and interacts with environment, the interaction between them should be monitored for proper operation. It may be desirable to have a compliant manipulator at the slave side so that a harmful force would not be exerted on the environment. Time delay in the communication channel further complicates this problem. Impedance control is an effective way to avoid excessive force in teleoperation applications [7], [103]–[105].

Selection of desired impedance parameters is based on two factors. The first one is the stability constraint. Bounded Impedance Absolute Stability (BIAS) was introduced to study the stability of a teleoperation system in [106]. BIAS is proved to be less conservative when compared with the traditional absolute stability criterion. The second one is the interaction task at hand. For example, when the manipulator contacts a rigid object, the manipulator should be controlled to be compliant. Accordingly, the stiff component in the desired impedance model should be selected to be small.

1.4 Contributions and Organizations of the Thesis

This thesis aims to solve some of the challenges in homecare systems. Specifically, it seeks to develop a compliant robotic manipulator interaction system which can perform

effectively under uncertainty, and without using a force sensor. The thesis makes several key contributions, which are listed below:

1. The dynamic model of a commercial robot manipulator is identified. The dynamic model of a simplified two-DOF system is derived analytically. The reference trajectory is selected for the experiments of dynamic model identification. The dynamic model identification is implemented off line to determine the nominal values of the dynamic parameters. On-line dynamic parameters identification is proposed to show the time-varying property of the friction coefficients. The accuracy of the dynamic parameters is evaluated by comparison between the calculated and measured joint torques. The dynamic model identified here is the basis for impedance control and external force estimation.
2. The external interaction force is estimated using sliding mode observers. The adaptive high-order sliding mode observers provide satisfactory simultaneous estimation of states and the interaction force.
3. Impedance controllers based on the estimated interaction force and states are proposed. In order to enhance the accuracy of impedance control, sliding mode based controllers are designed. Experimental results show that the accuracy of impedance control is superior to that of other impedance controllers.
4. The proposed sliding mode-based impedance control algorithms are implemented in a bilateral teleoperation system. Two applications are proposed. The first one is a bilateral impedance control scheme for teleoperation. Criteria for selecting desired impedance parameters are proposed. The second application in teleoperation is based on the H-infinity framework to study the robust performance under uncertainties in human operator and environmental impedances. It is formulated as a Linear Parameter-Varying (LPV) system.

The organization of the thesis is as follows:

Chapter 1 gives a brief introduction on the present homecare robotic project and points out some of the challenges in this field. The problem is formulated and the research objectives are outlined. The related past work is presented as a literature survey.

Chapter 2 concerns dynamic model identification of the robot manipulator that is used in this project. The experimental setup is redesigned in order to verify the

algorithms easily in the subsequent chapters. The dynamic model of the simplified two-DOF system is identified, which is the basis for the subsequent developments.

Chapter 3 discusses the estimation of the interaction force using observers. Different algorithms are proposed for interaction force estimation. Their performance is compared with regard to estimation accuracy.

Chapter 4 proposes the impedance controllers that use sliding mode control. The estimated external interaction force and the states of the manipulator are used in the impedance control algorithms. A new impedance control algorithm is proposed to improve the accuracy of impedance control.

Chapter 5 uses the sliding mode-based impedance controllers in teleoperation systems. Two applications are introduced in this chapter. The first one is the bilateral teleoperation under impedance control with task-dependent desired impedance parameters. The second one is the robust performance in the framework of gain-scheduling based control of LPV teleoperation system.

Chapter 6 summarizes the overall thesis. The possible future work is suggested.

Chapter 2: Modeling and Identification of Robot Dynamics

For a robot with revolute and/or prismatic (serial-link) joints, the dynamic model is a generalized relationship between the joint positions and the joint actuator torques. A mathematical description of this relationship is the basis for simulation, trajectory planning, control algorithm development and verification, and design.

In this chapter, the dynamic model of a commercial four DOF robotic manipulator, the Whole Arm Manipulator (WAMTM) from Barrett Technology, in the joint space and the Cartesian space is derived. As pointed out in the user manual provided by Barrett Technology, the dynamic parameters given by them are ideal ones and have unavoidable errors. Also, more complicated items, such as the joint friction, are not given in the manual. In addition, the reflected inertias of the transmission system which are almost at the same level as the link inertias are modelled approximately. They assume as well that there is no coupling between the rotor of an actuator and the base which it is mounted on. All these assumptions and approximations lead to an inaccurate dynamic model if we directly use the dynamic parameters provided by the manufacturer. For this reason, in this chapter, the dynamic parameters of our robot manipulator are identified through experiments.

To simplify the identification process, only the second and fourth joints of the original four-DOF system are considered. This simplifies the four-DOF model into a simple planar two-link manipulator. Also, to improve the accuracy of the identified dynamic model, the viscous and Coulomb friction components, as well as the interaction between the actuator rotor and the base are considered in the dynamic model. With the explicit inclusion of all these dynamic effects, the dynamic model is identified experimentally. The identified dynamic parameters are verified by comparing the torque outputs calculated using the output from inverse dynamics with the actual actuating torque. Also, online dynamic identification is proposed and implemented to show the time-varying property of some parameters. To deal with the uncertainties in the friction components, a neural-network-based compensator is applied to compensate for it.

2.1 Introduction

The robot manipulator that is used in the present project is a four-DOF serial link manipulator, as shown in Figure 2.1. It is a lightweight and back-drivable robotic manipulator, which is suitable for human-robot interaction applications. This manipulator can be easily integrated with Barrett Hand from the same manufacturer to form a mobile manipulation system.



Figure 2.1: Whole Arm Manipulator (WAM) from Barrett Technology

The DH parameters of this manipulator are given in Table 2.1. The dynamic parameters for the manipulator provided by the manufacturer are derived from CAD models. A schematic representation of the arm is given in Figure 2.2.

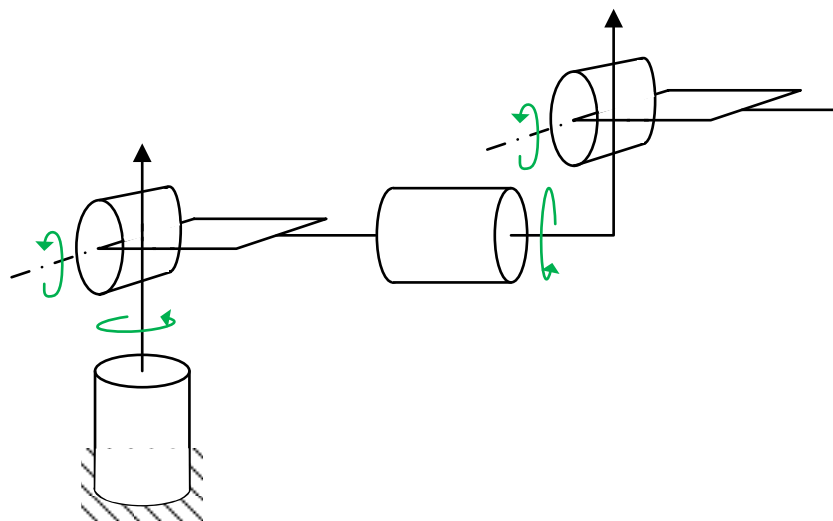


Figure 2.2: Schematic representation of WAM

Table 2.1: DH parameters of WAM

Joint i	a_i (m)	α_i (rad)	d_i (m)	q_i (rad)
1	0	$-\pi/2$	0	q_1
2	0	$\pi/2$	0	q_2
3	0.045	$-\pi/2$	0.55	q_3
4	-0.045	$\pi/2$	0	q_4
Tool frame	0	0	0	

In each joint, there is an optical encoder with position sensing resolution $2\pi/4096$. These encoders are relative (incremental) encoders. Hence, prior to each task of absolute position sensing, the manipulator has to be returned to its home position.

The joint actuators are DC motors, whose dynamic information is not provided by the manufacturer. However, other users of this manipulator have observed that the bandwidth of the DC motors of the manipulator is much larger than that of the main mechanical arm. Hence, the dynamics of the actuators may be ignored in a manipulator model. The joint actuators are assumed to be ideal current-controlled DC motors. WAM has an internal PC running a Linux-based real-time operating system with a loop sampling frequency of 500Hz. All the hardware, including the joint optical sensors, the joint DC motors, the control pendant and the display pendant are all connected to the internal PC by the use of the Controller Area Network (CAN) bus, which is a fast and reliable field control bus initially developed by BoschTM, as shown in Figure 2.3.

The manufacturer provides some ready-to-use source codes, which help users who are not particularly concerned about dynamic control accuracy when using the arm. For example, the sample code provides an interface to command the robot manipulator to some specific location with a trapezoidal trajectory profile. This manipulator can even be commanded to exert some desired interaction force/torque in the Cartesian space onto a manipulated object. However, the control algorithms in the provided codes are developed based on the dynamic model derived from a CAD model of the manipulator. Unavoidably, there will be deviations between the actual mechanical parts and the CAD model during the manufacturing process. What is more, the friction parameters in each joint are ignored in the provided control algorithms. For example, for Cartesian space

tracking control, the Cartesian space reference trajectories are converted into the corresponding joint space reference trajectories. These joint space reference trajectories are tracked by the PID controllers in the joints. The PID controllers do not take the dynamic coupling between links into consideration. The coupling is considered as a disturbance to the trajectory tracking system in each joint. Even though properly selected PID parameters can guarantee the steady state tracking error to be zero, it cannot guarantee the intermediate tracking performance, especially when the motion in each joint is fast. However, due to its simplicity, this control algorithm is widely used in industrial robots. One probable reason is the proprietary issues of the manufacturer. Since the main objectives of the present work concern dynamics and control of robot manipulators, we need to be able to send torque commands to actuators directly. Fortunately, Barrett Technology does allow users to perform this command.

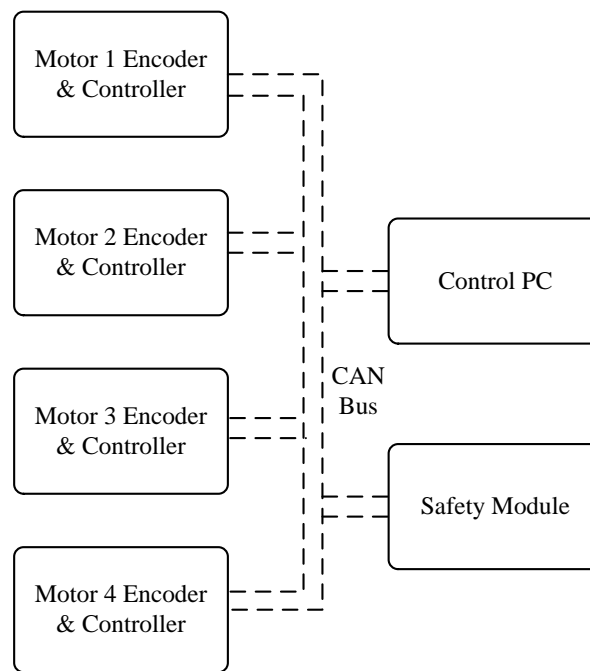


Figure 2.3: Hardware communications in WAM

Model-based control algorithms have better accuracy during fast motions. However, model-based control algorithms require a dynamic model for the robot manipulator. While some manufactures provide CAD-based dynamic parameters to the users, most of them require the users to identify the dynamic parameters by themselves. The identified dynamic parameters are the basis for the design of model-based controllers. This justifies the need to identify the dynamic model of a robot manipulator. Before the dynamic

identification experiments are carried out, we should develop an experimental setup with a proper user interface and sufficiently powerful processing capacity. This aspect is presented next.

2.2 Development of the Experimental Setup

Model-based control of a robot manipulator starts with a model of the manipulator and based on that, determining a mathematical relationship between the current state of the manipulator and the torque command to be delivered by the joint actuators in the next step. Fast prototyping is important in verifying the control algorithms of robot dynamic control.

For some commercial robots, the manufacturers provide their own Integrated Development Environments (IDE), such as RobotStudio from ABB. These IDEs provide some available libraries to the user, which may be used to verify their own algorithms. However, these IDEs are designed specifically for the manufacturers' products only. The users may not have full access to everything they need for the verification of control algorithms. For example, most industrial robots do not allow the user to send torque commands to their actuators.

Some generic IDEs, such as Microsoft Robotics Studio, are able to provide a uniform platform for different types of robots. Researchers and engineers who use such platforms may contribute enhancements and test algorithms to them. Such a platform may be used, for example, to compare the performance of different control algorithms. Similarly, the Robot Operating System (ROS) provides an open-source system for multi-user collaboration. Many software libraries are contributed by researchers and most of them are open-source. ROS is based on a Linux system, which also provides an interface to some scientific computing software such as MatlabTM/SimulinkTM.

The disadvantages with these generic IDEs include the difficulty of using them in conjunction with other software packages. For the implementation of robot dynamic control algorithms, complicated mathematical relationships have to be programmed. Considerable symbolic calculations may be needed in robot dynamic control. Software packages of scientific computing such as Matlab/Simulink and Maple are suitable for this purpose. Even though there is an interface for communication between the generic IDEs and Matlab/Simulink, yet the real-time requirement may not be easily satisfied.

The real-time operating system in the internal PC of WAM does not have a graphical interface. The user should login to the internal PC by using another PC that is connected to the same network. Additionally, the internal PC has very limited processing capacity.

In view of the issues mentioned above, a real-time control system with powerful symbolic calculation capacity is desirable for the present project. xPC target technology based on Matlab/Simulink from Mathworks may be applied here. Also, an external PC with greater processing capacity is integrated to bypass the internal PC of WAM. The software system is redesigned as well. The algorithms provided by the manufacturer are not useful anymore, and are redesigned. This aspect is discussed in the following subsections.

2.2.1 Hardware of xPC Target System

xPC Target is a fast prototyping technology based on Matlab/Simulink of Mathworks. It is widely used in real-time control and hardware-in-the-loop applications. The xPC Target setup comprises two PCs communicating with each other through an Ethernet, as shown in Figure 2.4. The two PCs, named as host PC and target PC, have different functions.

The host PC is a regular PC running the Windows operating system and has some compilers such as Microsoft Visual Studio. Control algorithms may be built in the same way as regular numerical simulation models using available or customized blocks of Simulink. Once the model is ready, it may be compiled into C/C++ automatically by simply clicking the “build” icon. The quality of the compiled C/C++ may be controlled in the configuration panel. Once it is compiled, it is downloaded to the target PC through the Ethernet. After that, the target PC takes over the task.

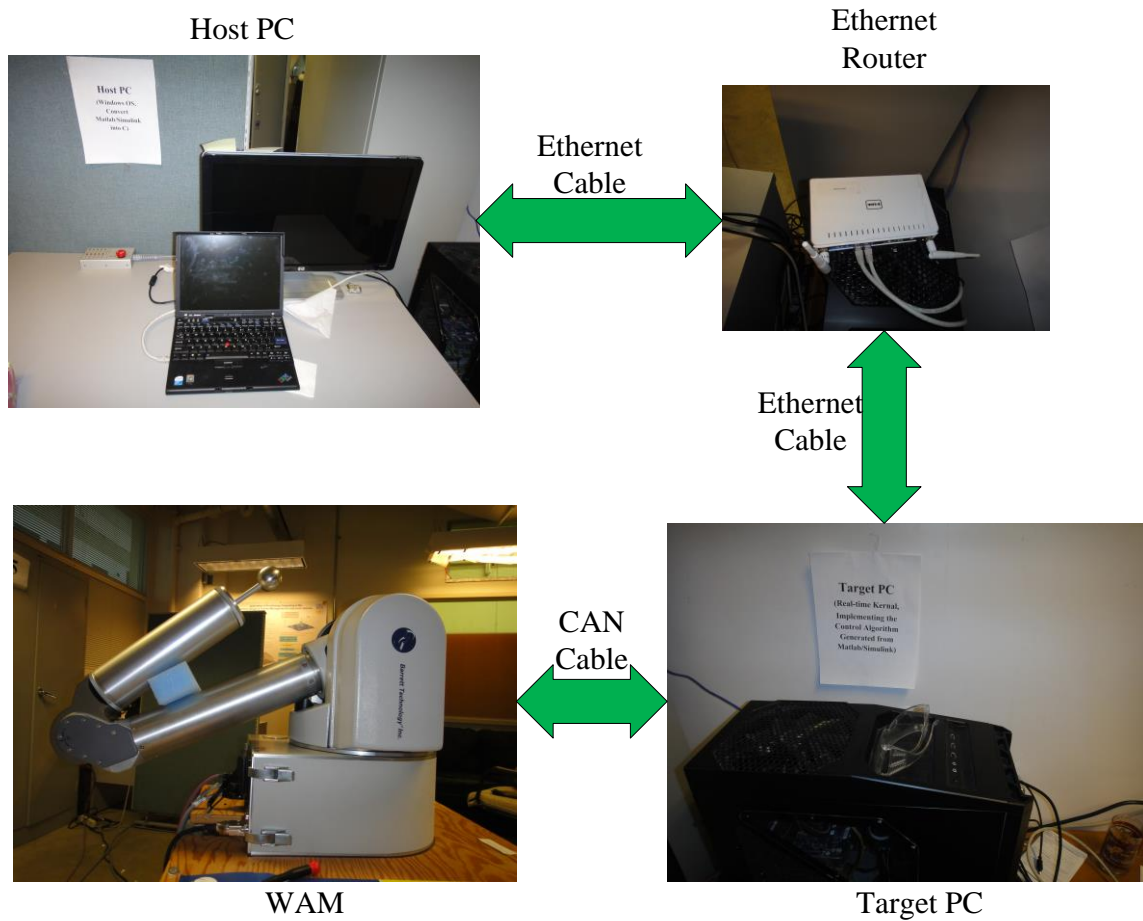


Figure 2.4: xPC Target-based experimental setup

The target PC has a real-time kernel generated by Matlab/Simulink. The target PC can boot from this real-time kernel. Due to the small size of this kernel and the real-time requirement of the control system, it does not have drivers for some hardware. Only hardware certified by Matlab/Simulink has the available driver support. However, experienced users may perform the necessary coding to create their own drivers. Due to the limited hardware support, the Ethernet card used for communication between the host PC and target PC has to be selected carefully. The target PC is the hardware which actually implements the control algorithms and provides the input and output interface.

Only some specific CAN cards are supported by the real-time kernel in the target PC. A supported CAN card from SoftingTM is installed in the target PC for CAN message sending and receiving.

2.2.2 Software of xPC Target System

Since the internal PC of the manipulator is disabled, the software system in the internal PC cannot be used any more. Thus, a new software system based on xPC target platform is designed. The new system includes an input and output interface, timing, and CAN bus initialization. Corresponding SimulinkTM blocks from the xPC target toolbox are used for obtaining the joint positions and sending out the joint torque commands that are calculated by the dynamic control algorithms.

The first step is to initialize the CAN bus by sending some CAN messages using the CAN initialization block, as shown in Figure 2.5. These CAN messages include CAN bus initialization, torque limits of the joint actuators, and joint velocity limits. For the purpose of safety, the CAN messages for setting the actuator torque limits and velocity limits are used during the initialization phase.

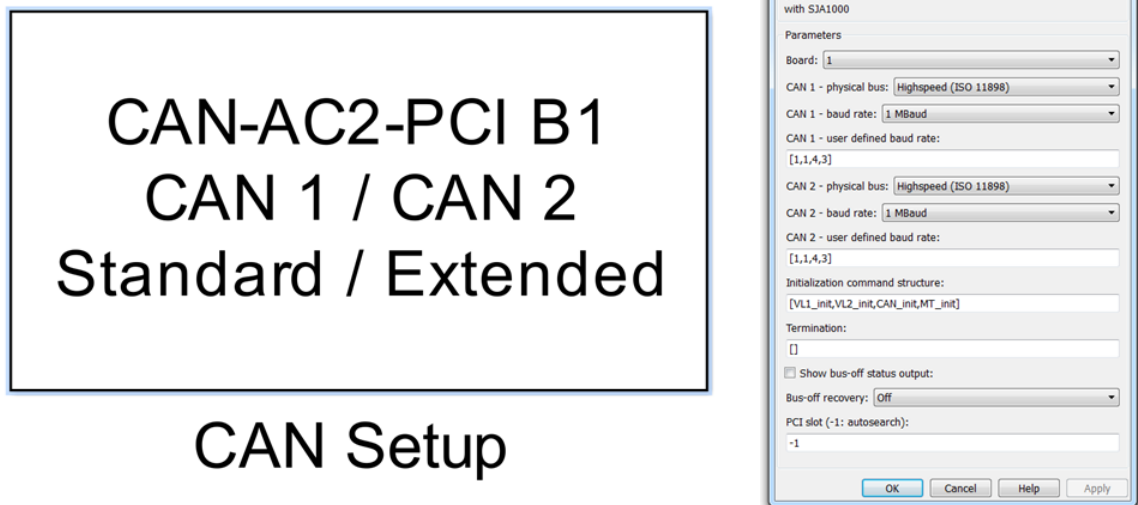


Figure 2.5: CAN initialization block

After the CAN bus is initialized, it is ready to transmit CAN messages between each node within CAN. The external PC will send out a group message to each joint actuator node. This group message is a request to each node to report their current joint position to the external PC. The property value from the manufacturer user manual is 48. The group ID (including the setting property) is 1024. The corresponding block for this process is given in Figure 2.6.

The joint nodes will response to this request and report their present joint positions in the form of CAN messages to the external PC. The corresponding process is given in

Figure 2.7. Some constant matrices are used to convert the corresponding motor positions into joint positions. In this way, the joint positions will be available to the dynamic control algorithms.

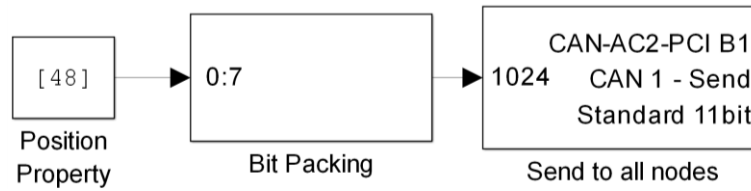


Figure 2.6: Joint position report request

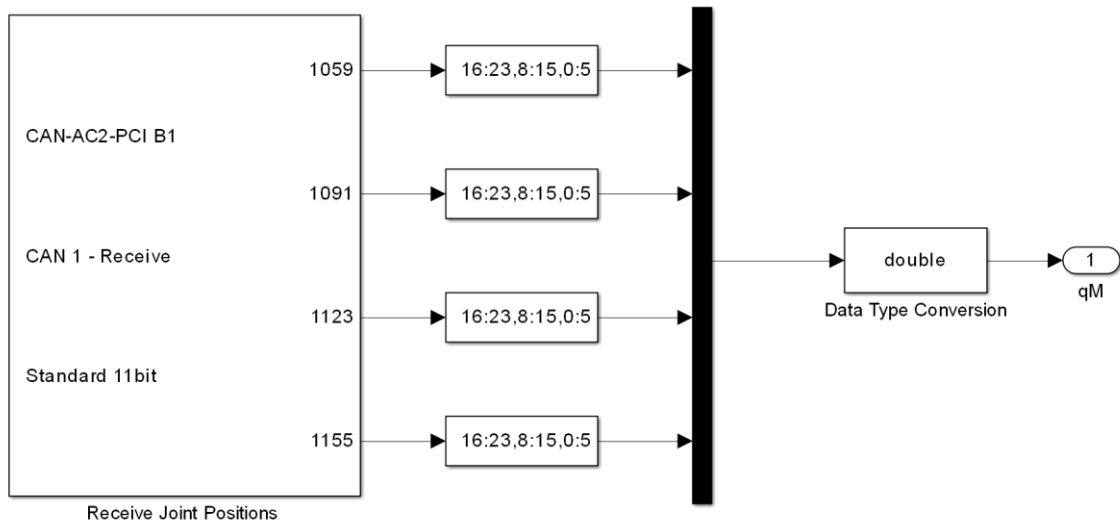


Figure 2.7: Reporting joint positions

Based on the available joint positions, the dynamics control algorithms calculate the corresponding joint torque command for the next step. After calculation, it will broadcast the torque command in the form of group CAN messages. All joint actuator nodes are in this group and will listen to this broadcast. However, there is a safety panel in this WAM. For safety purposes, if the “Active” button in the control panel is not activated, a nonzero torque command cannot be sent to WAM. Otherwise, the safety module will be triggered and put WAM into the “Idle” mode. Thus, the current mode should be checked before sending out the torque command. This is shown in Figure 2.8 and Figure 2.9. A conditional switch should be done here to avoid the problem mentioned above. This process is illustrated in Figure 2.10. If the WAM mode is zero (the “Active” button in the control panel is not activated), zero torques will be sent to each joint. Otherwise, the actual torque calculated by the dynamics control algorithms will be sent out. After all these have been done, the torque command will be sent out in the form of CAN

messages, as shown in Figure 2.11. The flowchart indicating the software system is given in Figure 2.12.

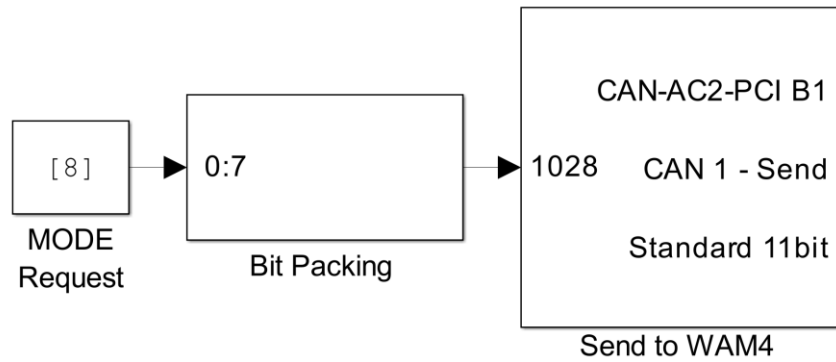


Figure 2.8: WAM mode check request

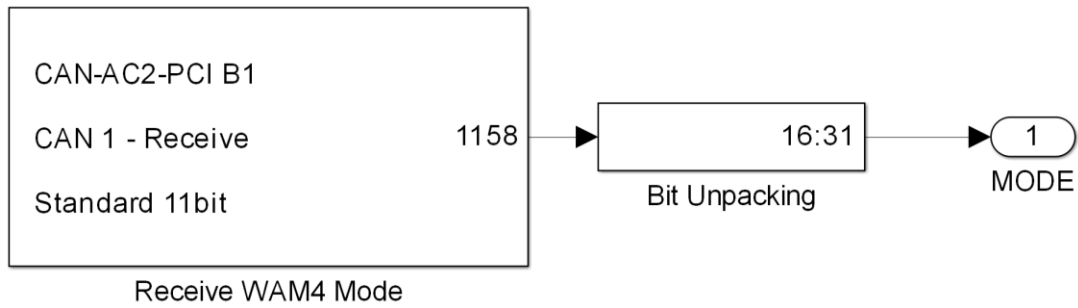


Figure 2.9: WAM mode report

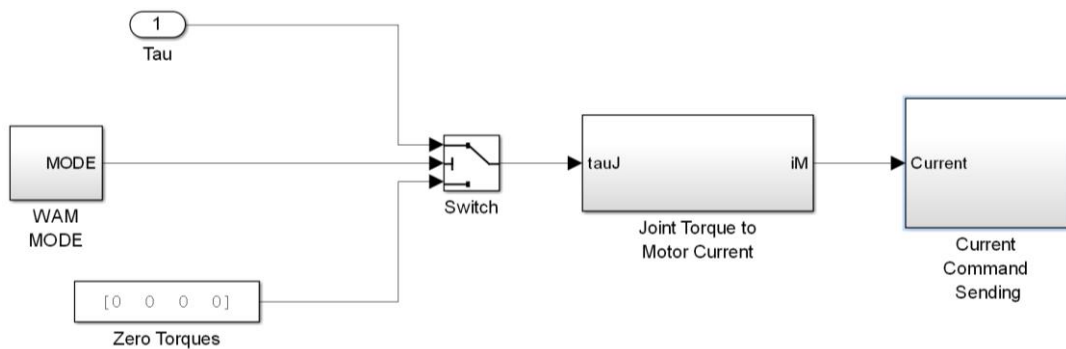


Figure 2.10: WAM torque command sending switch

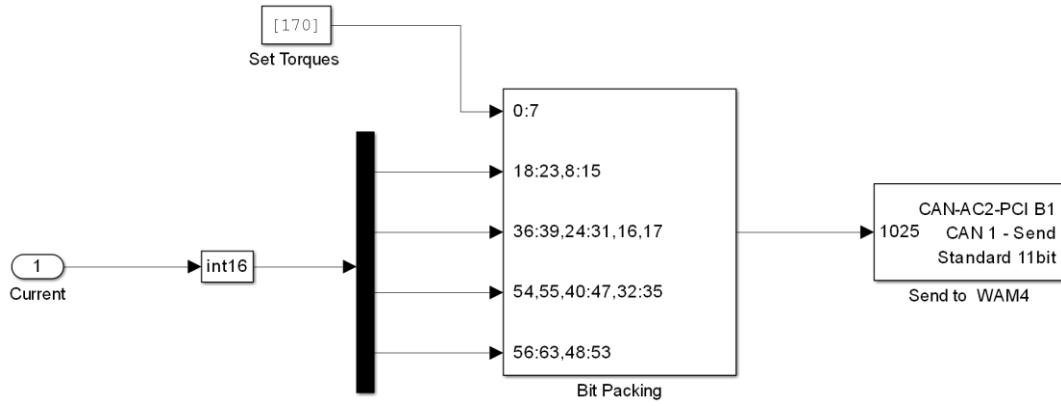


Figure 2.11: WAM torque command sending

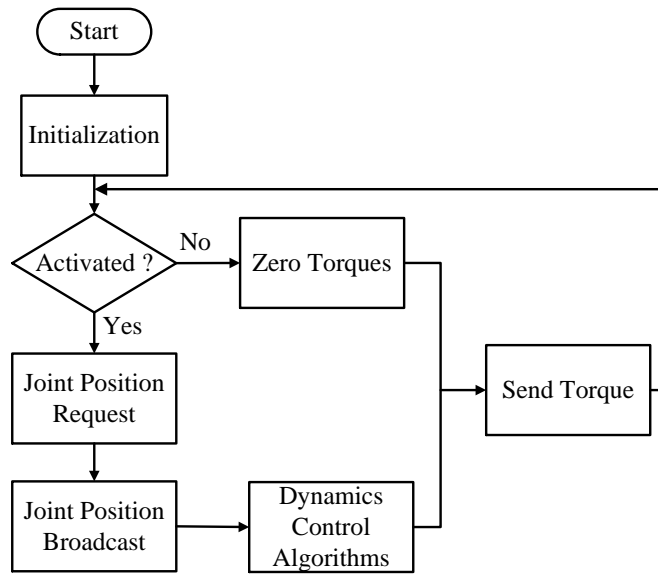


Figure 2.12: Flowchart of the software system

2.2.3 WAM Simplification

In model-based dynamic control algorithms of robot manipulators, researchers typically concentrate on a specific control challenge while ignoring the other issues that might complicate the problem. For example, redundancy and singularity are common issues that would complicate the dynamic control problem of the present study. In our specific application, the WAM system is a four-DOF manipulator. It becomes a redundant manipulator if only the translational motions of the end-effector are used. In order to avoid redundancy, we can lock one or two joints of the manipulator. Even though some commercial packages are available to calculate the symbolic representations

of the dynamic model, the system complexity increases with the number of degree of freedom of the robot manipulator.

The four-DOF WAM is simplified into a two-link planar manipulator by locking two joints out of four. This may be done by locking joint 1 and joint 3. Subsequently, the dynamic model of this simplified planar two-link manipulator is identified.

The first joint of the manipulator is manually locked at its home position. This makes body attached frame of link 1 coincide with the world coordinate frame. Also, joint 3 is locked so that the WAM is simplified into a planar two-DOF arm, which greatly simplifies the symbolic calculations. The simplified two-DOF manipulator schematic representation is given in Figure 2.13.

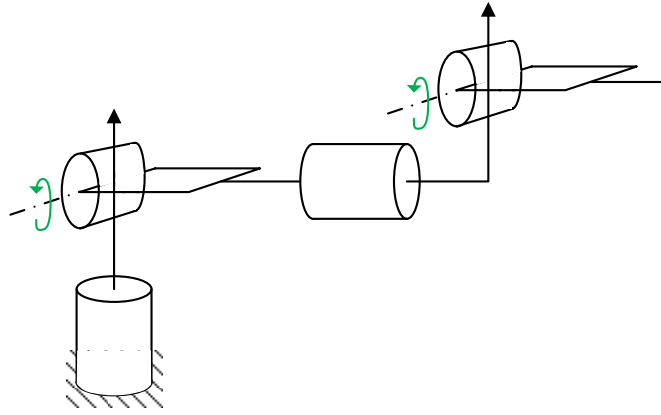


Figure 2.13: Schematic representation of simplified WAM

The kinematics of the simplified planar two-link manipulator is shown in Figure 2.14. In the simplified two-link manipulator, joint 1 is the second joint in the original WAM system, while joint 2 is the fourth joint in the original WAM system. However, for simplicity, we use the terms joint 1 and joint 2 to describe the two joints in the simplified manipulator. The corresponding coordinate frame attached to each link is shown in Figure 2.15. Special attention should be paid to the definition of the world coordinate frame. Also, the zero positions of each joint are positions where the two links are fully extended in the vertical z_0 direction. We use q_1 and q_2 to describe the motion of joint 2 and joint 4, respectively, of the actual WAM system. $q = [q_1 \quad q_2]^T$ is the vector of joint motions which are used in the rest of this thesis if not stated otherwise. The rationale for this nomenclature is the convenience of description of the following complex dynamic equations.

Since kinematic uncertainty is not as serious as dynamic uncertainty, we use the DH parameters provided by the manufacturer, which is a convention in most dynamics control problems.

Some kinematic definitions are given in equation (2.1) – equation (2.4).

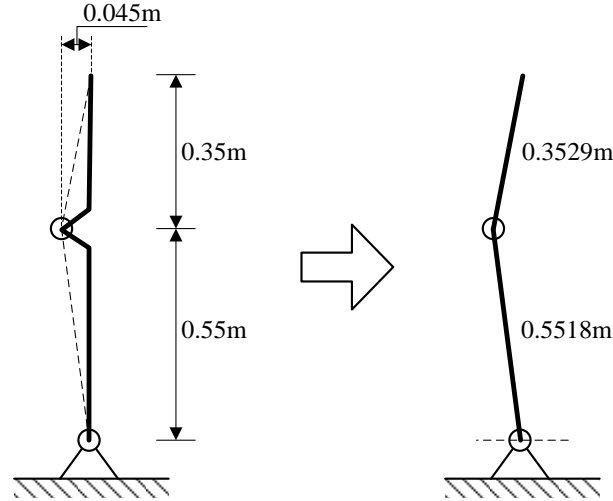


Figure 2.14: Simplified two-link planar manipulator

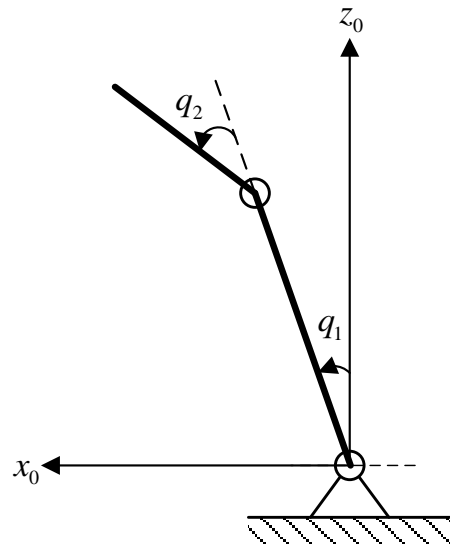


Figure 2.15: Coordinate frames of simplified two-link planar manipulator

$$x = l_1 \cdot \sin q_1 + l_2 \cdot \sin(q_1 + q_2) \quad (2.1)$$

$$z = l_1 \cdot \cos q_1 + l_2 \cdot \cos(q_1 + q_2) \quad (2.2)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (2.3)$$

$$J = \begin{bmatrix} l_1 \cdot \cos q_1 + l_2 \cdot \cos(q_1 + q_2) & l_2 \cdot \cos(q_1 + q_2) \\ -l_1 \cdot \sin q_1 - l_2 \cdot \sin(q_1 + q_2) & -l_2 \cdot \sin(q_1 + q_2) \end{bmatrix} \quad (2.4)$$

There is a rotation limit for each joint, as given by:

$$-2.0 \text{ rad} \leq q_1 \leq 2.0 \text{ rad}, -0.9 \text{ rad} \leq q_2 \leq 3.1 \text{ rad} \quad (2.5)$$

The workspace is defined by

$$-0.8519 \leq x \leq 0.5486, -0.1325 \leq y \leq 0.3655 \quad (2.6)$$

This has to be considered in the position command sent by the haptic device.

In order to make the simplification valid, a mechanical device and the control software are used to keep both joint 1 and joint 3 in their zero positions. However, adding mechanical devices will change the dynamic parameters provided by the manufacturer. Even though these dynamic parameters will be identified in the following sections, it is better to keep the WAM in its original state. Other members of our laboratory perform their experiments using the same platform with the algorithms provided by the manufacturer. We use strong adhesive tapes to fix the positions of joint 1 and joint 3, as shown in Figure 2.16 and Figure 2.17 respectively. These tapes are light-weight and their effect on the original dynamic model may be ignored. Besides this, in the software system, a proportional controllers plus gravity loading compensation are used to further guarantee fixing of joint 1 and joint 3. This is discussed under the symbolic calculation in section 2.2.4.

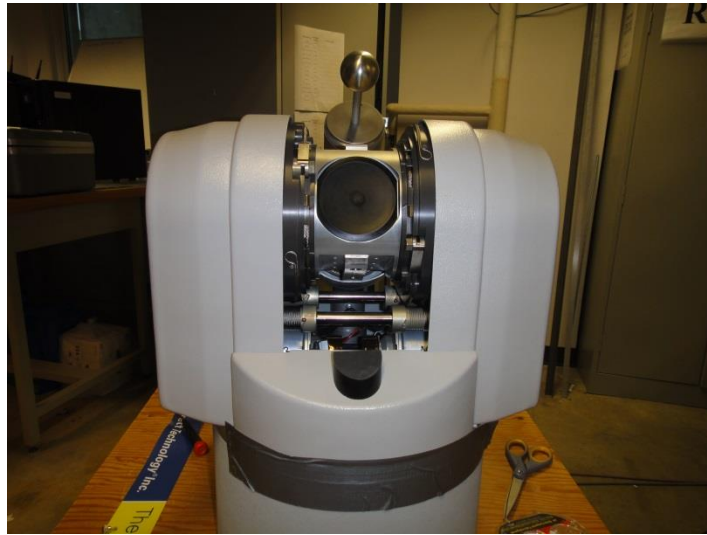


Figure 2.16: Simplified WAM system (joint 1)

2.2.4 Fixing Joints 1 and 3 by Software

As discussed in Section 2.2.3, adhesive tapes are used to fix joint 1 and joint 3 in their zero positions so that the simplified system may be modelled as a planar two-link manipulator. The validity of this assumption depends on the actual position of joint 1 and joint 3 during experiments. Besides the tapes used in joint 1 and joint 3, an additional approach implemented in the software may be used to enhance the performance. A simple proportional controller with gravity loading compensation is used in joint 1 and joint 3, which may be described by

$$\tau_1 = g_1(q) - K_{p1} \cdot q_1 \quad (2.7)$$

$$\tau_3 = g_3(q) - K_{p3} \cdot q_3 \quad (2.8)$$

where τ_1 and τ_3 are the joint torques sent to joint 1 and joint 3, respectively, q is the joint position vector, $g_1(q)$ and $g_3(q)$ are the corresponding component of the gravity loading, and K_{p1} and K_{p3} are the corresponding proportional gains of joint 1 and joint 3. The values of K_{p1} and K_{p3} should be selected properly in order to avoid generating excessive joint torques. A too large joint torque command will trigger the WAMTM hardware self-protection system to put WAM into the “Idle” mode. Note that q_1 and q_3 are the joint position variables of the original four-DOF WAM, not the simplified two-DOF one.



Figure 2.17: Simplified WAM system (joint 3)

2.3 Dynamic Model Derivation

2.3.1 Joint Space Dynamic Model

There are generally two ways to derive the dynamic model of a robot manipulator systematically. The first one is based on the Lagrange formulation. It seeks to determine the Lagrangian of the dynamic system, which is the difference between the kinetic energy and the potential energy. Using the Lagrange equation, the dynamic model may be determined in a straightforward way. The second approach is based on the Newton-Euler formulation. The dynamic model is derived in a recursive way, which is computationally more efficient. Both methods result in identical dynamic models. The free body diagram of one individual link i is given in Figure 2.18.

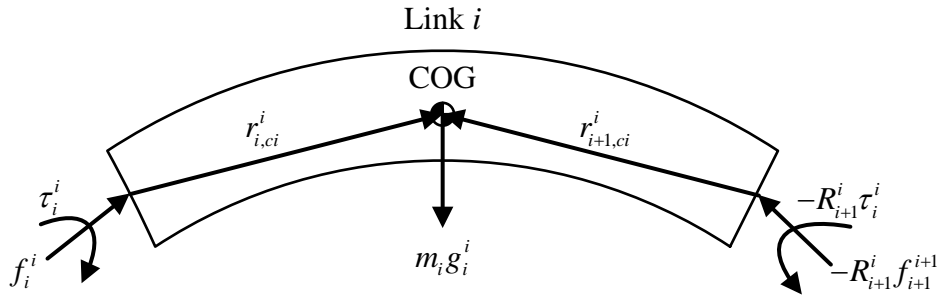


Figure 2.18: Free body diagram of link i

where each variable has the following physical meaning:

m_i - mass of link i

g_i^i - gravity vector, represented in link i body attached frame

$r_{i,ci}^i$ - vector pointing from joint i to center of mass of link i , represented in body frame attached to link i

$r_{i+1,ci}^i$ - vector pointing from joint $i+1$ to center of mass of link i , represented in body frame attached to link i

f_i^i - force exerted by link $i-1$ on link i , represented in body frame attached to link i

τ_i^i - torque exerted by link $i-1$ on link i , represented in body frame attached to link i

f_{i+1}^{i+1} - force exerted by link $i+1$ on link i , represented in body frame attached to link $i+1$

τ_{i+1}^{i+1} - torque exerted by link $i+1$ on link i , represented in link $i+1$ body attached frame

R_{i+1}^i - transformation matrix from link $i+1$ body frame to link i body frame

Let the following variables represent the velocity and acceleration of link i

a_{ci}^i - center of mass acceleration of link i , represented in link i body frame

a_{ei}^i - acceleration of the end of link i , represented in link i body frame

ω_i^i - angular velocity of frame i , represented in link i body frame

α_i^i - angular acceleration of frame i , represented in link i body frame

Following the standard recursive procedures of the Newton-Euler formulation, there are two steps to derive the dynamic model of a robot manipulator.

(i) Forward Kinematics Recursion

$$\omega_i^i = (R_i^{i-1})^T (\omega_{i-1}^{i-1} + z_{i-1}^{i-1} \cdot \dot{q}_i) \quad (2.9)$$

$$\alpha_i^i = (R_i^{i-1})^T (\alpha_{i-1}^{i-1} + z_{i-1}^{i-1} \ddot{q}_i + \dot{q}_i \omega_{i-1}^{i-1} \times z_{i-1}^{i-1}) \quad (2.10)$$

$$a_i^i = (R_i^{i-1})^T a_{i-1}^{i-1} + \alpha_i^i \times r_{i-1,i}^i + \omega_i^i \times (\omega_i^i \times r_{i-1,i}^i) \quad (2.11)$$

$$a_{ci}^i = a_i^i + \alpha_i^i \times r_{i,ci}^i + \omega_i^i \times (\omega_i^i \times r_{i,ci}^i) \quad (2.12)$$

Starting from the initial conditions

$$\omega_0^0 = 0, \alpha_0^0 = 0, a_0^0 = 0, a_{c0}^0 = 0$$

Some vectors should be formed before the forward and backward recursion and they are as follows:

$$z_{i-1}^{i-1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, r_{0,1}^1 = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}, r_{1,2}^2 = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix}, r_{1,c1}^1 = \begin{bmatrix} -l_{c1} \\ 0 \\ 0 \end{bmatrix}, r_{2,c2}^2 = \begin{bmatrix} -l_{c2} \\ 0 \\ 0 \end{bmatrix}$$

• Forward Recursion: Link 1

$$\omega_1^1 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix}, \alpha_1^1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1 \end{bmatrix}, g_1 = g \cdot \begin{bmatrix} -\sin q_1 \\ -\cos q_1 \\ 0 \end{bmatrix}$$

$$a_1^1 = \begin{bmatrix} -l_1 \cdot \dot{q}_1^2 \\ l_1 \cdot \ddot{q}_1 \\ 0 \end{bmatrix}, a_{c1}^1 = \begin{bmatrix} l_{c1} \cdot \dot{q}_1^2 - l_1 \cdot \dot{q}_1^2 \\ l_1 \cdot \ddot{q}_1 - l_{c1} \cdot \ddot{q}_1 \\ 0 \end{bmatrix}$$

• Forward Recursion: Link 2

$$\omega_2^2 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 + \dot{q}_2 \end{bmatrix}, \alpha_2^2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1 + \ddot{q}_2 \end{bmatrix}, g_2 = g \cdot \begin{bmatrix} -\sin(q_1 + q_2) \\ -\cos(q_1 + q_2) \\ 0 \end{bmatrix}$$

$$a_2^2 = \begin{bmatrix} l_1 \cdot \sin q_2 \cdot \ddot{q}_1 - l_2 \cdot (\dot{q}_1 + \dot{q}_2)^2 - l_1 \cdot \cos q_2 \cdot \dot{q}_1^2 \\ l_1 \cdot \sin q_2 \cdot \dot{q}_1^2 + l_2 \cdot (\ddot{q}_1 + \ddot{q}_2) + l_1 \cdot \cos q_2 \cdot \ddot{q}_1 \\ 0 \end{bmatrix},$$

$$a_{c2}^2 = \begin{bmatrix} (l_{c2} - l_2) \cdot (\dot{q}_1 + \dot{q}_2)^2 + l_1 \cdot \sin q_2 \cdot \ddot{q}_1 - l_1 \cdot \cos q_2 \cdot \dot{q}_1^2 \\ (l_2 - l_{c2}) \cdot (\ddot{q}_1 + \ddot{q}_2) + l_1 \cdot \cos q_2 \cdot \ddot{q}_1 + l_1 \cdot \sin q_2 \cdot \dot{q}_1^2 \\ 0 \end{bmatrix}$$

(ii) Backward Force Recursion

$$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + m_i g_0^i + m_i a_{ci}^i \quad (2.13)$$

$$\mu_i^i = -f_i^i \times (r_{i-1,i}^i + r_{i,ci}^i) + R_{i+1}^i \mu_{i+1}^{i+1} + R_{i+1}^i f_{i+1}^{i+1} \times r_{i,ci}^i + I_i^i \alpha_i^i + \omega_i^i \times (I_i^i \omega_i^i) \quad (2.14)$$

$$\tau_i = (R_i^{i-1} \mu_i^i)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + F_{vi} \dot{q}_i + F_{Ci} \text{sgn}(\dot{q}_i) \quad (2.15)$$

• Backward Recursion: Link 2

$$f_2^2 + m_2 g_2 = m_2 a_{c2}^2 \quad (2.16)$$

$$\mu_2^2 = -f_2^2 \times (r_{1,2}^2 + r_{2,c2}^2) + I_2^2 \alpha_2^2 + \omega_2^2 \times (I_2^2 \omega_2^2) \quad (2.17)$$

$$\tau_2 = (R_2^1 \mu_2^2)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + F_{v2} \dot{q}_2 + F_{C2} \text{sgn}(\dot{q}_2) \quad (2.18)$$

• Backward Recursion: Link 1

$$f_1^1 + m_1 g_1 - (R_2^1)^T f_2^2 = m_1 a_{c1}^1 \quad (2.19)$$

In the world inertial coordinate frame, the Euler equation for the rotational motion is

$$f_1^1 \times (r_{0,1}^1 + r_{1,c1}^1) + \mu_1^1 - f_2^2 \times r_{1,c1}^1 - \mu_2^2 = \frac{d}{dt} (I_1 \omega_1 + I_{m1} z_0 \cdot 18 \cdot \dot{q}_2 + I_p z_p \cdot 5.23 \cdot \dot{q}_2) \quad (2.20)$$

By simplifying it and expressing each item in the body attached frame 1, we have

$$\begin{aligned} \mu_1^1 = & -f_1^1 \times (r_{0,1}^1 + r_{1,c1}^1) + R_2^1 \cdot \mu_2^2 + (R_2^1 \cdot f_2^2) \times r_{1,c1}^1 + I_1^1 \alpha_1^1 + \omega_1^1 \times (I_1^1 \omega_1^1) \\ & + 18 I_{m1} \ddot{q}_2 (R_1^0)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + 5.23 I_p \left[\ddot{q}_2 (R_1^0)^T z_p + \dot{q}_2 \omega_1^1 \times ((R_1^0)^T z_p) \right] \end{aligned} \quad (2.21)$$

$$\tau_1 = (R_1^0 \mu_1^1)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + F_{v1} \dot{q}_1 + F_{C1} \text{sgn}(\dot{q}_1) \quad (2.22)$$

Here, I_{m1} and I_p are scalar values representing the axial inertia of the rotor of motor 1, and the axial inertia of the intermediate stage pulley of the fourth joint, respectively, and z_p is the unit vector of the pulley rotating axis.

After some complex symbolic calculations, we arrive at the following equations to describe the dynamic model of the simplified planar two-link manipulator:

$$\begin{aligned}
& [m_1 l_{c1}^2 + I_1 + I_2 + 798.0625 I_{m1} + 798.0625 I_{m2} + m_p l_p^2 + I_p + m_2 l_1^2 + m_2 l_{c2}^2 \\
& + 2m_2 l_1 l_{c2} \cos(q_2)] \cdot \ddot{q}_1 + [m_2 l_1 l_{c2} \cos(q_2) + m_2 l_{c2}^2 + I_2 + 5.23 I_p] \cdot \ddot{q}_2 \\
& - 2m_2 l_1 l_{c2} \sin(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 - m_2 l_1 l_{c2} \sin(q_2) \cdot \dot{q}_2^2 + F_{c1} \cdot \text{sgn}(\dot{q}_1) + F_{v1} \cdot \dot{q}_1 \\
& - m_1 g l_{c1} \cdot \sin(q_1) - m_p g l_p \cdot \sin(q_1) - m_2 g l_1 \cdot \sin(q_1) - m_2 g l_{c2} \cdot \sin(q_1 + q_2) = \tau_1
\end{aligned} \tag{2.23}$$

$$\begin{aligned}
& [m_2 l_1 l_{c2} \cos(q_2) + m_2 l_{c2}^2 + I_2 + 5.23 I_p] \cdot \ddot{q}_1 + [m_2 l_{c2}^2 + I_2 + 324 I_{m3} + 27.3529 I_p] \cdot \ddot{q}_2 \\
& - m_2 l_1 l_{c2} \sin(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 + F_{c2} \cdot \text{sgn}(\dot{q}_2) + F_{v2} \cdot \dot{q}_2 - m_2 g l_{c2} \cdot \sin(q_1 + q_2) = \tau_2
\end{aligned} \tag{2.24}$$

where each variable has the following physical meaning (for $i = 1, 2$).

m_i - mass of link i

I_i - moment of inertia of link i with respect to the center of gravity

I_{mi} - moment of inertia of the rotor of each joint

m_p - mass of the intermediate transmission pulley for joint 2 mounted in the elbow

I_p - moment of inertia of intermediate transmission pulley for joint 2 mounted in the elbow, which is the one about its rotating axis

l_p - distance between the joint 1 axis and the rotating axis of pulley in the elbow

l_{ci} - distance from the center of gravity of link i to the initial body frame i

l_i - length of link i

F_{ci} - Coulomb friction coefficient of joint i

F_{vi} - viscous friction coefficient of joint i

The linear regressor representation is given in equation (2.25).

$$\begin{bmatrix} \ddot{q}_1 & 2\cos(q_2) \cdot \dot{q}_1 + \cos(q_2) \cdot \ddot{q}_2 - 2\sin(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 - \sin(q_2) \cdot \dot{q}_2^2 & \ddot{q}_2 & \text{sgn}(\dot{q}_1) & \dot{q}_1 & -\sin(q_1) & -\sin(q_1 + q_2) & 0 & 0 & 0 \\ 0 & \ddot{q}_1 \cdot \cos(q_2) - \sin(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 & \ddot{q}_1 & 0 & 0 & 0 & -\sin(q_1 + q_2) & \ddot{q}_2 & \text{sgn}(\dot{q}_2) & \dot{q}_2 \end{bmatrix} \times \begin{bmatrix} m_1 l_{c1}^2 + I_1 + I_2 + 798.0625 I_{m1} + 798.0625 I_{m2} + m_p l_p^2 + I_p + m_2 l_1^2 + m_2 l_{c2}^2 \\ m_2 l_{c2}^2 + I_2 + 5.23 I_p \\ F_{c1} \\ F_{v1} \\ m_1 g l_{c1} + m_p g l_p + m_2 g l_1 \\ m_2 g l_{c2} \\ m_2 l_{c2}^2 + I_2 + 324 I_{m3} + 27.3529 I_p \\ F_{c2} \\ F_{v2} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad \dots \quad (2.25)$$

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} \ddot{q}_1 & 2\cos(q_2) \cdot \dot{q}_1 + \cos(q_2) \cdot \ddot{q}_2 - 2\sin(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 - \sin(q_2) \cdot \dot{q}_2^2 & \ddot{q}_2 & \text{sgn}(\dot{q}_1) & \dot{q}_1 & -\sin(q_1) & -\sin(q_1 + q_2) & 0 & 0 & 0 \\ 0 & \ddot{q}_1 \cdot \cos(q_2) - \sin(q_2) \cdot \dot{q}_1 \cdot \dot{q}_2 & \ddot{q}_1 & 0 & 0 & 0 & -\sin(q_1 + q_2) & \ddot{q}_2 & \text{sgn}(\dot{q}_2) & \dot{q}_2 \end{bmatrix} \quad (2.26)$$

$$\Pi = \begin{bmatrix} m_1 l_{c1}^2 + I_1 + I_2 + 798.0625 I_{m1} + 798.0625 I_{m2} + m_p l_p^2 + I_p + m_2 l_1^2 + m_2 l_{c2}^2 \\ m_2 l_{c2}^2 + I_2 + 5.23 I_p \\ F_{c1} \\ F_{v1} \\ m_1 g l_{c1} + m_p g l_p + m_2 g l_1 \\ m_2 g l_{c2} \\ m_2 l_{c2}^2 + I_2 + 324 I_{m3} + 27.3529 I_p \\ F_{c2} \\ F_{v2} \end{bmatrix} \quad (2.27)$$

After rearrangement of the elements, the joint space dynamic model can be represented in a compact matrix form as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v \dot{q} + F_c \text{sgn}(\dot{q}) + G(q) = \tau - J^T(q)F_e \quad (2.28)$$

where

q - joint position vector, in this project, it is a 2×1 vector

\dot{q} - joint velocity vector, in this project, it is a 2×1 vector

\ddot{q} - joint acceleration vector, in this project, it is a 2×1 vector

$M(q)$ - inertia matrix, in this project, it is a 2×2 square matrix

$C(q, \dot{q})$ - Coriolis and Centrifugal matrix, in this project, it is a 2×2 square matrix

F_v - viscous friction matrix, in this project, it is a 2×2 square matrix

F_c - Coulomb friction matrix, in this project, it is a 2×2 diagonal square matrix

$\text{sgn}(\dot{q})$ - 2×1 vector whose components are given by the sign functions of the single

joint velocity

$G(q)$ - gravity vector, in this project, it is a 2×1 vector

τ - joint actuator torque vector, in this project, it is a 2×1 vector

$J(q)$ - Jacobian of the manipulator, in this project, it is a 2×2 vector

F_e - external force acting in the end-effector, in this project, it is a 2×1 vector

The joint space dynamic model has the following properties.

Property 1 $M(q)$ is positive definite and norm bounded in the whole workspace

$$M(q) = M^T(q) > 0, \sigma_m(q) \leq |M(q)| \leq \sigma_M(q)$$

where $\sigma_m(q)$ and $\sigma_M(q)$ are scalar functions of the joint position

Property 2 $\dot{q}^T \cdot [\dot{M}(q) - 2C(q, \dot{q})] \cdot \dot{q} = 0$

Property 3 Linearity in the dynamic parameters

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v\dot{q} + F_c \operatorname{sgn}(\dot{q}) + G(q) = Y(q, \dot{q}, \ddot{q})\Pi \quad (2.29)$$

where $Y(q, \dot{q}, \ddot{q})$ is the linear regressor, which as seen from equation (2.26), is a function of the joint states and the geometrical parameters of the manipulator. Π as defined in equation (2.27) is a function of the unknown dynamic parameters of the manipulator, individually or in combination. This property is the basis of adaptive motion control of robot manipulators and dynamic parameter identification which are detailed in the following sections.

In fact joint space dynamics is the basis for Cartesian dynamics. Eventually all the control algorithms are realized by sending joint torques to actuators. Cartesian space dynamics is important for studying the interaction control algorithms. For this reason, the Cartesian space dynamics is discussed in section 2.3.2.

2.3.2 Cartesian Space Dynamic Model

Suppose that the world coordinate frame is defined as in Figure 2.2. The forward kinematics may be used to establish the relationship between the Cartesian space position of the end-effector and the joint position. The end effector coordinates $X \in R^2$ may be represented in terms of the joint position vector $q \in R^2$ as

$$X = f(q) \quad (2.30)$$

where f is a function determined by the DH parameters of the manipulator. It is reasonable to assume that the joint position vector $q \in R^2$ does not pass through any singularity point when the trajectory of the manipulator is properly planned.

The differential kinematics, which describes the relationship between the end effector velocity and the joint velocity is given by

$$\dot{X} = J(q)\dot{q} \quad (2.31)$$

where $J(q)$ is the manipulator Jacobian. The relation between the Cartesian space acceleration and the joint space acceleration is obtained by differentiating equation (2.31):

$$\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (2.32)$$

From equation (2.29), we have

$$\ddot{q} = -M^{-1}(q)[C(q, \dot{q})\dot{q} + F_v\dot{q} + F_c \operatorname{sgn}(\dot{q}) + G(q)] + M^{-1}(q)\tau - M^{-1}(q)J^T(q)F_e \quad (2.33)$$

Substituting equation (2.33) into equation (2.32), we get

$$\begin{aligned} \ddot{X} = & -J(q)M^{-1}(q)[C(q, \dot{q})\dot{q} + F_v\dot{q} + F_c \operatorname{sgn}(\dot{q}) + G(q)] \\ & + J(q)M^{-1}(q)\tau - J(q)M^{-1}(q)J^T(q)F_e + \dot{J}(q)\dot{q} \end{aligned} \quad (2.34)$$

From (2.31), we have

$$\dot{q} = J^{-1}(q)\dot{X} \quad (2.35)$$

Substitute equation (2.35) into equation (2.34):

$$\begin{aligned} \ddot{X} = & -J(q)M^{-1}(q)[C(q, \dot{q})J^{-1}(q)\dot{X} + F_vJ^{-1}(q)\dot{X} + F_c \operatorname{sgn}(\dot{q}) + G(q)] \\ & + J(q)M^{-1}(q)\tau - J(q)M^{-1}(q)J^T(q)F_e + \dot{J}(q)J^{-1}(q)\dot{X} \end{aligned} \quad (2.36)$$

Multiple both sides of equation (2.36) by $J^{-T}(q)M(q)J^{-1}(q)$ and simplify, to get

$$\begin{aligned} J^{-T}(q)M(q)J^{-1}(q)\ddot{X} + J^{-T}(q)[C(q, \dot{q})J^{-1}(q) + F_vJ^{-1}(q) - M(q)J^{-1}(q)\dot{J}(q)J^{-1}(q)]\dot{X} \\ + J^{-T}(q)F_c \operatorname{sgn}(\dot{q}) + J^{-T}(q)G(q) = J^{-T}(q)\tau - F_e \end{aligned} \quad \dots \quad (2.37)$$

Define

$$J^{-T}(q)M(q)J^{-1}(q) = M_c(X) \quad (2.38)$$

$$J^{-T}(q)[C(q, \dot{q})J^{-1}(q) - M(q)J^{-1}(q)\dot{J}(q)J^{-1}(q)] = C_c(X, \dot{X}) \quad (2.39)$$

$$J^{-T}(q)F_vJ^{-1}(q) = F_{vc}(X) \quad (2.40)$$

$$J^{-T}(q)F_c = F_{cc}(X) \quad (2.41)$$

$$J^{-T}(q)G(q) = G_c(X) \quad (2.42)$$

$$J^{-T}(q)\tau = F_\tau(X) \quad (2.43)$$

Then (2.37) may be rewritten as

$$M_c(X)\ddot{X} + C_c(X, \dot{X})\dot{X} + F_{vc}(X)\dot{X} + F_{cc}(X) + G_c(X) = F_\tau(X) - F_e \quad (2.44)$$

Equation (2.44) is called the Cartesian space dynamics of a manipulator. It has the following property.

Property 1 $M_c(X)$ is positive definite and norm bounded in the whole workspace

$$M_c(X) = M_c^T(X) > 0, \quad \sigma_{mC}(X) \leq |M_c(X)| \leq \sigma_{MC}(X) \quad (2.45)$$

where $\sigma_{mC}(X)$ and $\sigma_{MC}(X)$ are scalar functions of the joint position

2.3.3 Problem with Dynamic Model based on CAD Models

For comparison, the dynamic model is derived based on the dynamic parameters from the CAD models provided by the manufacturer. Since the inertia parameters for each link is given, the symbolic representation for each matrix in equation (2.29) may be easily derived using the Robotics Toolbox developed by Corke [107] for the original four-DOF manipulator. However, since the WAM system is simplified into a two DOFs system, the dynamic model for the new two-DOF manipulator instead of the original four-DOF one should be obtained. This is done in a straightforward manner by substituting $q_1 = q_3 = 0$ and $\dot{q}_1 = \dot{q}_3 = 0$ in the symbolic representations calculated by the Robotics Toolbox. The corresponding Cartesian space dynamic model may be derived in the same way as in section 2.3.2. However, the dynamic parameters provided by the manufacturer do not include the friction coefficients. This can result in serious dynamic modeling errors. Derivations based on CAD models using Robotics Toolbox ignore these items. Errors are expected in the inverse dynamics-based output torques and the actual joint torques under the same joint trajectories and same control algorithms.

Inverse dynamics-based trajectory tracking control algorithms using the CAD model-based dynamic parameters are implemented in the simplified WAM. This is a joint space control algorithm; so no singularity handling technique is used.

WAM does not have absolute joint encoders. Its joint positions are calculated with reference to some known fixed position in the world frame. Fortunately, WAM provides this position, which is the “home position.” WAM should be brought to its home position

before each experiment. However, this home position is the limit position of the two joints. A proper trajectory planning technique has to be applied to move WAM to some desired point far from these limit positions.

The initial values of $q = [q_1 \ q_2]^T$ when WAM is in the home position are $q = [-2.0051 \ 3.1428]^T$. Let us move WAM to some desired position $q = [0 \ \pi/2]^T$ after 4 seconds. At $t = 4\text{s}$, both joints will be still ($\dot{q} = [\dot{q}_1 \ \dot{q}_2]^T = [0 \ 0]^T$). After that, both joints are commanded to have some periodic motion such as a sinusoid or a combination of sinusoids. To have a smooth transition between these two phases, it is required to have consistent acceleration at $t = 4\text{s}$. Thus, for phase one, we have four known conditions for each joint:

$$q(t=0) = [-2.0051 \ 3.1428]^T, \quad \dot{q}(t=0) = [0 \ 0]^T$$

$$q(t=4) = [0 \ \pi/2]^T, \quad \dot{q}(t=4) = [0 \ 0]^T$$

Unique third order polynomials may be determined from these four known conditions. After that, sinusoidal waves are found for both joints. The amplitudes of both waves are selected to be 0.5. The trajectory commands for both joints are given in equation (2.46) and (2.47), respectively. The reference position, velocity and acceleration trajectories are given in Figure 2.19 – Figure 2.21, respectively.

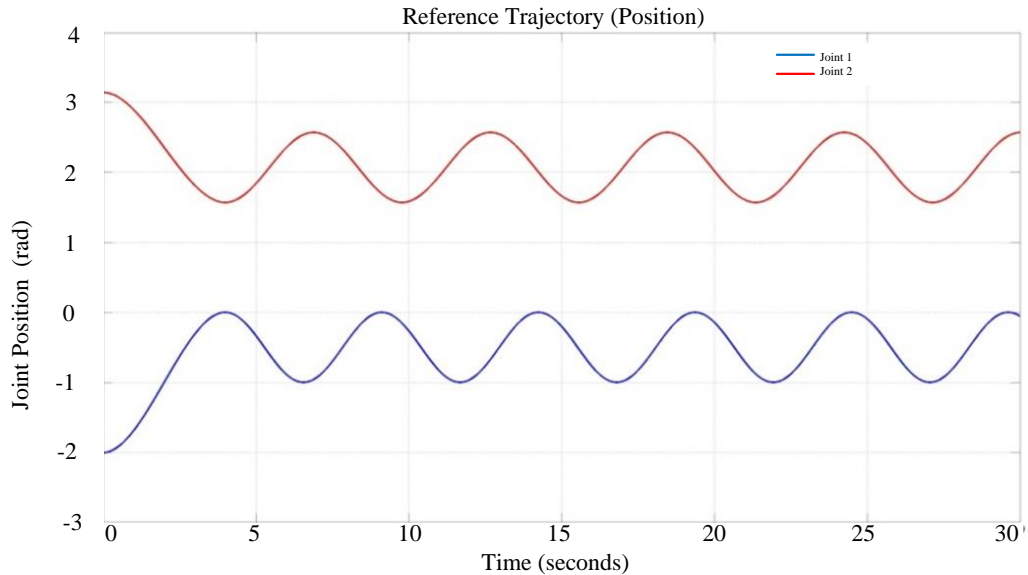


Figure 2.19: Joint space position reference trajectories

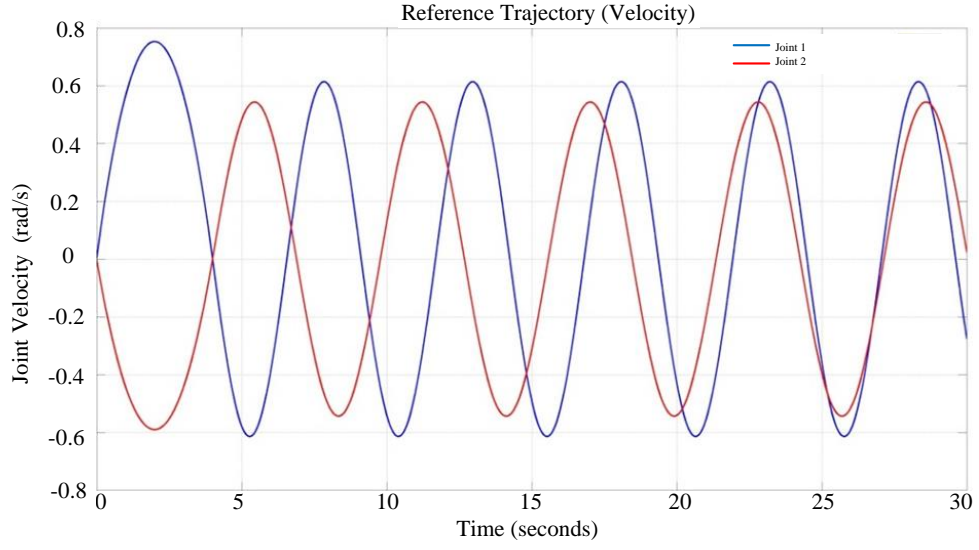


Figure 2.20: Joint space velocity reference trajectories

$$q_1(t) = \begin{cases} -0.0627t^3 + 0.3760t^2 - 2.0051 & 0 \leq t \leq 4 \\ 0.5 \sin(1.2270t + 1.5734) - 0.5019 & t \geq 4 \end{cases} \quad (2.46)$$

$$q_2(t) = \begin{cases} 0.0491t^3 - 0.2948t^2 + 3.1428 & 0 \leq t \leq 4 \\ -0.5 \sin(1.0852t + 1.5678) + 2.0684 & t \geq 4 \end{cases} \quad (2.47)$$

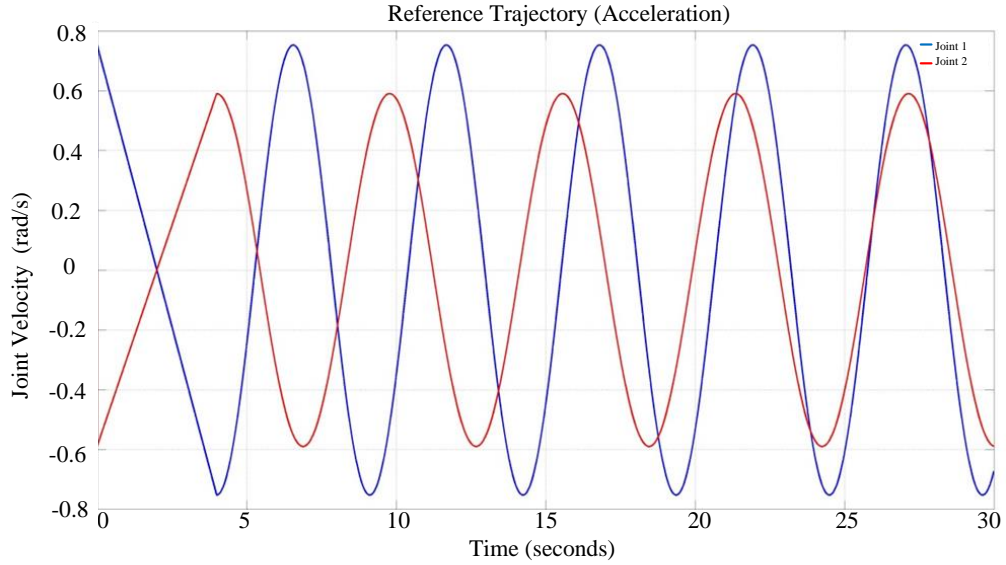


Figure 2.21: Joint space acceleration reference trajectories

The inverse dynamics-based control algorithm is used to carry out the reference trajectories task, as

$$\tau = \overline{M}(q)y + \overline{C}(q, \dot{q})\dot{q} + \overline{G}(q) \quad (2.48)$$

$$y = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \quad (2.49)$$

where $\overline{M}(q)$, $\overline{C}(q)$ and $\overline{G}(q)$ are the corresponding matrices based on the CAD model after simplification, q_d is the reference trajectory, and K_d and K_p are proper gains to guarantee the convergence of the tracking error.

The corresponding block diagram is designed to implement this algorithm. In the WAM system, the starting time of any experiment should be the time when the “Active” button on the safety panel is pressed. However, once the control algorithms are downloaded into the target PC, the time clock starts to record the time information regardless of whether WAM is in the “Active” mode. Only the running time after the algorithm is downloaded and executed in the target is available. For the planned trajectories described in equation (2.46) and equation (2.47), the starting time should be the time point when the “Active” button in the safety panel is pressed. Thus, the time at which the “Active” button in the safety panel is pressed is critical here and should be snapped by proper algorithms.

We have developed a method to detect the time point at which the “Active panel” is pressed. The value at this time point should be subtracted from the total target PC running time. The purpose of this is to let the trajectory generation algorithm start only when the “Active” is triggered. Otherwise, the trajectory generation algorithm will start from the time the application is downloaded, and will give incorrect reference trajectories to WAM.

The time point at which the “Active” button is pressed may be snapped by the block diagram shown in Figure 2.22. The global block representation of the algorithm implementation is given in Figure 2.24. The internal block with the details of this algorithm is given in Figure 2.23.

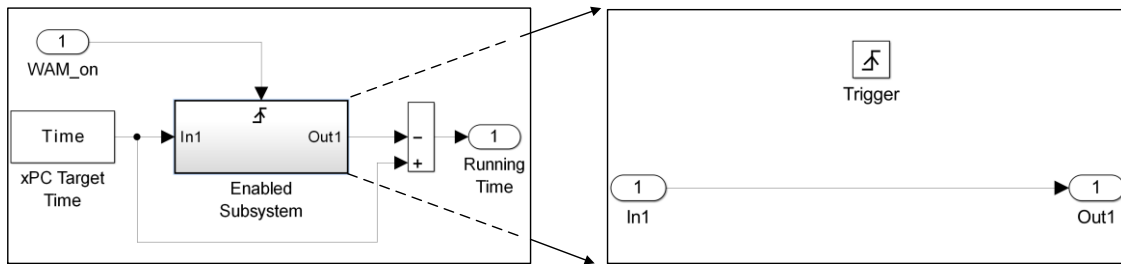


Figure 2.22: WAM starting time snapping

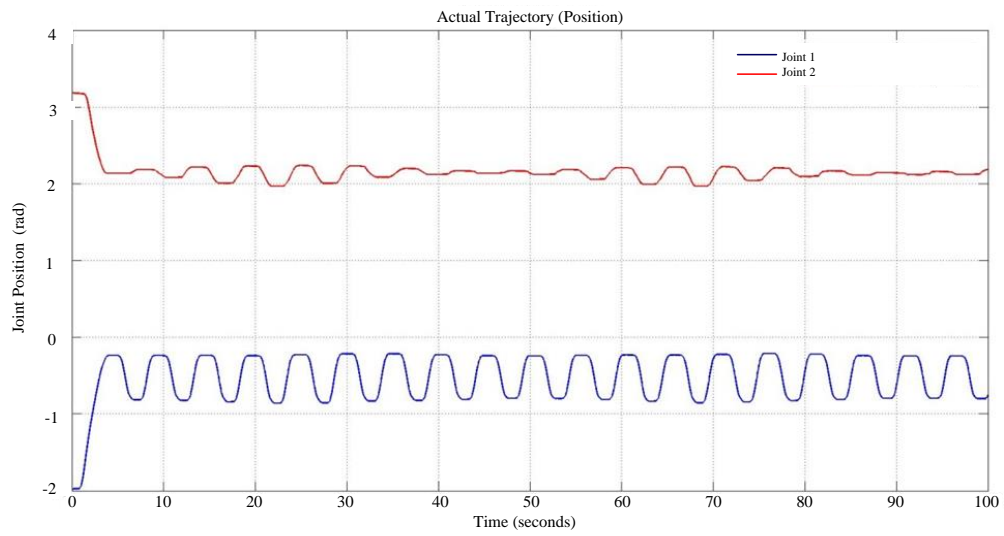


Figure 2.25: Actual joint space position trajectories

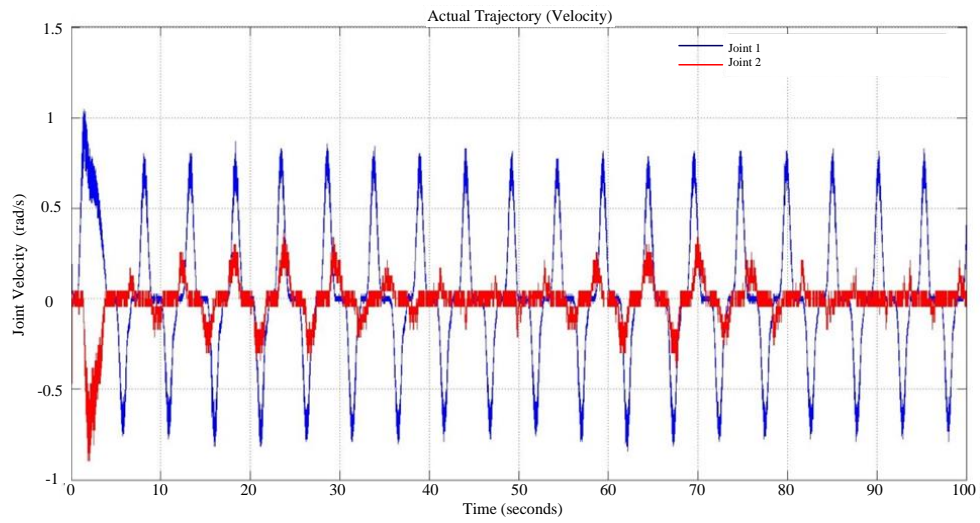


Figure 2.26: Actual joint space velocity trajectories

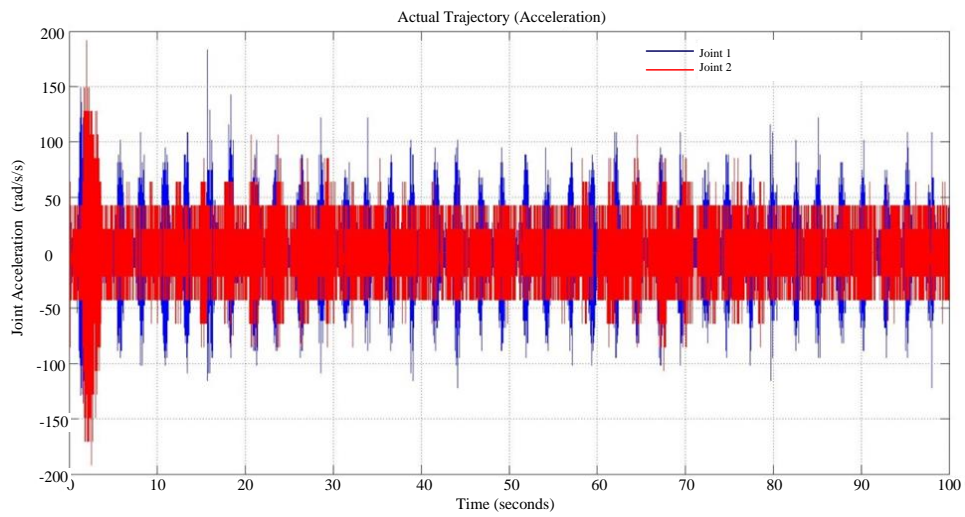


Figure 2.27: Actual joint space acceleration trajectories

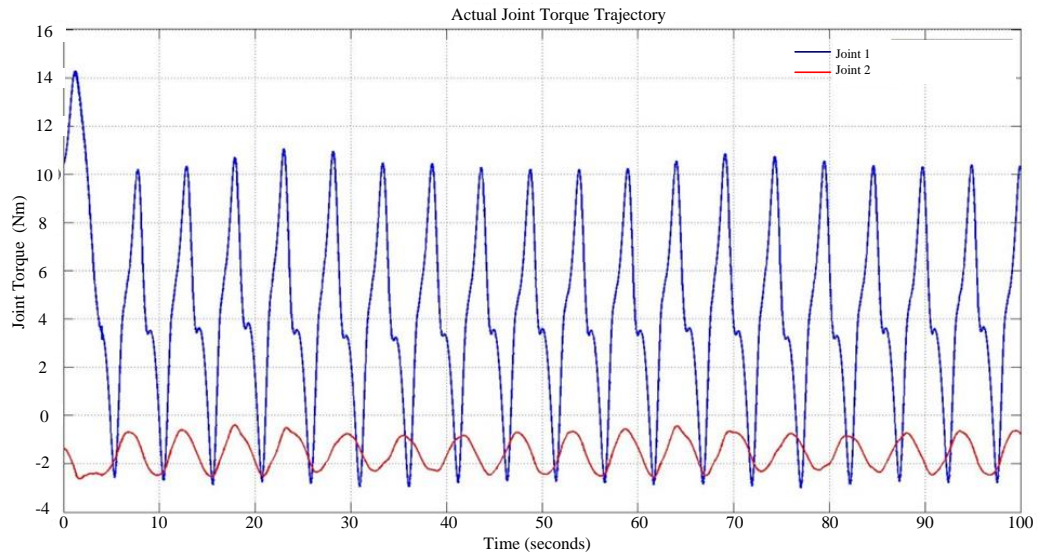


Figure 2.28: Joint torque trajectories

It is observed from Figure 2.26 and Figure 2.27 that the velocity and acceleration trajectories are very noisy. This is unacceptable in control algorithm implementations if the velocity and acceleration information is directly used in the controllers. The possible cause for the noisy velocity and acceleration trajectories is the noise in the original actual joint position measurements. Direct differentiation of the joint position with respect to time will further increase the amplitude of noise in the velocity and acceleration. Hence, robust velocity and acceleration reconstruction algorithms should be applied instead of direct differentiation. This is detailed in the following subsections.

From Figure 2.25 and Figure 2.28, we can see that even though the applied torque to joint 2 seems to be periodic, the joint 2 motion is not periodic. We speculate that this is due to the static friction in this joint. The same phenomenon is observed in joint 1. We can conclude that joint friction which is not taken into consideration in the manufacturer provided algorithms should be considered in an accurate dynamic model.

Summarizing, the dynamic model derived based on the manufacturer provided dynamic parameters is not accurate. We noticed large trajectory tracking errors when using this nominal model in the inverse dynamics-based trajectory tracking control algorithms. This justifies the need for dynamic parameter identification of WAM by experiments, as detailed next.

2.4 Dynamic Model Identification

Dynamic model identification concerns experimentally determining an accurate dynamic model of a dynamic system such as a robot. A model of this type may be used, in particular, for model-based control algorithms. How to determine the dynamic parameters using experimental measurements is discussed in this section.

There are two types of approaches of dynamic model identification. The first one is done off line, after obtaining the experimental data. Specifically, after the experiments, the recorded inputs and outputs of the manipulator are analyzed to determine the model. The second type of dynamic identification is done on line. Here, the identification is carried out as the data is acquired during the experiment.

Two challenges exist in dynamic model identification, which have been addressed by researchers in the past decades. The first one is the persistent excitation of the reference trajectory that is tracked, by using proper controllers. A reference trajectory must excite all necessary dynamics of the manipulator. In the design of a reference trajectory, the workspace of the manipulator should be considered as well. The most common choice for the reference trajectory is a combination of harmonic signals of different magnitudes and appropriate frequencies.

The second challenge in dynamic model identification is the calculation of velocity and acceleration using joint position. Velocity and acceleration are required during on-line or off-line data analysis. As we noticed in section 2.3, there is significant noise in the velocity and acceleration information when determined by direct differentiation. This problem can be reduced by using a low-pass filter to remove the noise before calculating the velocity or acceleration. However, this cannot guarantee the completeness of the velocity (or acceleration) information. In this thesis we use robust exact differentiators to reconstruct velocity and acceleration.

2.4.1 Reference Trajectory Selection

In order to improve the convergence rate and the immunity to noise of the identified dynamic parameters, reference trajectories have to be selected properly. The reference trajectories have to have the persistent excitation property in order to correctly identify the dynamic parameters. Also, the reference trajectories have to be selected such that they will not pass the joint limits of the manipulator. For the current robot, the maximum

allowable joint velocity of each joint is 2rad/s. All these factors have to be taken into consideration in the selection of reference trajectories.

Let Y be the regression matrix of measurements, as described in equation (2.50), where n is the number of measurements:

$$Y = [Y_1 \ Y_2 \ \dots \ Y_n]^T \quad (2.50)$$

Due to the high sampling frequency during experiments, two adjacent regression matrices might become almost identical. This will cause a singularity problem in the subsequent data analysis. In order to avoid this problem, in the data analysis process, decimation of five will be applied to extract the measurements. This is equivalent to having a sampling period of 0.01s (100Hz). The experiment is planned to last for 30 seconds after the initial transitional phase. Hence, we will have 3000 measurements, which are adequate for dynamic parameter identification.

Even though the dynamic parameter identification is based on the actual joint motions during the identification experiments, the actual joint motions will be close to the reference ones after the transitional phase, when proper trajectory tracking control algorithms are used. The sensitivity of the LSE results with respect to noisy measurements may be characterized by the condition number of the observation matrix Y . This is a nonlinear optimization problem and many software packages are available to solve it.

We use the criterion introduced in [108], which is the condition number of the observation matrix, as given in equation (2.51). This criterion seeks to maximize the immunity of the identification results to measurement noise.

$$C = \text{cond}(Y) \quad (2.51)$$

As governed by hardware structural limitations, the ranges of motion of the two joints are given by

$$-2.0 \text{ rad} \leq q_1 \leq 2.0 \text{ rad}, -0.9 \text{ rad} \leq q_2 \leq 3.1 \text{ rad} \quad (2.52)$$

In view of the challenges in (2.52), the reference trajectories are selected as

$$q_{di}(t) = \sum_{j=1}^3 a_{ij} \sin(j \cdot \omega_i t + \varphi_{ij}) + b_i \quad (2.53)$$

$$\dot{q}_{di}(t) = \sum_{j=1}^3 a_{ij} \cdot j \cdot \omega_i \cos(j \cdot \omega_i t + \varphi_{ij}) \quad (2.54)$$

$$\ddot{q}_{di}(t) = -\sum_{j=1}^3 a_{ij} \cdot j^2 \cdot \omega_i^2 \cdot \sin(j \cdot \omega_i t + \varphi_{ij}) \quad (2.55)$$

with $i=1,2$, the biases b_i for the reference trajectories of joints 1 and 2 are 0 and 1.1, respectively. The choice of these biases is based on the range of motion of each joint. In order to obtain an accurate dynamic model, the range of motion of each joint should be covered as completely as possible. Then we have 14 parameters at our disposal in order to minimize the condition number of the decimated observation matrix. The constraints for this nonlinear optimization problem are the velocity limits in each joint, which may be described by

$$\sqrt{a_{i1}^2 + 4a_{i2}^2 + 9a_{i3}^2} \cdot \omega_i \leq 1.95 \quad (2.56)$$

Here, for safety reasons, we limit the maximum velocity of each joint to 1.95, even though the allowable value for it is 2 as mentioned before. Commercial software packages such as Matlab are used to determine these values.

The determined reference trajectories are shown in Figure 2.29 and Figure 2.30. It is easy to observe that the reference trajectories are within these ranges and will not harm the equipment.

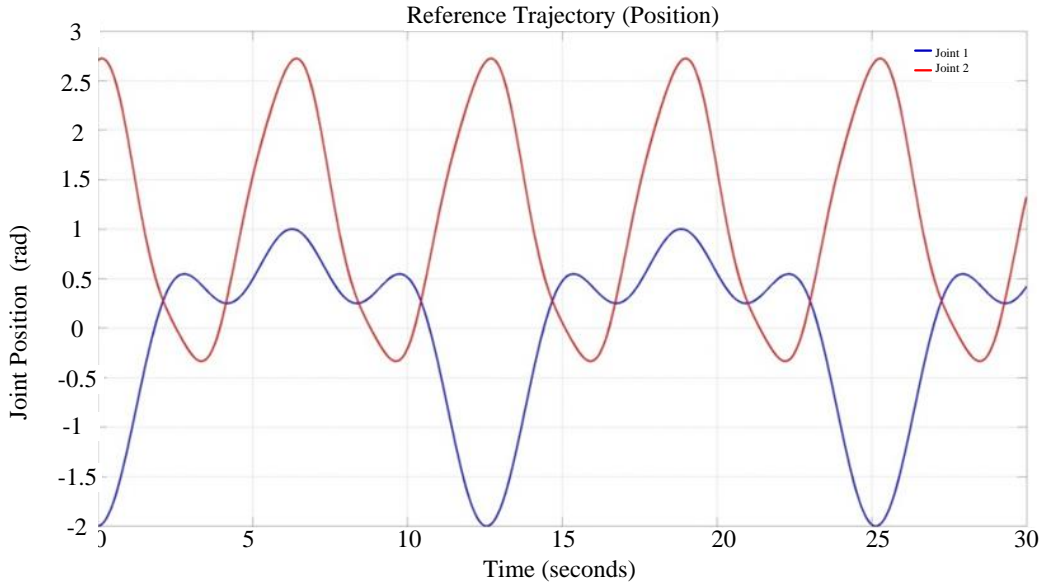


Figure 2.29: Position reference trajectories for each joint

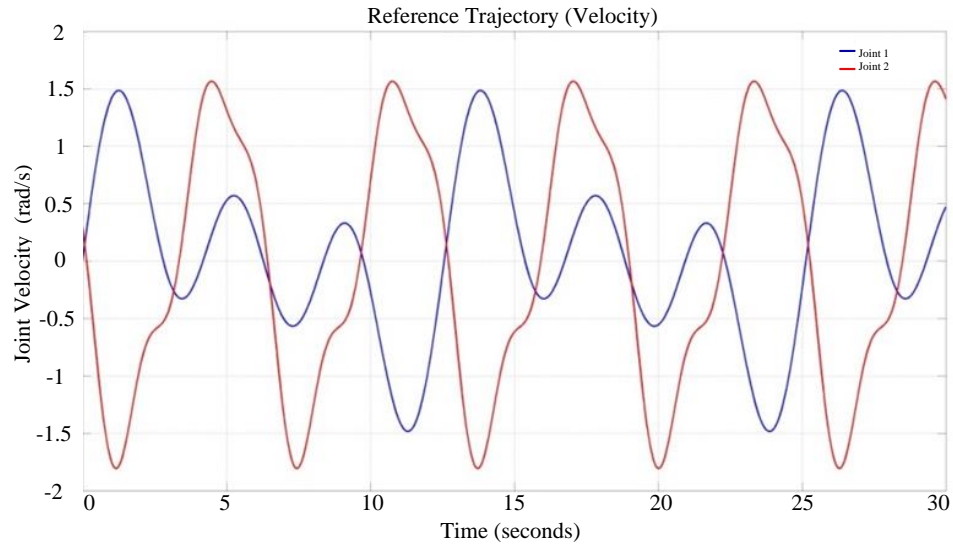


Figure 2.30: Velocity reference trajectories for each joint

2.4.2 Velocity and Acceleration Reconstruction

As can be observed from equation (2.26), the elements of the observation matrix are functions of joint positions, velocities and accelerations. However, there is only one encoder in each joint to measure the joint position, while the joint velocity and acceleration have to be calculated or estimated. Measurement noise is unavoidable in the joint position measurements. Obtaining the joint velocity and acceleration by direct differentiating the position information will lead to noisy results. In the assumed model for friction, this will further result in poor accuracy in dynamic model identification.

Some researchers have used periodic reference trajectories so that the response is also periodic, at steady state. The assumed white noise may be characterized by analyzing the discrepancy of the same data point in different sampling periods. The detected noise will be removed from the measured joint position information. The calculation of velocity and acceleration using noise-free joint position data is straightforward. It can be done in the frequency domain just using algebraic operations.

However, the velocity information calculated in this way is not exact since noise cannot be removed completely. Also, filtering is necessary in this conventional differentiator which makes the calculated velocity and acceleration not exact even when the measurement noise is absent. Model-based filters, such as Kalman filter may be applied to have an estimation of the velocity and acceleration under noise. However, an accurate dynamic model is required in the implementation of the filtering process.

Robust finite time convergence differentiators are applied in the present work to estimate the velocity and acceleration. It is based on the sliding mode theory, and can reconstruct the velocity and acceleration without needing a filter. The estimation is exact in the absence of measurement noise. In the presence of measurement noise, the estimation accuracy may be controlled within some acceptable range, making this robust differentiator ideal for many applications.

The advantage of this type of differentiator is that it is finite-time convergent, and robust to external disturbance. Additionally, it does not require a dynamic model of the plant, but can still provide accurate estimation results for the higher order derivatives of the estimated signal.

The joint position signal $q(t)$ is a function defined on $[0, \infty)$ consisting of a bounded Lebesgue-measurable noise without known features. The joint position signal without noise is denoted by $q_0(t)$, which is unknown. When the Lipschitz constant of the joint position signal is known as L , the conventional sliding mode-based robust differentiator given by the following equations may be applied to estimate $\dot{q}_0(t)$ and $\ddot{q}_0(t)$:

$$\dot{z}_0 = -\lambda_2 L^{1/3} |z_0 - q|^{2/3} \text{sign}(z_0 - q) + z_1 \quad (2.57)$$

$$\dot{z}_1 = -\lambda_1 L^{1/2} |z_1 - v_0|^{1/2} \text{sign}(z_1 - v_0) + z_2 \quad (2.58)$$

$$\dot{z}_2 = -\lambda_0 \cdot L \cdot \text{sign}(z_2 - v_1) \quad (2.59)$$

where z_0 , z_1 and z_2 are states of the robust estimator; v_0 , v_1 are intermediate variables introduced in order to implement this differentiator; and λ_0 , λ_1 and λ_2 are proper positive constant observer gains to guarantee the finite-time convergence of the proposed observer. The finite time convergence of the proposed differentiator is guaranteed by selecting $\lambda_0 = 1.1$, $\lambda_1 = 1.5$ and $\lambda_2 = 3$. This fact has been proved by using the concept of homogeneity. In the absence of measurement noise, after finite-time transient process, we have

$$z_0 = q_0 \quad (2.60)$$

$$z_1 = v_0 = \dot{q}_0 \quad (2.61)$$

$$z_2 = v_1 = \ddot{q}_0 \quad (2.62)$$

Let the measurement noise be a Lebesgue-measurable function with magnitude bounded by ε . The differentiator errors are bounded by

$$|z_1 - \dot{q}_0| = O(\varepsilon^{2/3}) \quad (2.63)$$

$$|z_2 - \ddot{q}_0| = O(\varepsilon^{1/3}) \quad (2.64)$$

However, the robust differentiator given by equation (2.57) – equation (2.59) is based on the assumption that the Lipschitz constant of the joint acceleration signal is known. In practice, due to uncertainty and measurement noise, this constant is not easy to determine. The estimation results become very poor when this parameter is incorrectly chosen.

The adaptive second order robust differentiator is used here to estimate the velocity and acceleration signals $\dot{q}_0(t)$, $\ddot{q}_0(t)$. It has been applied in [49] for dynamic model identification of a two-axis manipulator, and it has been proved to be very effective compared to the conventional Euler differentiator and low-pass filter. However, only offline dynamic parameter identification was carried out there. Here we have a different platform and it is desirable to use this differentiator in the present work to develop an online dynamic model identification algorithm. The adaptive robust differentiator is given by:

$$\dot{z}_0 = -\hat{\lambda}_0 |z_0 - q|^{2/3} \text{sign}(z_0 - q) + z_1 - K_0(z_0 - q) \quad (2.65)$$

$$\dot{z}_1 = -\hat{\lambda}_1 |z_1 - v_0|^{1/2} \text{sign}(z_1 - v_0) - z_2 - K_1(z_1 - v_0) \quad (2.66)$$

$$\dot{z}_2 = -\hat{\lambda}_2 \text{sign}(z_2 - v_1) \quad (2.67)$$

where z_0 , z_1 and z_2 are states of the robust estimator; v_0 , v_1 are intermediate variables introduced in order to implement this differentiator; $\hat{\lambda}_0$, $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are observer gains to be adapted in order to deal with the uncertainty of the Lipschitz constant of the joint acceleration signal; and K_0 and K_1 are two matrices used to ensure the convergence of the observer gains to the actual ones. They are designed to be diagonal, and each element of these two matrices is a positive scalar. The adaption laws for $\hat{\lambda}_0$, $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are given by:

$$\begin{bmatrix} \dot{\hat{\lambda}}_1 \\ \dot{\hat{\lambda}}_2 \end{bmatrix} = -\Gamma_1^{-1} \begin{bmatrix} -|z_1 - v_0|^{1/2} \text{sign}(z_1 - v_0) \\ -\int \text{sign}(z_1 - v_0) d\tau \end{bmatrix} (z_1 - v_0) \quad (2.68)$$

$$\dot{\hat{\lambda}}_0 = (z_0 - q) |z_0 - q|^{2/3} \quad (2.69)$$

This adaptive robust differentiator guarantees the convergence of the velocity and acceleration reconstruction errors to zero. Reconstruction of the velocity and acceleration information based on this differentiator is used in the subsequent dynamic model identification tasks.

The robot manipulator is commanded to move in free space to track joint space trajectories. Any simple reference trajectory tracking algorithm could be used to track the joint space trajectory. The widely used inverse dynamics algorithm is given by:

$$\tau = \bar{M}(q)y + \bar{C}(q, \dot{q})\dot{q} + F_v\dot{q} + F_c \operatorname{sgn}(\dot{q}) + \bar{G}(q) \quad (2.70)$$

$$y = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q) \quad (2.71)$$

where K_P and K_D are PD gains. Both of them are diagonal matrices, for simplicity.

Since the actual mass and inertia properties of the simplified two-link manipulator are unknown, a very rough estimation of $\bar{M}(q)$, $\bar{C}(q, \dot{q})$, F_v , F_c and $\bar{G}(q)$ will be used in the control algorithm given by (2.70) and (2.71). The values used are from the CAD model discussed in section 2.3.3 except for F_v and F_c . In order to improve the tracking performance, extensive experiments have been carried out. The trajectory tracking accuracies are compared using the control law (2.70) with different F_v and F_c . The values of F_v and F_c used here are the ones that give the best tracking performance. Even though the actual tracking performance in system identification experiments is not important, this dedicated choice of F_v and F_c is used here. The actual joint position and actuator output, instead of the reference trajectories, are used for post-experiment analysis. The actual velocity information used in the control law may be obtained by the robust differentiator developed in this section.

The actual trajectories are shown in Figure 2.31. The corresponding position values and the joint torque are saved into a file in the target PC. After the experiments, these files are transmitted back to the host PC for analysis.

From Figure 2.31, it is observed that by including the friction component in the model based controller, the joint responses have become periodic, even though there are some “flattened” joint positions, which imply that there is static friction in each joint. However, at least the response now is periodic.

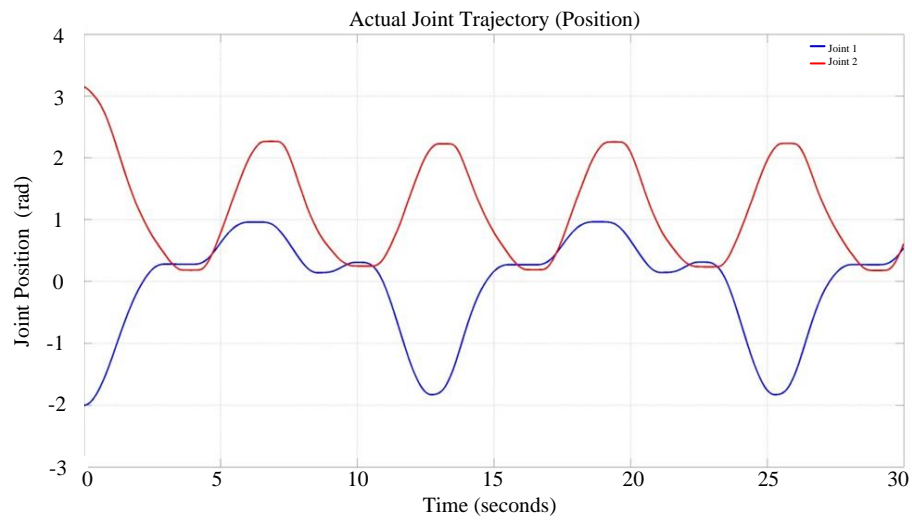


Figure 2.31: Actual joint position trajectories

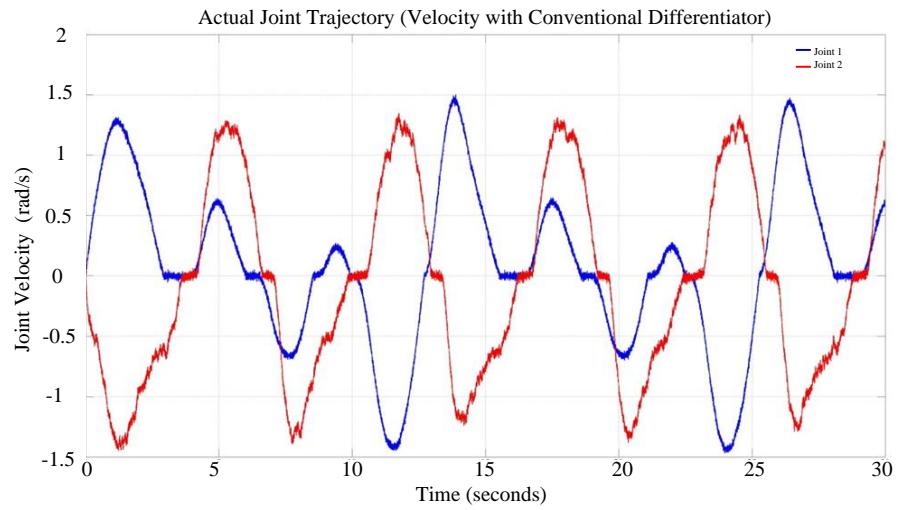


Figure 2.32: Actual joint velocity trajectories based on conventional differentiator

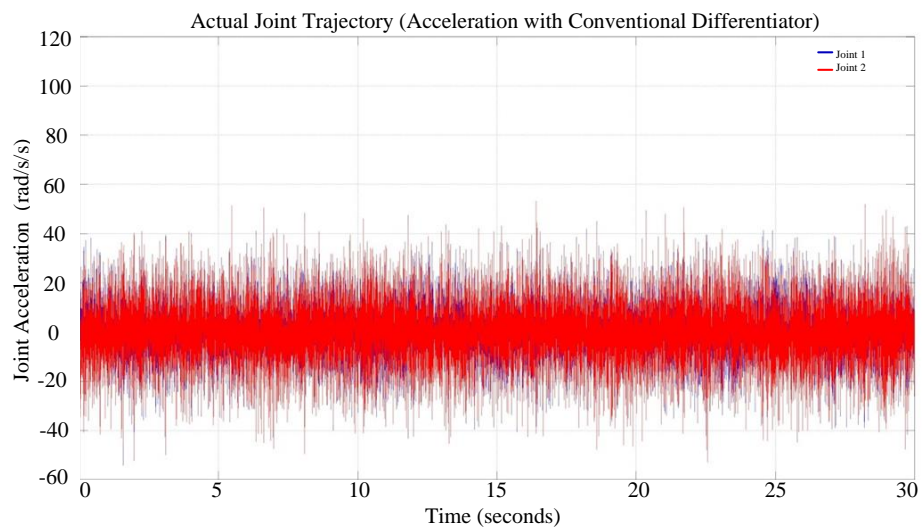


Figure 2.33: Actual joint acceleration trajectories based on conventional differentiator

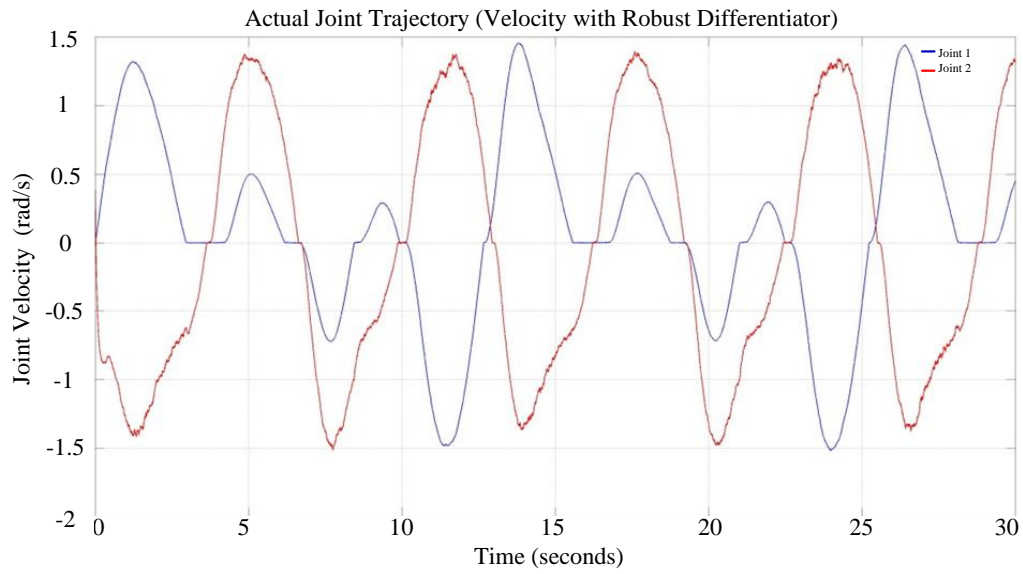


Figure 2.34: Actual joint velocity trajectories based on robust differentiator

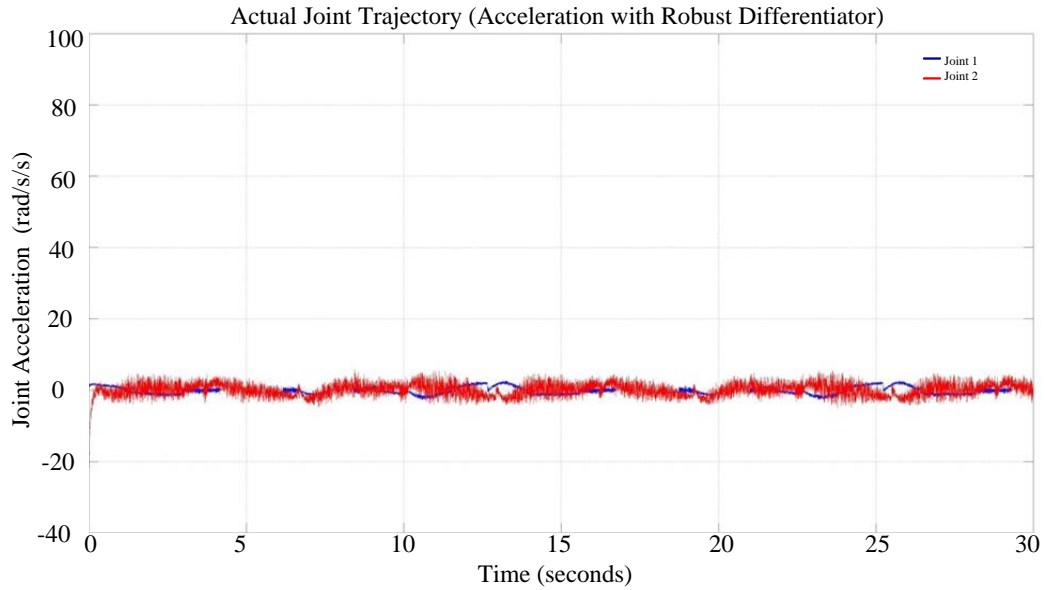


Figure 2.35: Actual joint acceleration trajectories based on robust differentiator

What is more, as observed from Figure 2.32, the velocities obtained by the conventional Euler differentiator have some noise. This is more obvious in Figure 2.33, which are the trajectories of acceleration. On the other hand, Figure 2.34 and Figure 2.35 show that the velocity and acceleration trajectories obtained by the robust differentiator have more desirable features with regard to the noise level. This further justifies the selection of the robust differentiator instead of the conventional Euler differentiator. The reconstructed velocity and acceleration are used in the offline and online of dynamic identification schemes.

2.4.3 Offline Identification of Dynamic Model

Offline identification of dynamic model is used to identify the model parameters (or functions of some of the parameters) after obtaining the necessary data using experiments. Specifically, the input joint actuator torques and the output joint positions of the experimental system are recorded during trajectory tracking experiments.

Least squares estimation is applied to the measured data, to estimate the unknown dynamic parameters, as

$$\Pi = (\bar{Y}^T \bar{Y})^{-1} \bar{Y}^T \bar{\tau} \quad (2.72)$$

where $\bar{Y} = [Y_1 \ Y_2 \ \dots \ Y_n]^T$ and $\bar{\tau} = [\tau_1 \ \tau_2 \ \dots \ \tau_n]^T$ are the vectors formed by the corresponding data points. The friction component in the dynamic model is also identified. Even though this component is small compared to other components, it is better to include it for better accuracy.

The validation process of the identified dynamic parameters is illustrated in Figure 2.36.

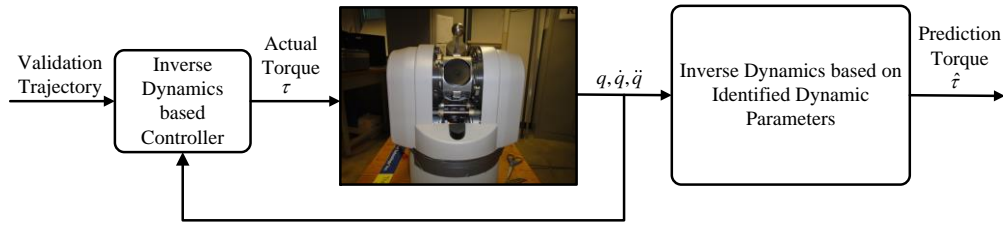


Figure 2.36: Validation scheme of the offline identified dynamic parameter

For validation of the identified dynamic parameters, new reference trajectories shown in Figure 2.37 are commanded to the trajectory tracking algorithm. The inverse dynamics-based trajectory tracking algorithm is used.

The actual position, velocity and acceleration of the joints are the input to the offline identified dynamic model. The output of the inverse dynamics model is the required joint torques to induce the actual joint motions. These joint torques are compared with the actual joint torques produced by the joint actuator. The smaller the discrepancy between them, the more accurate the identified dynamic model is.

The reference validation trajectory for each joint is given in Figure 2.37. This trajectory corresponds to the desired Cartesian space motion for impedance control which is detailed in chapters 3 and 4. The trajectory tracking performance is given in Figure 2.38. As we can observe, model based controller using the dynamic model

obtained by offline identification significantly improves the trajectory tracking performance. At least the motion of joint 2, seems more regular. This may be due to the inclusion of the friction components in the model based controller.

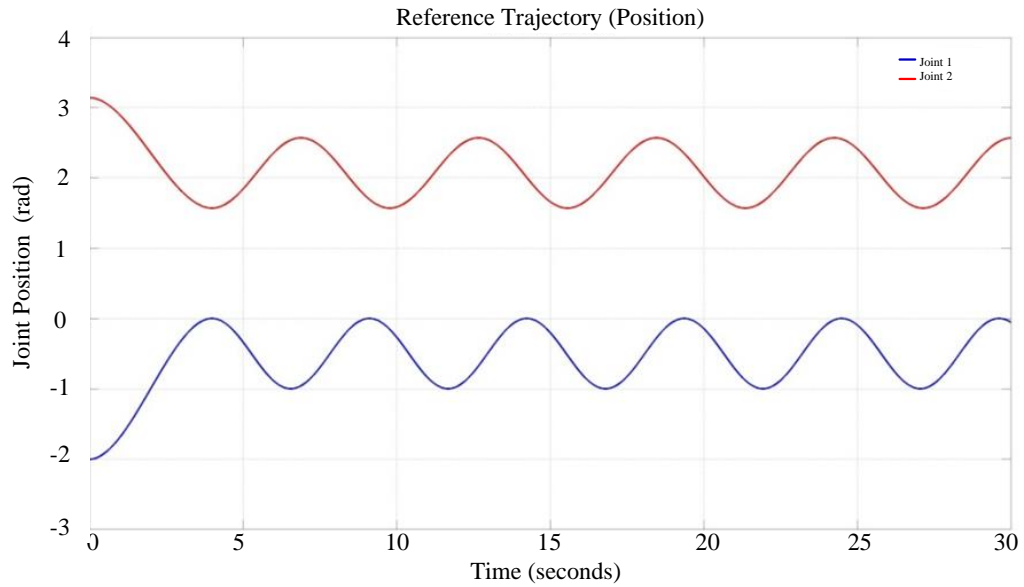


Figure 2.37: Reference trajectories for joint 1 and 2

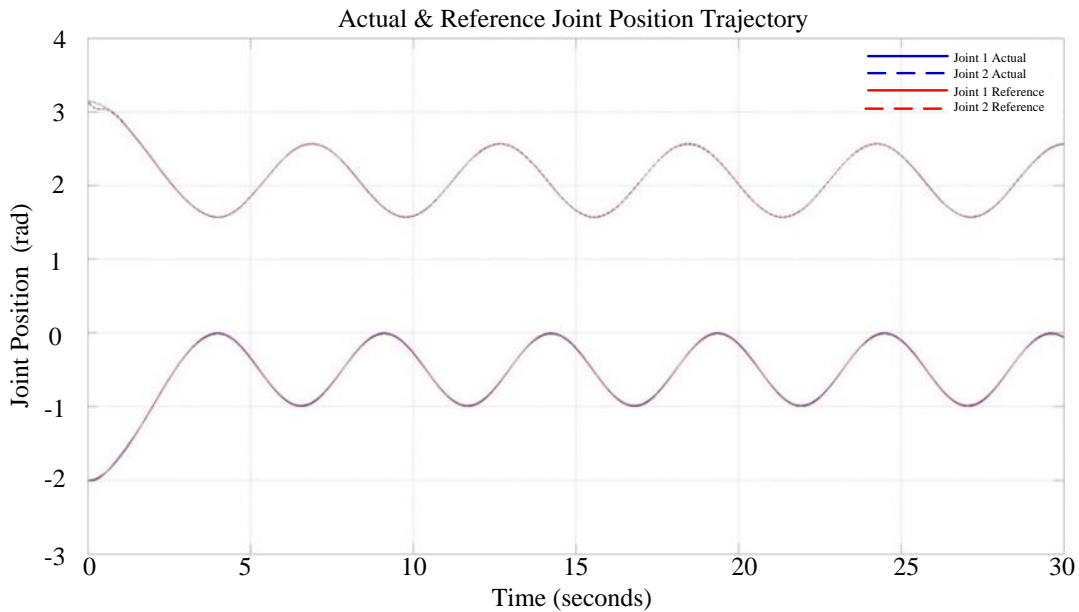


Figure 2.38: Trajectory tracking performance for joints 1 and 2 (offline)

The comparison between the actual joint torque and the one calculated based on the inverse dynamics using offline identified dynamic parameters for each joint are given in Figure 2.39 and Figure 2.41 respectively. A satisfactory match between the predicted torque based on the identified dynamic model and the actual torque delivered by the joint actuators is observed. The difference between the measured and the predicted torques for

each joint is given in Figure 2.40 and Figure 2.42, respectively. It is observed that the torque error for joint 1 is ignorable, while for joint 2 it is noticeable. The possible causes for this are the time-varying dynamic parameters in WAM or the un-modelled dynamics, such as the flexibility of the cable-drive system. The friction coefficient in each joint is a time-varying parameter as has been pointed out by many researchers. A complex friction model can be used, such as one that could characterize the nonlinear dynamics of friction. For simplicity, in the present work, we will use the friction model that we proposed.

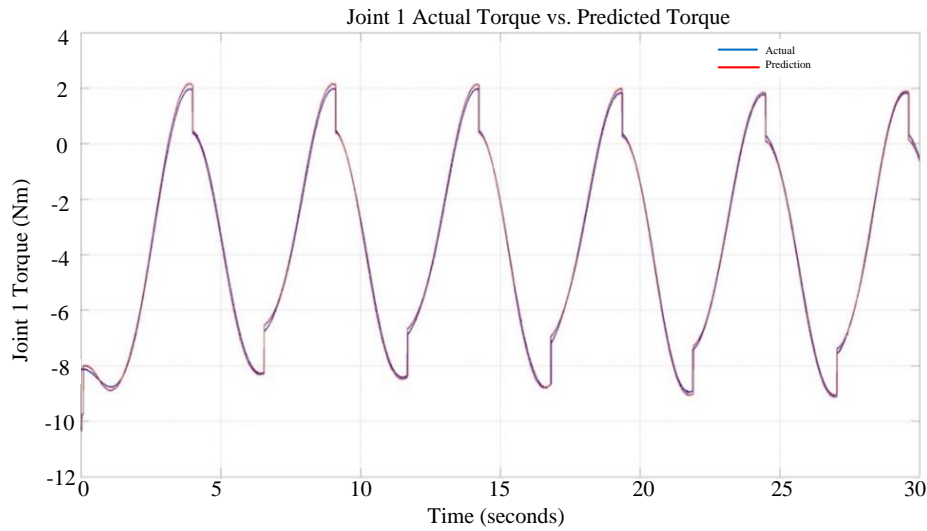


Figure 2.39: Comparison between the predicted torque and the actual torque (offline)

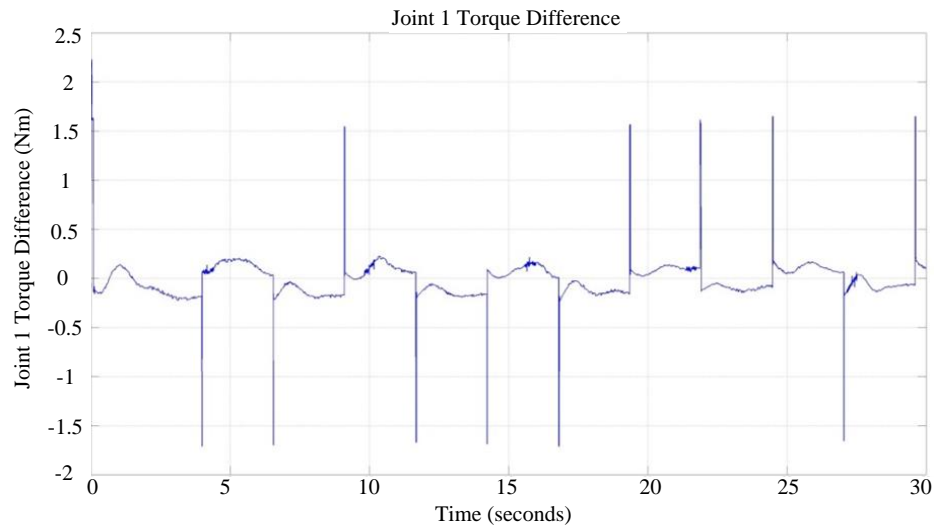


Figure 2.40: Difference between joint 1 predicted torque and actual torque (offline)

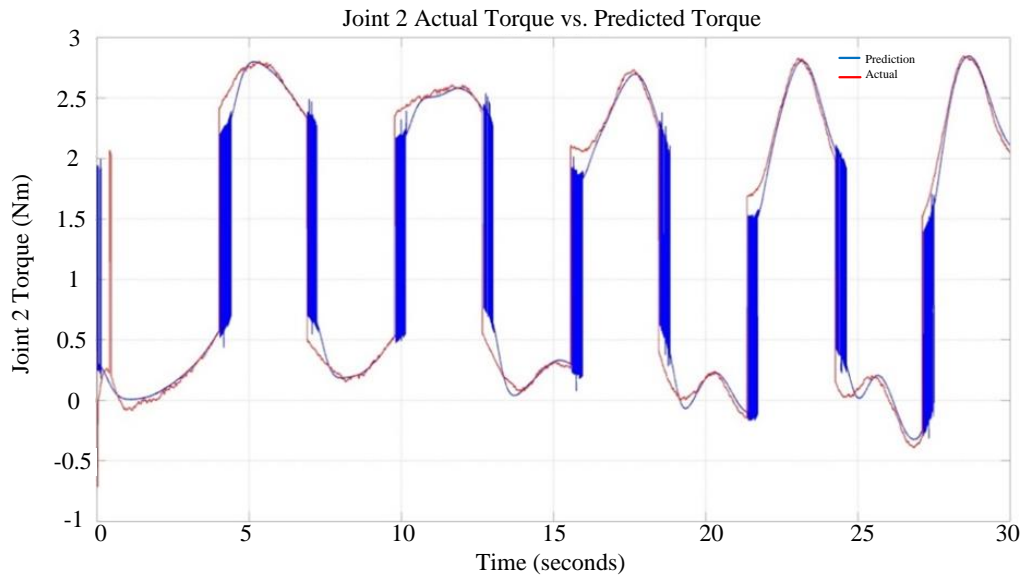


Figure 2.41: Comparison between joint 2 predicted torque and actual torque (offline)

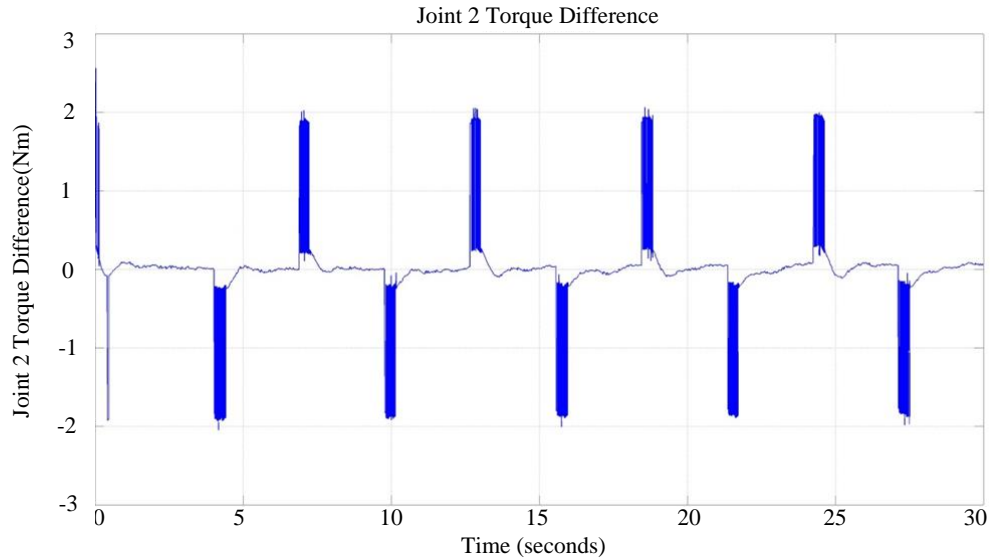


Figure 2.42: Difference between the predicted torque and the actual torque (offline)

In conclusion, the dynamics identification algorithm that is proposed in the foregoing subsections is an offline one. It provides acceptable identified dynamic parameters for model-based control algorithms. However, to deal with time-varying parameters, it is better to have an online dynamic parameter identification algorithm. This problem is discussed in next.

2.4.4 Online Dynamic Model Identification

While offline identification of a dynamic model can provide acceptable results, in some delicate manipulations, a more accurate dynamic model may be required. Friction coefficient in the dynamic model of a manipulator is usually time-varying. The offline dynamic parameter identification algorithm is based on the assumption that the dynamic parameters are identical in different working conditions. This assumption is not very practical. The online dynamic identification algorithm may be used to improve the accuracy of the identified dynamic model. It identifies the dynamic parameters based on the most recent information under the same working conditions.

Since the offline identified parameters are acceptable, the online dynamic parameter identification process is based on these results. We assume that only the friction component is time-varying. The online parameter identification algorithm is used on the friction coefficients. From equation (2.28), after rearrangement, we have

$$F_v \dot{q} + F_c \operatorname{sgn}(\dot{q}) = \tau - [M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)] \quad (2.73)$$

$$\begin{bmatrix} F_{v1} & 0 & F_{c1} & 0 \\ 0 & F_{v2} & 0 & F_{c2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \operatorname{sgn}(\dot{q}_1) \\ \operatorname{sgn}(\dot{q}_2) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (2.74)$$

where $F_v = \begin{bmatrix} F_{v1} & \\ & F_{v2} \end{bmatrix}$, $F_c = \begin{bmatrix} F_{c1} & \\ & F_{c2} \end{bmatrix}$, $\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \tau - [\hat{M}(q)\ddot{q} + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q)]$, $\hat{M}(q)$,

$\hat{C}(q, \dot{q})$ and $\hat{G}(q)$ are calculated using the offline identified dynamic parameters.

The friction parameters to be identified are decoupled, and equation (2.74) may be written as

$$\dot{q}_1 \cdot F_{v1} + \operatorname{sgn}(\dot{q}_1) \cdot F_{c1} = \tau_1 \quad (2.75)$$

$$\dot{q}_2 \cdot F_{v2} + \operatorname{sgn}(\dot{q}_2) \cdot F_{c2} = \tau_2 \quad (2.76)$$

Recursive least squares estimation may be used in online identification of these time-varying dynamic parameters. The first joint is considered as an example. We have

$$\tau_1 = \dot{q}_1 \cdot F_{v1} + \operatorname{sgn}(\dot{q}_1) \cdot F_{c1} = \begin{bmatrix} \dot{q}_1 & \operatorname{sgn}(\dot{q}_1) \end{bmatrix} \begin{bmatrix} F_{v1} \\ F_{c1} \end{bmatrix} \quad (2.77)$$

Let $x = \begin{bmatrix} \dot{q}_1 \\ \text{sgn}(\dot{q}_1) \end{bmatrix}$, $\theta_f = [F_{c1} \ F_{v1}]$, and $y = \tau_1$. Then $y = \theta x$. In the discrete time domain, the recursive least squares estimation for joint 1 is given by

$$\hat{\theta}_f(k+1) = \hat{\theta}_f(k) + K(k+1) [y(k+1) - \hat{\theta}_f(k)x(k+1)] \quad (2.78)$$

$$K(k+1) = \frac{P(k) \cdot x(k+1)}{1 + x^T(k+1)P(k)x(k+1)} \quad (2.79)$$

$$P(k+1) = P(k) - \frac{P(k)x(k+1)x^T(k+1)P(k)}{1 + x^T(k+1)P(k)x(k+1)} \quad (2.80)$$

where k is the time stamp.

The initial value for $\theta_f(0)$ is taken from the offline identified results. The initial value for

$$P \text{ is } P(0) = \begin{bmatrix} 10 & \\ & 10 \end{bmatrix}.$$

The online identified dynamic parameters are used in the model based controller to calculate the next step actuating torque. This process is shown in Figure 2.43. The friction is compensated online.

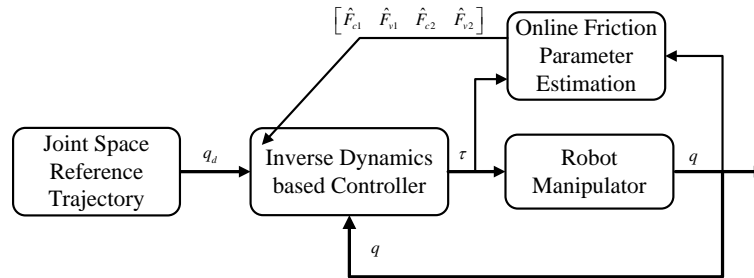


Figure 2.43: Friction compensation based on online parameter estimation

The same optimized reference trajectory as in the offline identification case has been used here for online identification of the time-varying dynamic parameters. An inverse dynamics-based controller algorithm is used for trajectory tracking control. The trajectory tracking performance is shown in Figure 2.44.

The algorithm gives the friction coefficient shown in Figure 2.45 and Figure 2.46. Due to the noise in the velocity information, the estimated parameters have some chattering. However, those friction parameters are within a specific range. The exact same procedure may be carried out on joint 2 to obtain the recursive least squares estimation of the friction parameters of joint 2. The identified results are given in Figure 2.47 and Figure 2.48. Similar conclusions may be made on the estimated parameters.

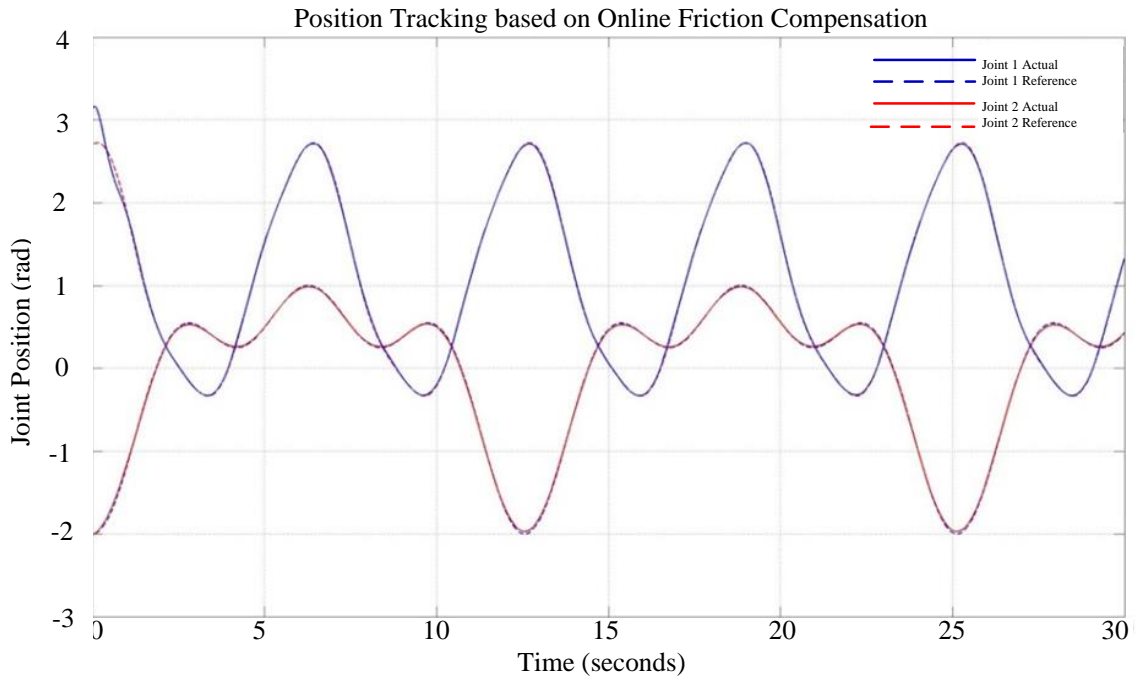


Figure 2.44: Trajectory tracking performance for joint 1 and 2 (online)

The chattering may also be attributed to the noise in the velocity information. This problem can be reduced by using an improved model for friction torque modelling. For example, for Coulomb friction, a modification to the friction model may be done by defining the friction torque when the joint velocity crosses zero. Further complex models for friction such as the Stribeck friction model, may also be applied to reduce this problem.

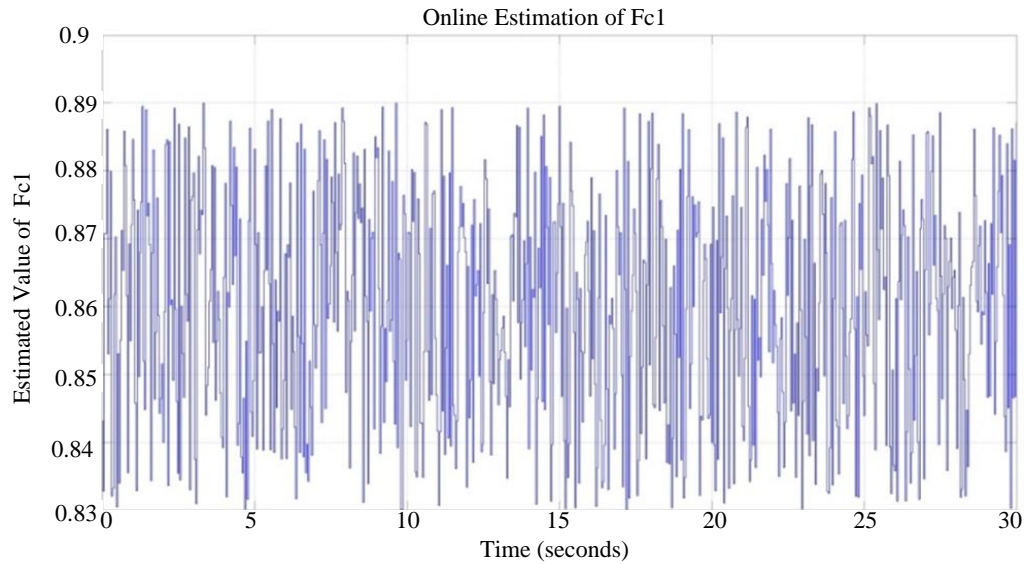


Figure 2.45: Online estimation of F_{c1}

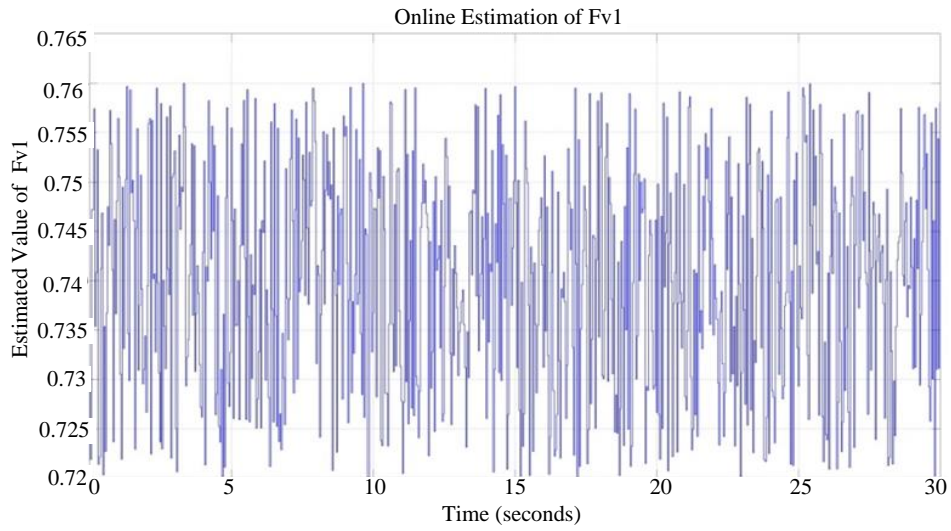


Figure 2.46: Online estimation of F_{v1}

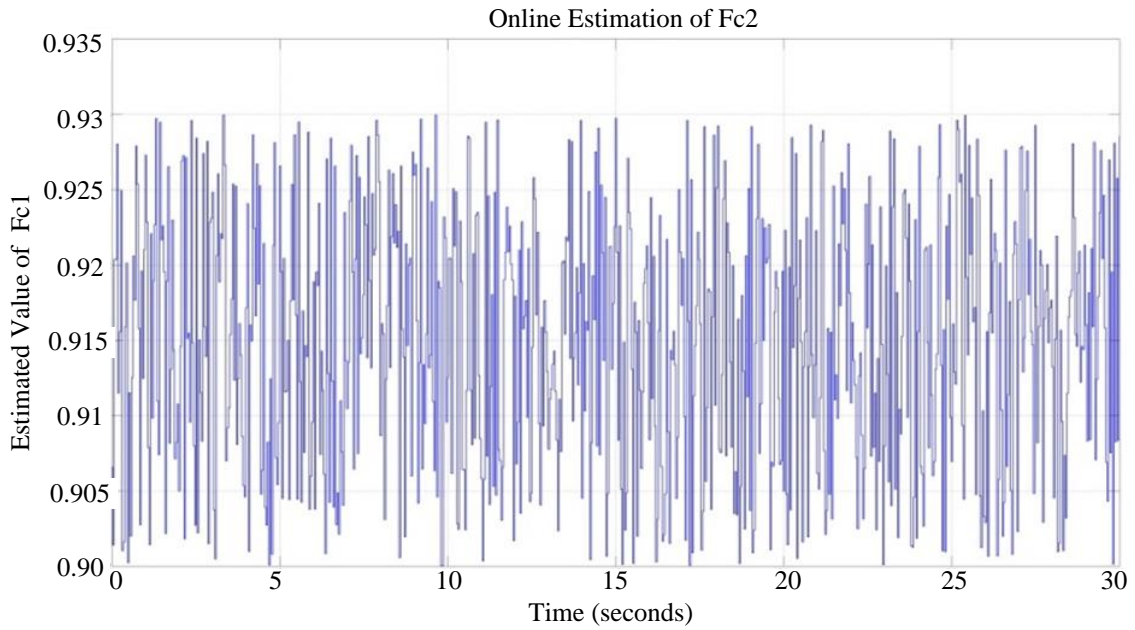


Figure 2.47: Online estimation of F_{c2}

The validation process for the online dynamic parameter identification algorithm is shown in Figure 2.49. For validation of the proposed online identification algorithm, the same validation trajectory as in the offline validation process is selected. Also, the online identified friction parameters of each joint will be used for the next time step control. A validation of the trajectory tracking performance is given in Figure 2.50.

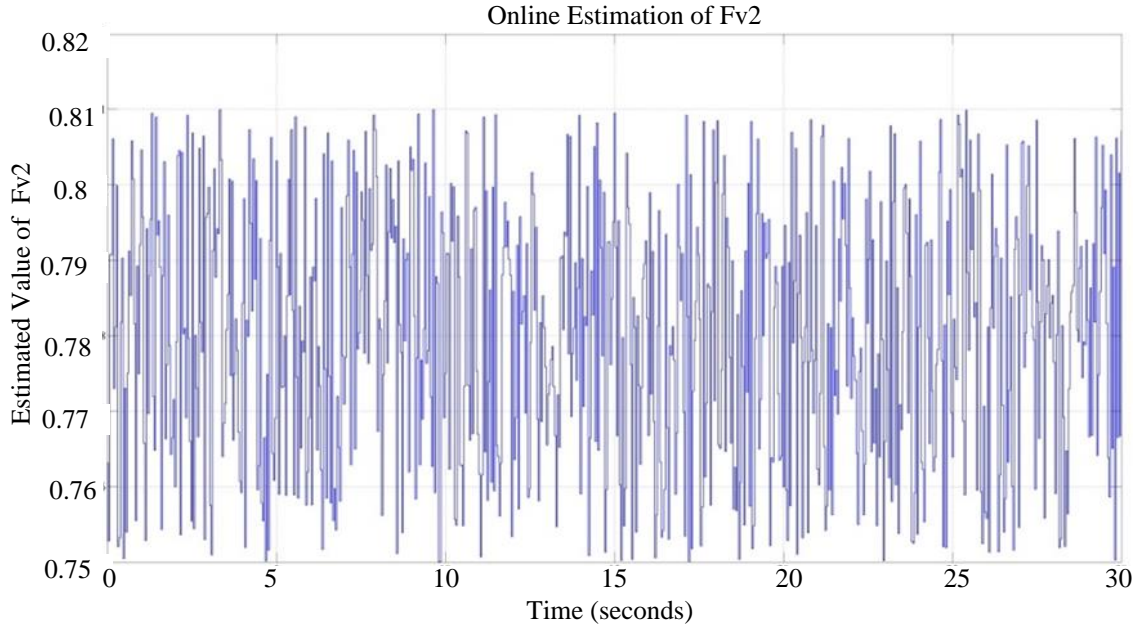


Figure 2.48: Online estimation of F_{v2}

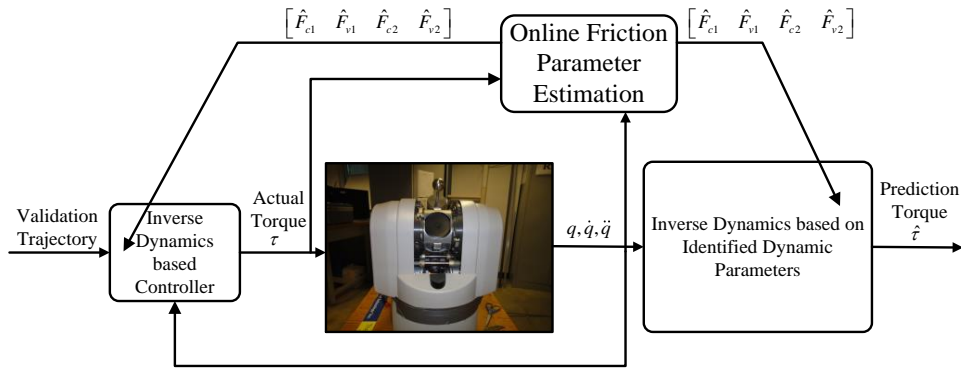


Figure 2.49: Scheme of the online dynamic parameter validation process

A comparison of the actual joint torque and the one calculated based on the inverse dynamics using online identified dynamic parameters is given in Figure 2.51 and Figure 2.53. The torque prediction error in each joint is given by Figure 2.52 and Figure 2.54. We can observe from Figure 2.52 and Figure 2.54 that online estimation-based friction compensation can slightly improve the performance. This may be attributed to the noisy joint velocity measurement. Joint friction modelling in robot dynamics is extremely complex. Obtaining an accurate dynamic model seems difficult or even impossible due to the inherent nonlinear characteristics or friction torque at low joint velocities. A soft computing technique may be applied to model this uncertain part.

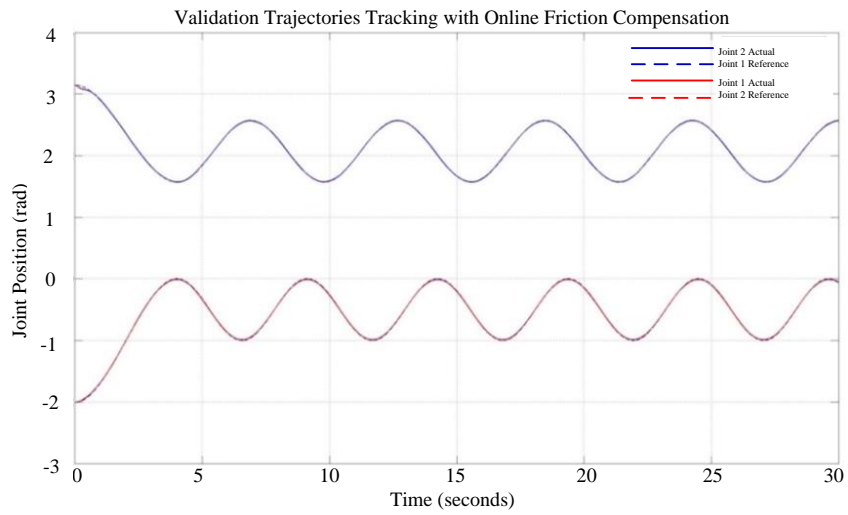


Figure 2.50: Trajectory tracking with online friction compensation

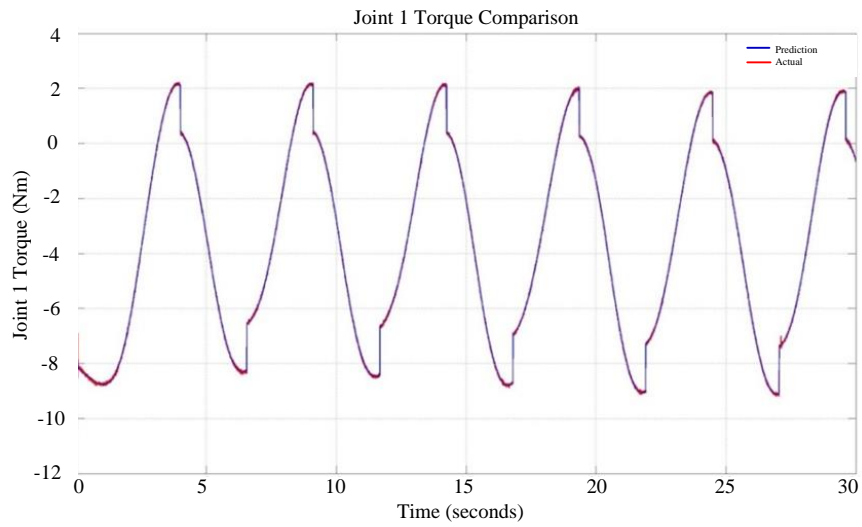


Figure 2.51: Joint 1 actual torque vs. predicted torque with online identification

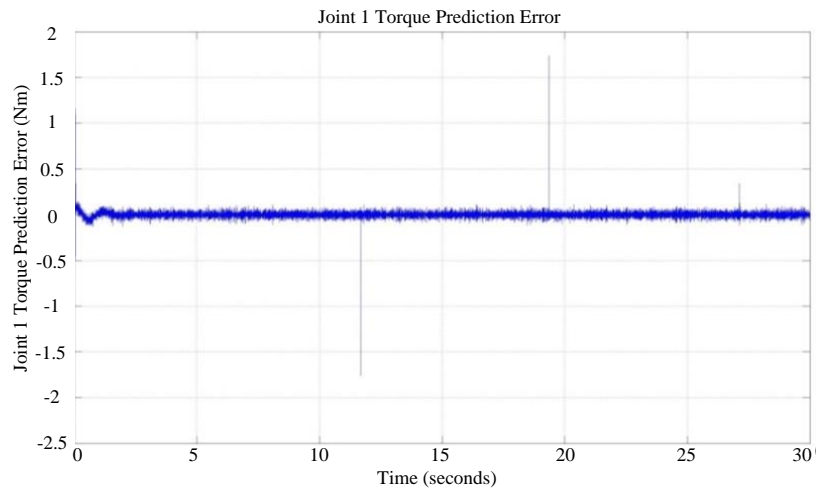


Figure 2.52: Joint 1 torque prediction error with online dynamic parameter identification

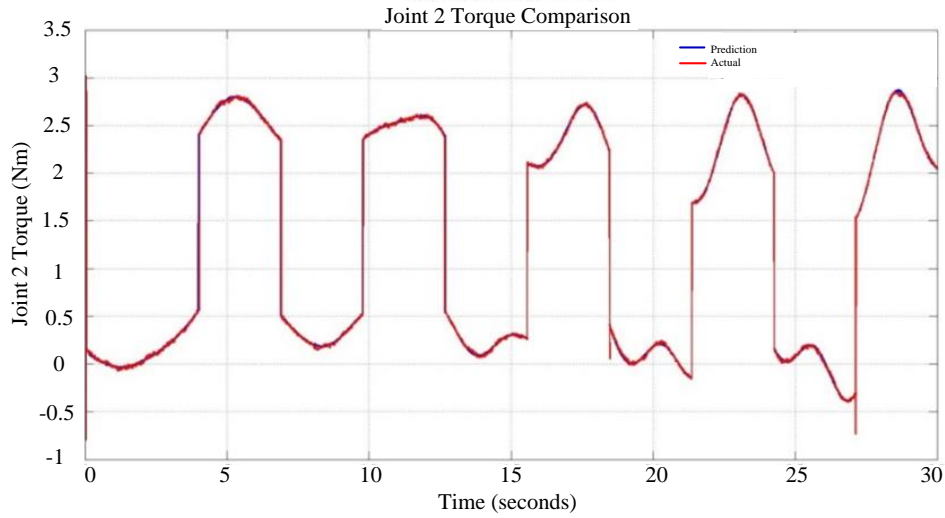


Figure 2.53: Joint 2 actual torque vs. predicted torque with online identification

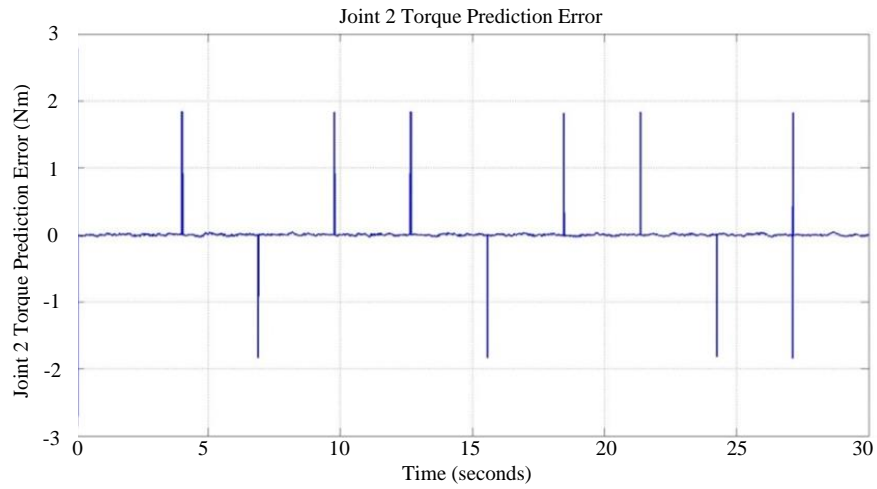


Figure 2.54: Joint 2 torque prediction error with online dynamic parameter identification

2.4.5 Offline Dynamic Model Identification with Neural Network based Compensator

It is observed from the online identification results that the friction parameters are time-varying. In fact, friction is the most complex component in the dynamic model of a robot manipulator. It is both uncertain and nonlinear, which are rather difficult to describe using explicit mathematical equations. Soft computing techniques, such as Fuzzy logic and Neural Network may be applied to describe this complicated input-output relationship after sufficient training.

In order to decrease the mismatch between predicted torque and measured torque, the offline identified dynamic parameters and a Neural Network based compensator are

applied here. This will provide a more accurate description of the manipulator dynamics. The input to the neural network is the joint state, which is given by the corresponding joint position q , velocity \dot{q} , and acceleration \ddot{q} . This velocity and acceleration information is reconstructed using the sliding mode based robust differentiator. The output of the Neural Network will be the torque difference between the actual joint torques calculated based on the joint actuator current, and the predicted joint torques. The Neural Network is trained using the inputs and the corresponding outputs. After training, it will act as a compensator to compensate for the torque difference which will be useful in subsequent chapters.

A Neural Network with one hidden layer and back propagation algorithm is used here. Since the joint torque residue calculation is based on the states of two joints, the compensation torques for joint 1 and joint 2 are considered simultaneously using one Neural Network, which is given in Figure 2.55. Eight hidden layer nodes are used. w_{ij} is the connection weight between the input layer and the hidden layer, while w_{jk} is the connection weight between the hidden layer and the output layer. $\Delta\tau_1$ is the torque compensation for joint 1, and $\Delta\tau_2$ is the torque compensation for joint 2.

The sigmoid function is used here

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.81)$$

The same offline identification experiment is done ten times, and the duration of each experiment is 100 seconds. With the backward propagation algorithm, after sufficient time, this Neural Network may repeat the torque residue accurately to an acceptable level.

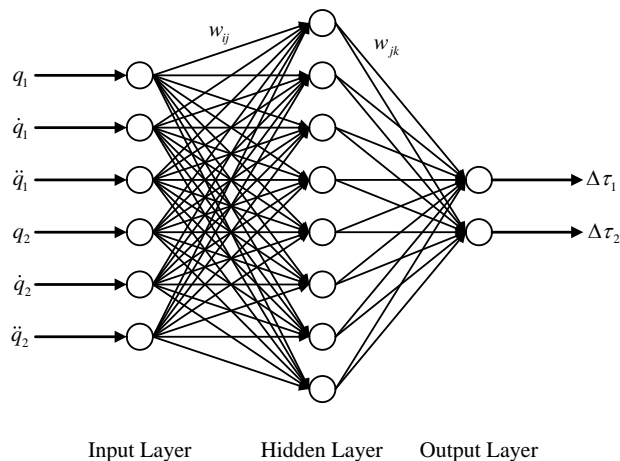


Figure 2.55: Neural Network-based torque compensator

Cross validation is done using experiment results from different rounds to further verify the effectiveness of the Neural Network in describing the unknown input and output relationship.

The validation process for offline identified dynamic parameters with the Neural Network based compensator is shown in Figure 2.56. Here, $\Delta\tau$ is the compensation torque vector, and $\hat{\tau}_n$ is the torque vector after compensation.

The actual joint torque and the predicted torque for the algorithms proposed in this subsection for each joint (offline identified dynamic parameters with Neural Network based compensator) are shown in Figure 2.57 and Figure 2.59, respectively. The joint torque prediction error for each joint is given in Figure 2.58 and Figure 2.60, respectively. It is seen that after compensation, the torque residues in both joints are reduced. The residues after compensation are bounded within a very low level.

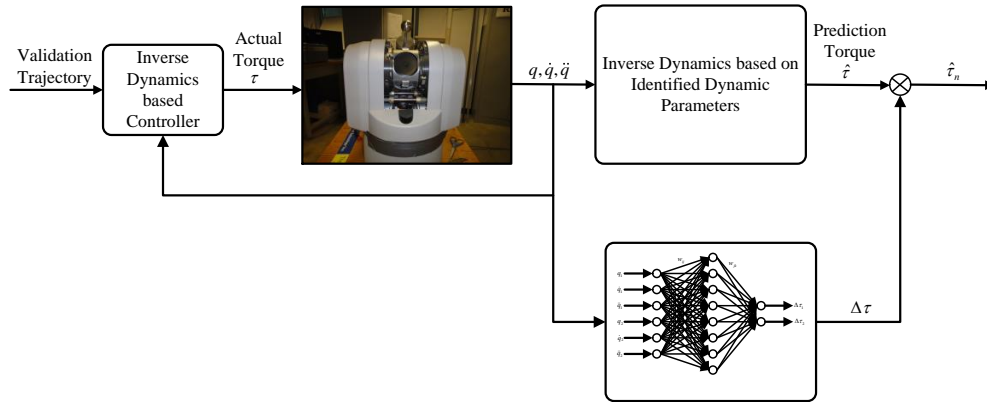


Figure 2.56: Scheme of the validation process of offline dynamic parameters with Neural Network-based compensator

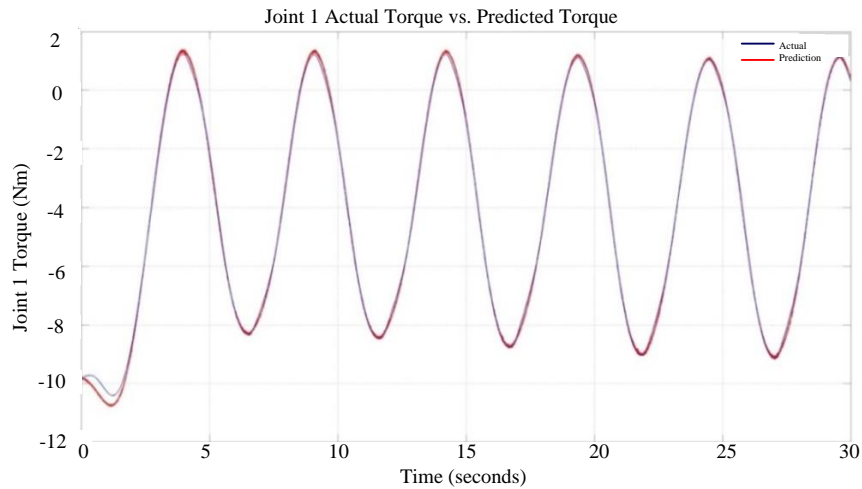


Figure 2.57: Joint 1 actual torque vs. predicted torque with offline identification (with torque compensator)

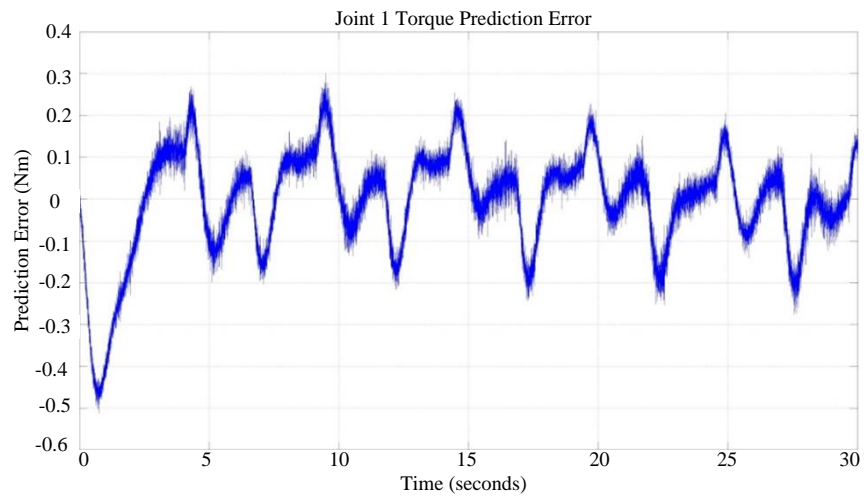


Figure 2.58: Joint 1 torque prediction error (with torque compensator)

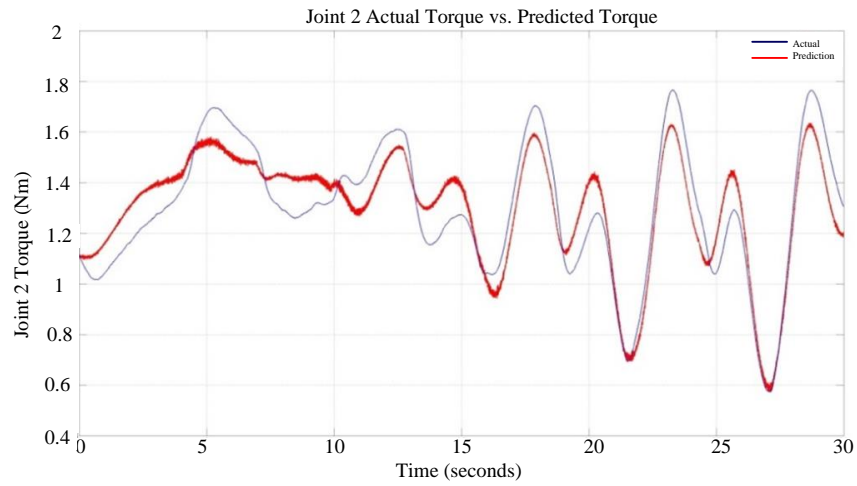


Figure 2.59: Joint 2 actual torque vs. predicted torque with offline identification (with torque compensator)

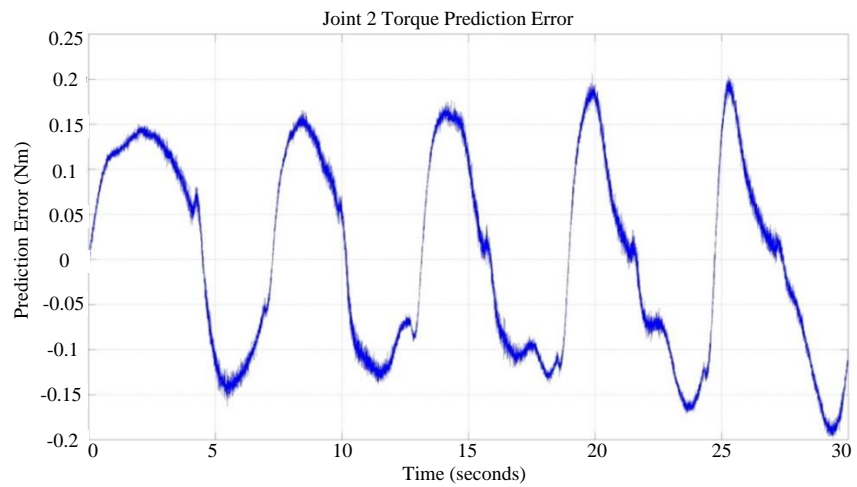


Figure 2.60: Joint 2 torque prediction error (with torque compensator)

2.4.6 Comparison with Conventional Differentiator based Identification

In this subsection, the dynamic parameters of the simplified two-link manipulator are identified using a conventional differentiator and a proper filter. This approach is widely used in literatures to identify a dynamic model for a robot manipulator.

Direct differentiation of the position signal is commonly used to calculate the velocity information:

$$\dot{q}(k) = \frac{q(k+1) - q(k)}{\Delta T} \quad (2.82)$$

where k is the time stamp and ΔT is the sampling period. Velocity information is not exact in this case, since $\dot{q}(k)$ calculated in this way is a better approximation of $\dot{q}(k + \frac{1}{2})$.

What is more, the velocity signal derived in this way is very noisy.

We have noticed that the direct differentiation of joint velocity gives really noisy acceleration results. In order to resolve this problem, a low-pass filter is used to reconstruct the acceleration information from the velocity information. From equation (2.29), we place a low-pass filter $F(s) = \frac{\omega_c}{s + \omega_c}$ at each side to have a filtered dynamic

representation given by

$$\tau_F = Y_F(q_F, \dot{q}_F)\theta \quad (2.83)$$

where τ_F represents the filtered torque measurements, $Y_F(q_F, \dot{q}_F)$ is the filtered regressor matrix, and q_F and \dot{q}_F are the low pass filtered versions of q and \dot{q} . By selecting the cut-off frequency of this low-pass filter to be located between the system bandwidth and the noise frequency, it is likely to reduce the noise effect on the identification accuracy. Here we use $\omega_c = 20 \text{ rad} / \text{s}$.

We notice that the acceleration information is not required in $Y_L(q_F, \dot{q}_F)$ due to the fact that it is represented by

$$\ddot{q}_F = \omega_c(\dot{q} - \dot{q}_F) \quad (2.84)$$

Using the same identification trajectory as in the offline identification algorithm, the filtered acceleration and the torque signal are given in Figure 2.61 and Figure 2.62, respectively.

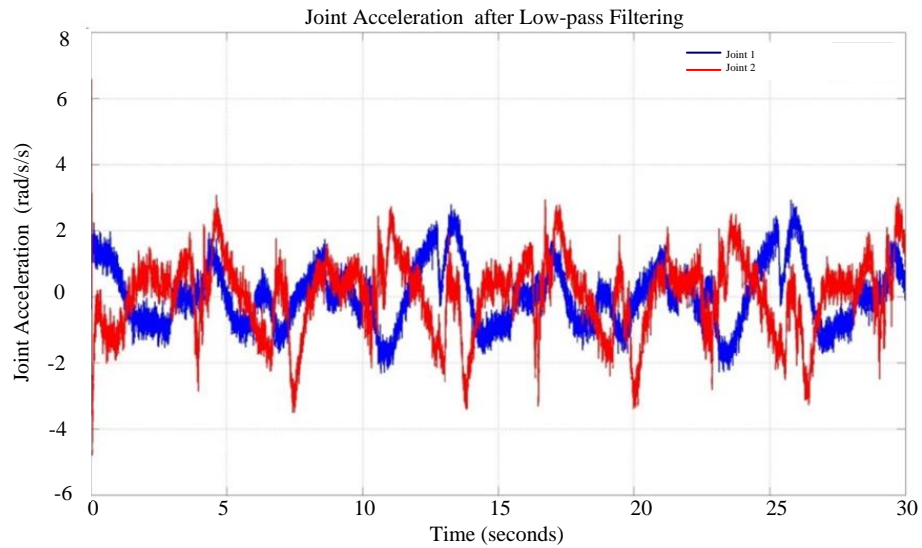


Figure 2.61: Joint acceleration trajectories after low-pass filtering

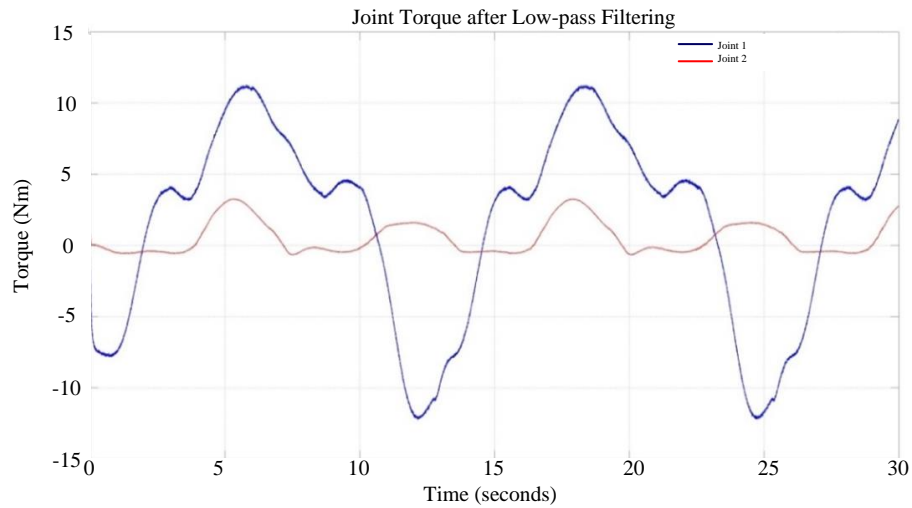


Figure 2.62: Joint torque trajectories after low-pass filtering

The low-pass filter does make the velocity and acceleration signals smoother. However, we are not certain yet whether this would give us more accurate identification results. This has to be verified using the same validation trajectories tracking problem as in the offline identification algorithm given in section 2.4.3. The reference trajectory tracking performance with the offline identified dynamic parameters in this subsection is given in Figure 2.63.

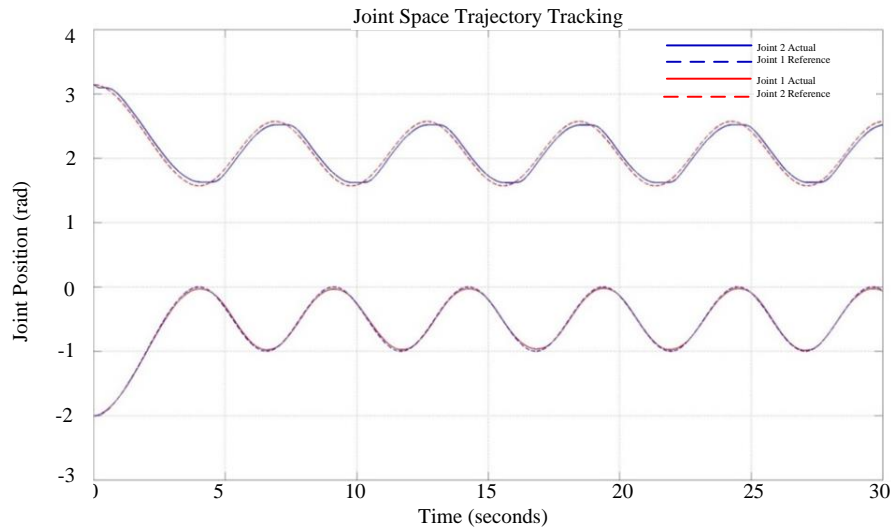


Figure 2.63: Joint space trajectory tracking (with dynamic parameters identified using filters)

We observe from Figure 2.61 that after filtering, the acceleration signal is still rather noisy while the torque signal is satisfactory. We can reduce the cut-off frequency of the low-pass filter in order manage this problem. In fact, the cut-off frequency used here is the one that provides best performance in the validation phase. High-order Butterworth filters may also be applied to realize a more comprehensive comparison.

The validation results are given in Figure 2.64 – Figure 2.67. It is found that for joint 1, there is no clear disadvantage when compared with robust differentiator based identification algorithm. However, the performance for joint 2 is not quite satisfactory. There are many more measurements with chattering in the torque prediction error when compared with the differentiator-based algorithm.

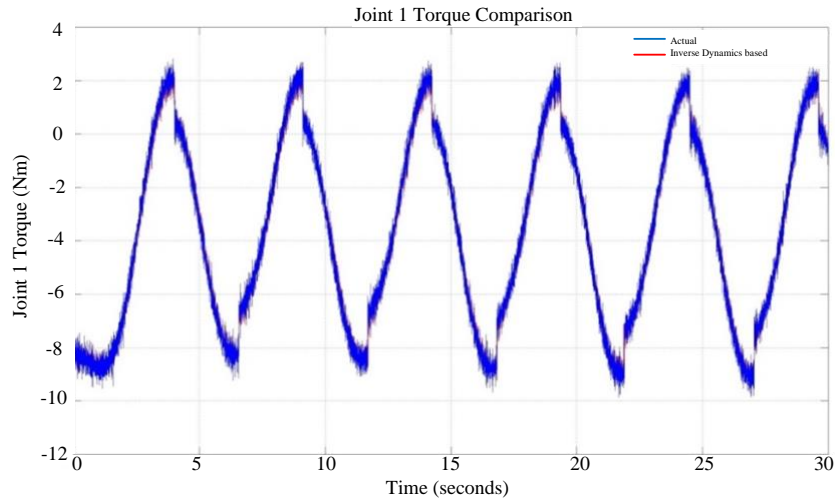


Figure 2.64: Joint 1 actual torque vs. predicted torque (with dynamic parameters identified using filters)

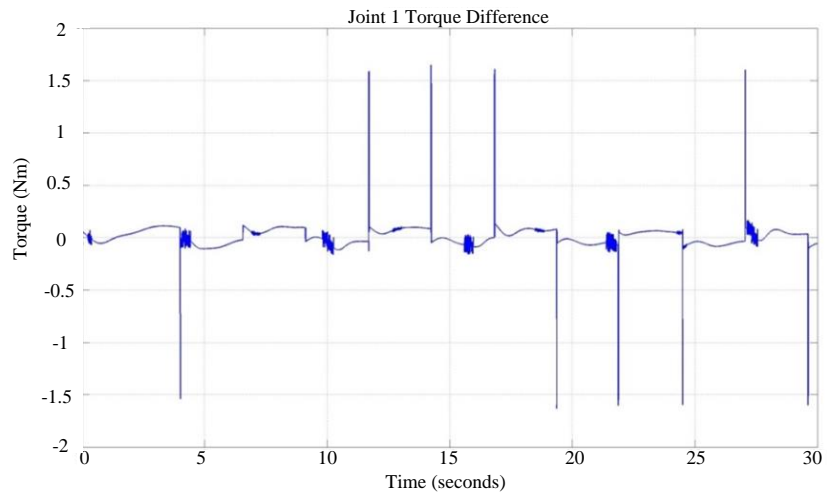


Figure 2.65: Joint 1 torque prediction error (with dynamic parameters identified using filters)

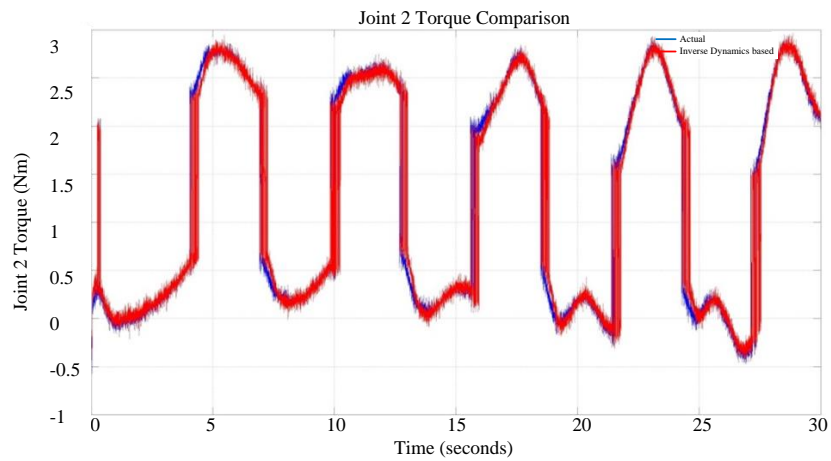


Figure 2.66: Joint 2 actual torque vs. predicted torque (with dynamic parameters identified using filters)

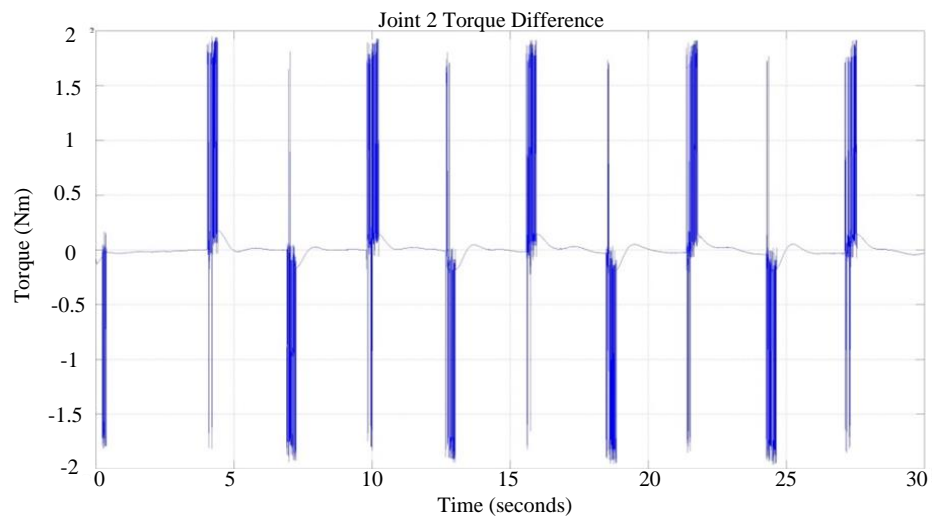


Figure 2.67: Joint 2 torque prediction error (with dynamic parameters identified using filters)

2.5 Summary

In this chapter, the dynamic model of a two-DOF robot manipulator is derived. To emphasize the problem under study, the four-DOF robot manipulator is simplified to a two-DOF planar manipulator by using mechanical as well as software methods. Due to this simplification, the software system is completely redesigned based on the xPC Target fast prototyping platform.

Because of the absence of an accurate dynamic model, which is required in the subsequent chapters, the dynamic model of the simplified manipulator is identified. The robust high order differentiator is applied to estimate the velocity and acceleration in each joint other than direct differentiation. This robust differentiator shows very good results in the presence of measurement noise when compared with a conventional differentiator.

Offline and online dynamic model identification algorithms are proposed, which use the estimated velocity and acceleration. The dynamic parameters obtained through model identification show satisfactory accuracy. However, due to the uncertainty in the friction component, they are not sufficiently accurate for delicate manipulation tasks.

A Neural Network-based torque compensator is proposed. After training, it may be used to accurately reproduce the torque prediction error.

This chapter lays the main foundation for subsequent chapters.

Chapter 3: Observers-based Estimation of Interaction Force

In constrained motion, the interaction force between the manipulator and the environment should be monitored. The interaction force is a straightforward way to describe the status of interaction. Interaction force is required in the algorithms of direct force control and indirect force control algorithms. Thus, the interaction force between the manipulator and the environment should be determined, which can be done by hardware or software.

The interaction force may be determined by using a force sensor mounted at the tip of the end-effector. However, there are some limitations with force sensors, such as the sensing noise, the self-variance, difficult mounting, and high cost. In view of these limitations, researchers have explored other approaches. For example sensor measurements may be passed through a filter to obtain smoother signals. A good approach to indirectly determine the interaction force is to reconstruct it by the use of observers. The interaction force estimation by observers is presented in this chapter.

Available algorithms of interaction force estimation are presented in this chapter. The actual interaction force during calibration experiments are sensed and compared with the reconstructed values based on the observers. The effectiveness of two algorithms is demonstrated and limitations of them are pointed out. An adaptive version high-order sliding mode-based interaction force observer is proposed. The proposed observer is shown to work effectively even when the bandwidth of system input is unknown. This new algorithm of interaction force estimation outperforms the two existing methods.

The estimated interaction force information is used in the impedance control algorithms that are proposed in the next chapter to shape the impedance of the manipulator during interaction.

3.1 Interaction Force Estimation

3.1.1 Cartesian Space Reference Trajectory Selection

In order to verify the effectiveness of the interaction force estimation algorithms, a proper trajectory has to be selected. When the trajectory tracking controllers are implemented to follow the reference trajectory, the robot end-effector will interact with environment. We will place a force sensor at the end-effector to measure the interaction

force during contact. This force sensor will only be used for calibration purposes. The experimental setup for calibration is shown in Figure 3.1.

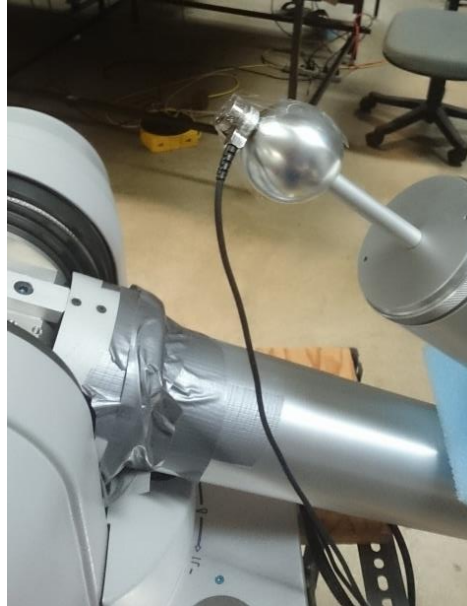


Figure 3.1: Interaction force estimation validation experimental setup

The coordinate system of the force sensor as defined by the manufacturer in software is shown in Figure 3.2. Data logging of the force sensor is done using a separate interface in the host PC running Microsoft Windows.



Figure 3.2: Coordinate system of the force sensor

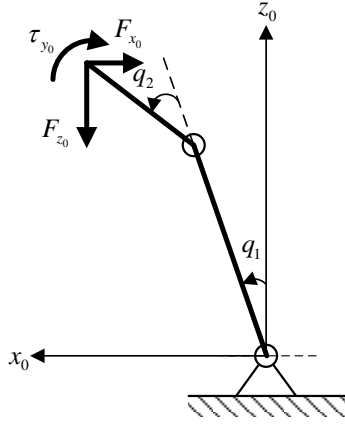


Figure 3.3: Two degree-of-freedom WAM under external force

Note that, due to the specific shape of the force sensor, precise point contact cannot be established with the obstacle. There will be both an interaction force and an interaction torque between the force sensor and the obstacle. Hence, there are three components in the wrench vector of the end effector. The static relationship between this external interaction force and the induced joint torque is given by

$$\begin{bmatrix} \tau_{1e} \\ \tau_{2e} \end{bmatrix} = J_c^T \begin{bmatrix} F_{x_0} \\ F_{z_0} \\ \tau_{y_0} \end{bmatrix} \quad (3.1)$$

where $J_c^T = \begin{bmatrix} l_1 \cdot \cos q_1 + l_2 \cdot \cos(q_1 + q_2) & -l_1 \cdot \sin q_1 - l_2 \cdot \sin(q_1 + q_2) & 1 \\ l_2 \cdot \cos(q_1 + q_2) & -l_2 \cdot \sin(q_1 + q_2) & 1 \end{bmatrix}$, $[\tau_{1e} \quad \tau_{2e}]^T$ is the induced joint torque vector due to interaction with obstacle, J_c is the Jacobian taking the torque component into consideration, and $[F_{x_0} \quad F_{z_0} \quad \tau_{y_0}]^T$ is the external force vector.

When WAM directly interacts with the obstacle without the force sensor mounted at the end-effector, τ_{y_0} will be zero because of the smoothness of the end effector. When $\tau_{y_0} = 0$, the subsequent analysis is greatly simplified. However, in the calibration phase, due to the shape of the force sensor, it is likely that $\tau_{y_0} \neq 0$. Since the Jacobian J_c in this case is not square, the vector $[F_{x_0} \quad F_{z_0} \quad \tau_{y_0}]^T$ cannot be determined explicitly from equation (3.1). Psuedo-inverse of J_c may be used to obtain a feasible solution for $[F_{x_0} \quad F_{z_0} \quad \tau_{y_0}]^T$, which is given by

$$\begin{bmatrix} F_{x_0} \\ F_{z_0} \\ \tau_{y_0} \end{bmatrix} = J_c (J_c^T J_c)^{-1} \begin{bmatrix} \tau_{1e} \\ \tau_{2e} \end{bmatrix} = J_c^+ \begin{bmatrix} \tau_{1e} \\ \tau_{2e} \end{bmatrix} \quad (3.2)$$

where J_c^+ is the right pseudo-inverse of J_c and it is defined as

$$J_c^+ = J_c (J_c^T J_c)^{-1} \quad (3.3)$$

However, the solution in equation (3.2) is not unique. The general solution is given by

$$\begin{bmatrix} F_{x_0} \\ F_{z_0} \\ \tau_{y_0} \end{bmatrix} = J_c^+ \begin{bmatrix} \tau_{1e} \\ \tau_{2e} \end{bmatrix} + (I - J_c^+ J_c) b \quad (3.4)$$

where b is a 3×1 vector. It is very easy to verify that the solution given by equation (3.4) satisfies equation (3.1). Thus it is impossible to determine the value of $\begin{bmatrix} F_{x_0} & F_{z_0} & \tau_{y_0} \end{bmatrix}^T$.

One possible solution is to attach a smooth surface to the force sensor or to control the motion of the end-effector to be strictly perpendicular to some flat surface so that one of three components in vector $\begin{bmatrix} F_{x_0} & F_{z_0} & \tau_{y_0} \end{bmatrix}^T$ becomes zero. In this way, the Jacobian J_c can be simplified to a square matrix, and the external force can be determined explicitly. However, in practice this assumption is not reasonable. The accuracy of the force sensor will be degraded if a cover is attached to it. Controlling the motion of the end effector to some specific pose is possible in steady state, while it is difficult in the transitional phase.

A compromise is made here, which has been made by others as well [52]. Here we use τ_{y_0} from the force sensor measurement. Then we have two ways to verify the accuracy of the estimation algorithms for external force. The first one is to verify whether the force sensor measurements $\begin{bmatrix} F_{mx_0} & F_{mz_0} & \tau_{my_0} \end{bmatrix}^T$ satisfy equation (3.5).

Here $\begin{bmatrix} \tau_{1e} \\ \tau_{2e} \end{bmatrix}$ will be reconstructed from different interaction force observers which are

detailed in the subsections of this section. The advantage of this choice is its simplicity. However, the obvious disadvantage is that we cannot verify the accuracy of F_{x_0} and F_{z_0}

individually. F_{x_0} and F_{z_0} are important in the subsequent sections since they are used in impedance control algorithms. It is desirable that we can verify the accuracy of them individually. We have

$$\begin{aligned} \begin{bmatrix} \tau_{1e} \\ \tau_{2e} \end{bmatrix} &= J_c^T \begin{bmatrix} F_{mx_0} \\ F_{mz_0} \\ \tau_{my_0} \end{bmatrix} \\ &= \begin{bmatrix} l_1 \cdot \cos q_1 + l_2 \cdot \cos(q_1 + q_2) & -l_1 \cdot \sin q_1 - l_2 \cdot \sin(q_1 + q_2) & 1 \\ l_2 \cdot \cos(q_1 + q_2) & -l_2 \cdot \sin(q_1 + q_2) & 1 \end{bmatrix} \begin{bmatrix} F_{mx_0} \\ F_{mz_0} \\ \tau_{my_0} \end{bmatrix} \end{aligned} \quad (3.5)$$

Thus we will explore the second choice. Here we rewrite equation (3.5) as

$$\begin{aligned} \begin{bmatrix} \tau_{1e} - \tau_{my_0} \\ \tau_{2e} - \tau_{my_0} \end{bmatrix} &= J^T \begin{bmatrix} F_{x_0} \\ F_{z_0} \end{bmatrix} \\ &= \begin{bmatrix} l_1 \cdot \cos q_1 + l_2 \cdot \cos(q_1 + q_2) & -l_1 \cdot \sin q_1 - l_2 \cdot \sin(q_1 + q_2) \\ l_2 \cdot \cos(q_1 + q_2) & -l_2 \cdot \sin(q_1 + q_2) \end{bmatrix} \begin{bmatrix} F_{x_0} \\ F_{z_0} \end{bmatrix} \end{aligned} \quad (3.6)$$

where J is the same square Jacobian matrix as the one defined in Chapter 2. As long as the Jacobian matrix is invertible, $\begin{bmatrix} F_{x_0} & F_{z_0} \end{bmatrix}^T$ can be determined as

$$\begin{bmatrix} F_{x_0} \\ F_{z_0} \end{bmatrix} = \begin{bmatrix} l_1 \cdot \cos q_1 + l_2 \cdot \cos(q_1 + q_2) & -l_1 \cdot \sin q_1 - l_2 \cdot \sin(q_1 + q_2) \\ l_2 \cdot \cos(q_1 + q_2) & -l_2 \cdot \sin(q_1 + q_2) \end{bmatrix}^{-1} \begin{bmatrix} \tau_{1e} - \tau_{my_0} \\ \tau_{2e} - \tau_{my_0} \end{bmatrix} \quad (3.7)$$

The invertibility of the Jacobian matrix J can be guaranteed by proper trajectory planning to make sure that the second joint of the manipulator does not pass through a singularity point. Unfortunately, due to the initial position of joint 2 (“home position” specified by the manufacturer) is 3.1428 rad, which is very close to the singularity point 0 rad (or 2π). In order to avoid this point, a piece of foam is used to lift link 2 away from its original home position, in the initial phase. When WAM is left at the home position, the foam is removed so that it will not constraint the motion of the manipulator. This is shown in Figure 2.1. With the foam, the new home position of joint 2 will be $2\pi/3$. For joint 1, the same reference trajectory as described by equation (2.45) is used:

$$q_1(t) = \begin{cases} -0.0627t^3 + 0.3760t^2 - 2.0051 & 0 \leq t \leq 4 \\ 0.5 \sin(1.2270t + 1.5734) - 0.5019 & t \geq 4 \end{cases} \quad (2.45)$$

In order to avoid the singularity point, joint 2 will be commanded to move to $q_2 = \pi/2$, and then commanded to execute a periodic motion. The joint limit and the Cartesian space motion limit as given by equation (2.5) and equation (2.6) have to be taken into consideration in the trajectory design. The expression for the new reference trajectory of joint 2 is given by

$$q_2(t) = \begin{cases} 0.0164t^3 - 0.0982t^2 + 2.0944 & 0 \leq t \leq 4 \\ \frac{\pi}{6} \sin(0.6124t + 2.2629) + 2.0944 & t \geq 4 \end{cases} \quad (3.8)$$

The joint space position and velocity reference trajectories are shown in Figure 3.4 and Figure 3.5, respectively.

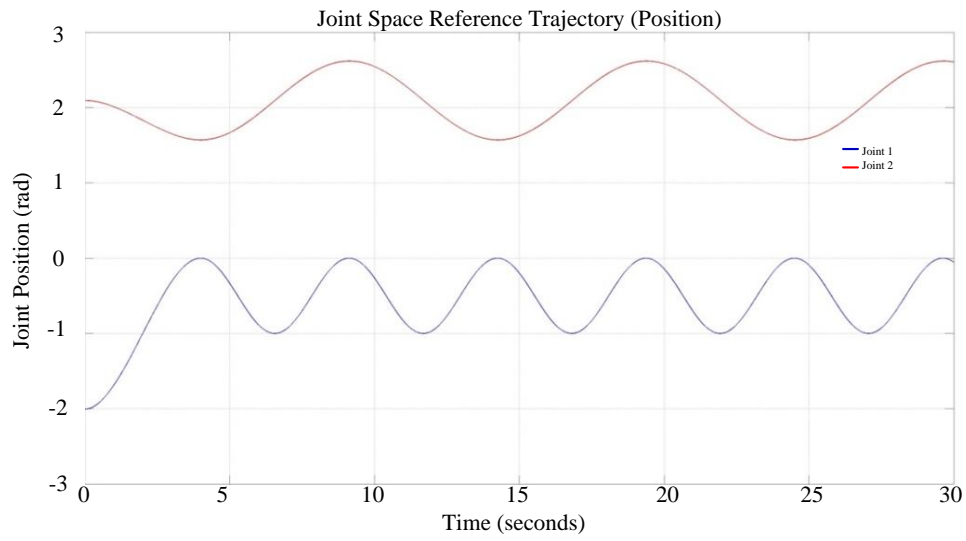


Figure 3.4: Reference trajectories of joint 1 and joint 2 (positions)

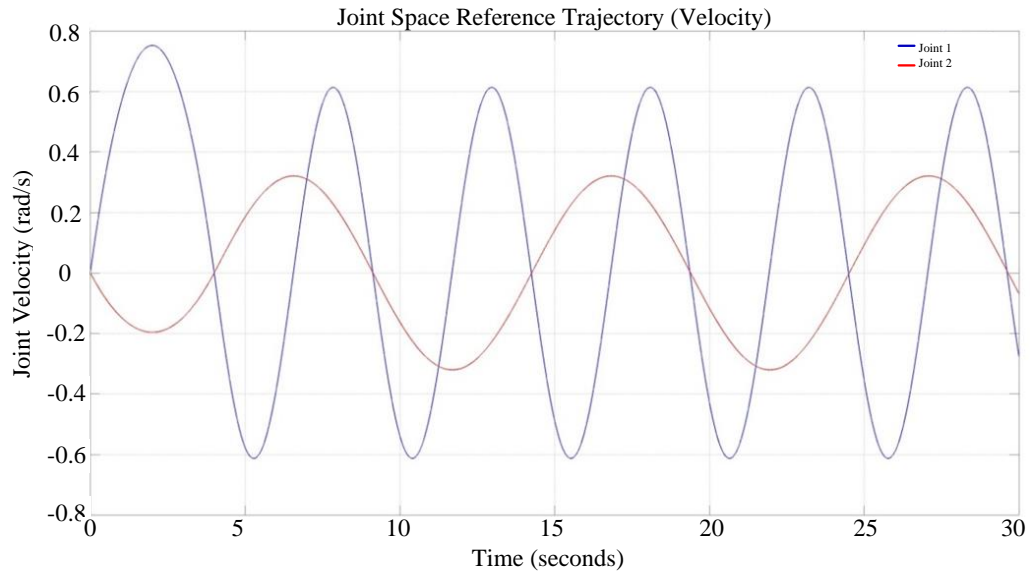


Figure 3.5: Reference trajectories of joint 1 and joint 2 (velocities)

The corresponding Cartesian space trajectories of the joint space reference trajectories are given in Figure 3.6. In the following validation experiments, joint space inverse dynamics-based controllers are used for tracking control, even though such algorithms are effective only when the manipulator end-effector is not constrained. Some soft cardboard is placed in front of WAM so that the interaction may happen when WAM attempts to track the new joint space reference trajectory. The cardboard is put perpendicular to the ground and x axis.

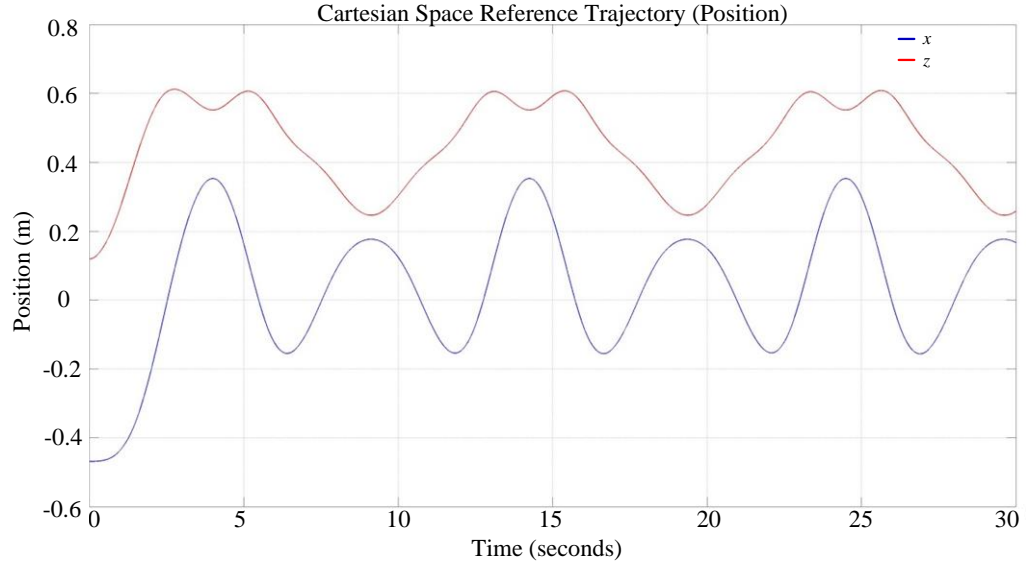


Figure 3.6: Cartesian space position reference trajectories

After the initial preparations, various algorithms of interaction force estimation are used to estimate the interaction force. The effectiveness of the estimation algorithms are discussed and compared and the best algorithm is used in the subsequent sections for impedance control. The validation scheme of the interaction force estimation is given in Figure 3.7.

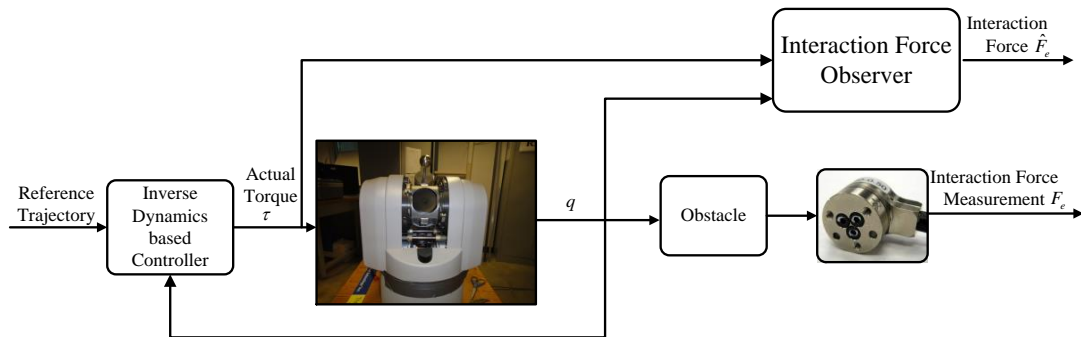


Figure 3.7: The validation scheme of the interaction force estimation

3.1.2 Interaction Force Calculation Using Identified Dynamic Model

From equation (2.28) of joint space dynamics, we can rewrite the expression for external interaction force as

$$\hat{F}_e = J^{-T}(q) \left\{ \tau - [\hat{M}(q)\ddot{q} + \hat{C}(q, \dot{q})\dot{q} + \hat{F}_v\dot{q} + \hat{F}_c \operatorname{sgn}(\dot{q}) + \hat{G}(q)] \right\} \quad (3.9)$$

The external interaction force can be reconstructed using the torques applied to the joint actuators and the corresponding joint motion. In order to improve the force estimation accuracy, the Neural Network-based dynamic model compensator introduced in section 2.4.5 is used here.

Two different approaches for the calculation of q , \dot{q} , and \ddot{q} are introduced now. The first one is based on direct differentiation of the position measurement with low-pass filtering, as discussed in Chapter 2, while the second one is based on the robust sliding mode differentiator.

The trajectories of joint space and Cartesian space are given by Figure 3.8 and Figure 3.9, respectively. The force estimation results and the corresponding estimation errors using velocity and acceleration information obtained by direct differentiation are given in Figure 3.10 – Figure 3.13. We observe that the use of the velocity and acceleration information obtained by direct differentiation and filtering leads to unsatisfactory results for interaction force estimation. The errors in interaction force estimation are varying and uncertain with large magnitudes. It may be concluded that the interaction force reconstruction in this manner may not be applicable to the impedance control algorithms that are developed in the next chapter.

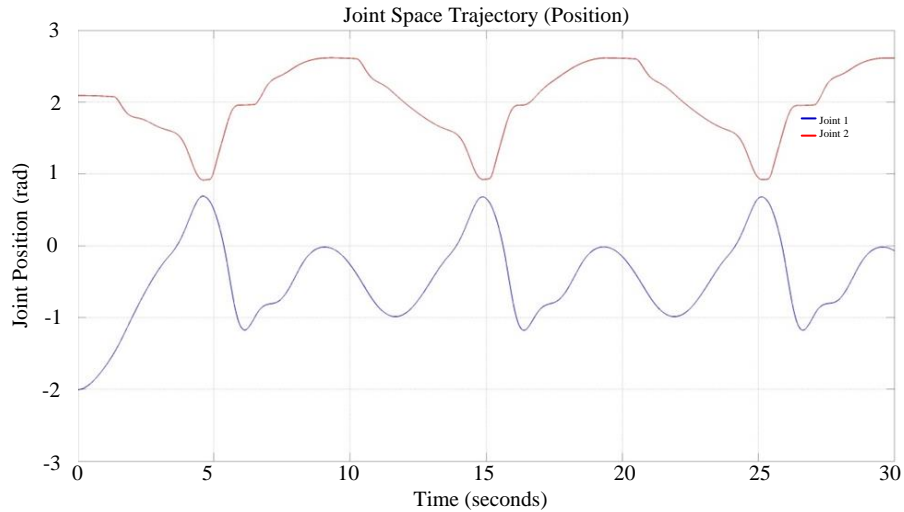


Figure 3.8: Actual joint space position trajectories (with inverse dynamics using identified dynamic parameters and direct differentiation)

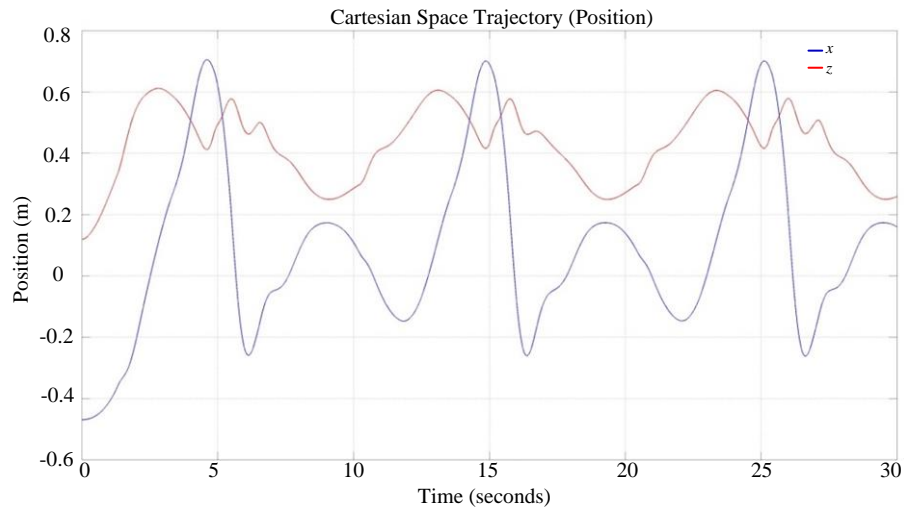


Figure 3.9: Actual Cartesian space trajectories (with inverse dynamics using identified dynamic parameters and direct differentiation)

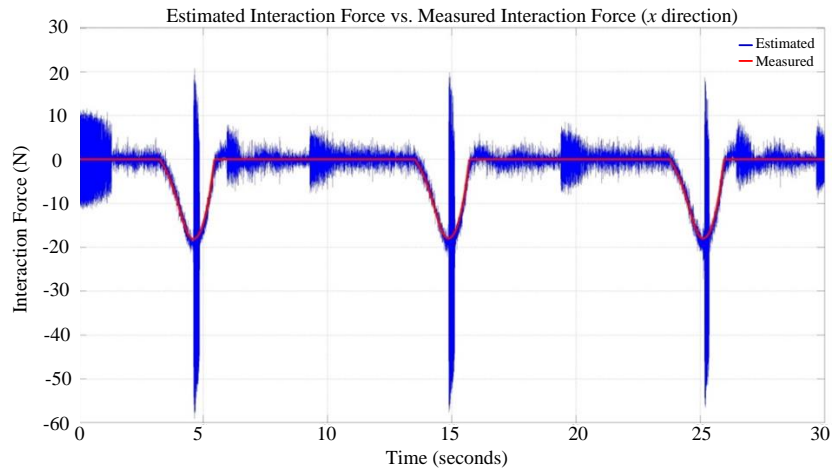


Figure 3.10: Estimated force vs. measured force (x direction with inverse dynamics using identified dynamic parameters and direct differentiation)

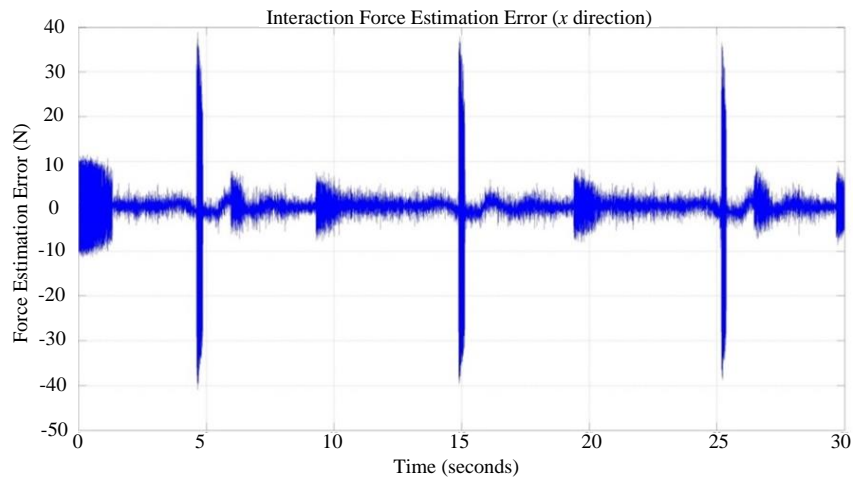


Figure 3.11: Force estimation error (x direction with inverse dynamics using identified dynamic parameters and direct differentiation)

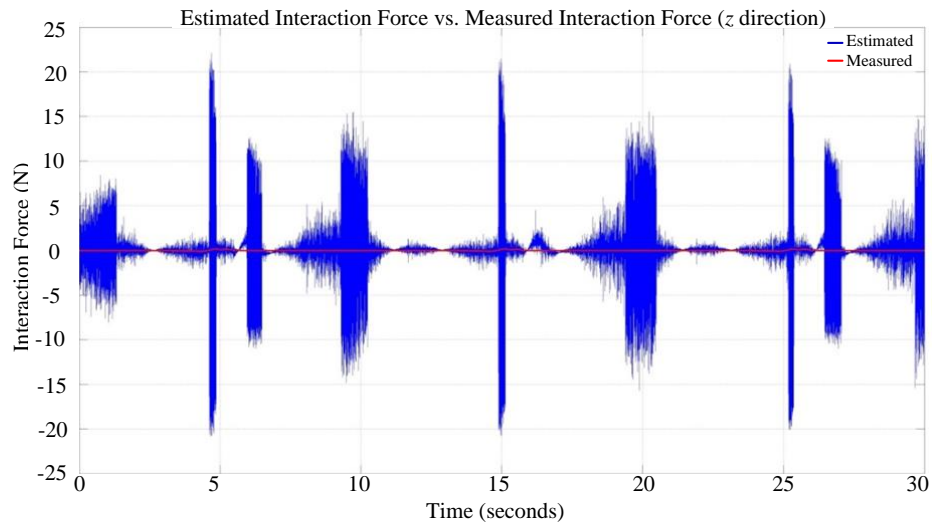


Figure 3.12: Estimated force vs. measured force (z direction with inverse dynamics using identified dynamic parameters and direct differentiation)

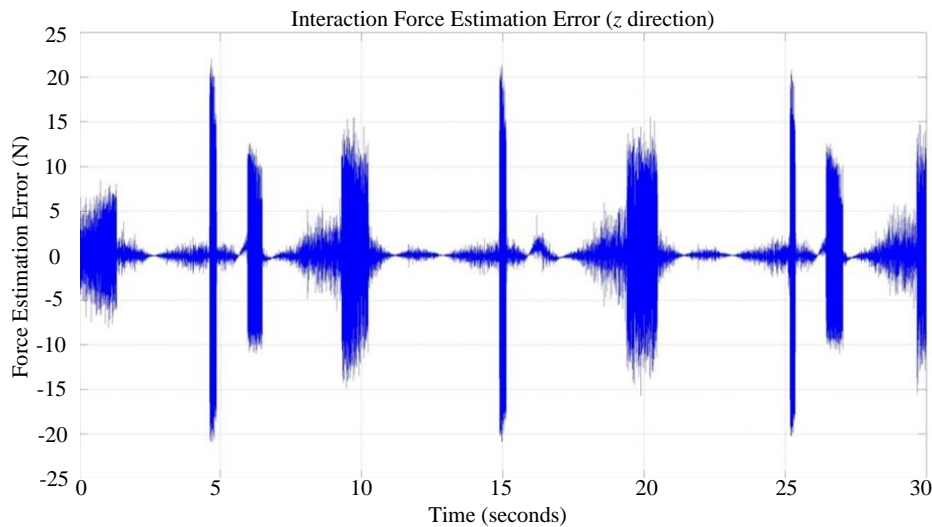


Figure 3.13: Force estimation error (z direction with inverse dynamics using identified dynamic parameters and direct differentiation)

The second approach for velocity and acceleration calculation is the robust differentiator used in Chapter 2. The reconstructed velocity information is used for the inverse dynamics-based trajectory tracking controller. The actual trajectories in joint space and Cartesian space are given in Figure 3.14 and Figure 3.15, respectively. The results of interaction force estimation and the estimation errors resulting from this robust differentiator are given in Figure 3.16 – Figure 3.19. We can observe from these figures that the estimation results are much better than the corresponding ones for the first approach. The estimation errors have a much smaller range in each direction compared to the ones estimated using the first approach. Meanwhile, the estimation error for the x

direction is smaller compared with that for the z direction. A possible reason for this is that the interaction force for the x direction is mainly due to the stiffness of the environment, and hence the estimation of this interaction force is less sensitive to velocity reconstruction noise. The interaction force in the z direction arises from the damping (or friction) when the force sensor touches the cardboard. This z direction estimation is more sensitive to velocity reconstruction noise. In our project, the x direction interaction force is critical since when the end-effector touches the human body. We must guarantee that no excessive interaction force is present in this direction. Also, we use the reconstructed velocity information in the inverse dynamics-based trajectory tracking controller. This is an important feature because, as explored in the Chapter 4, velocity information may be required in the design of the interaction control.

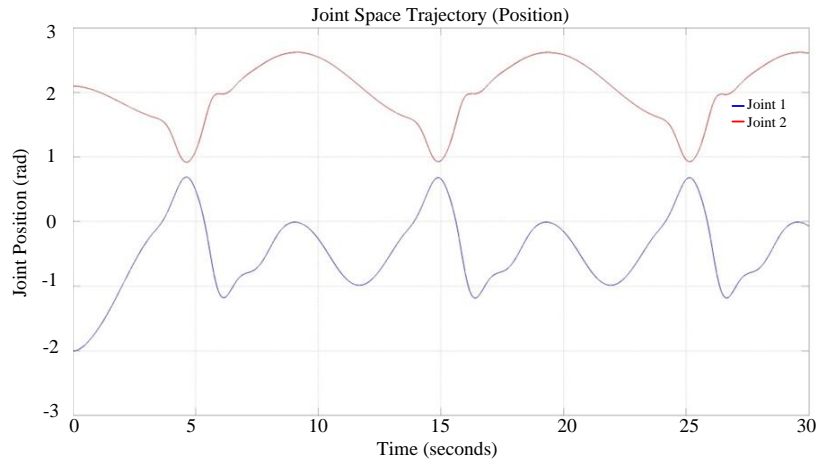


Figure 3.14: Actual joint space position trajectories (with inverse dynamics using identified dynamic parameters and robust differentiator)

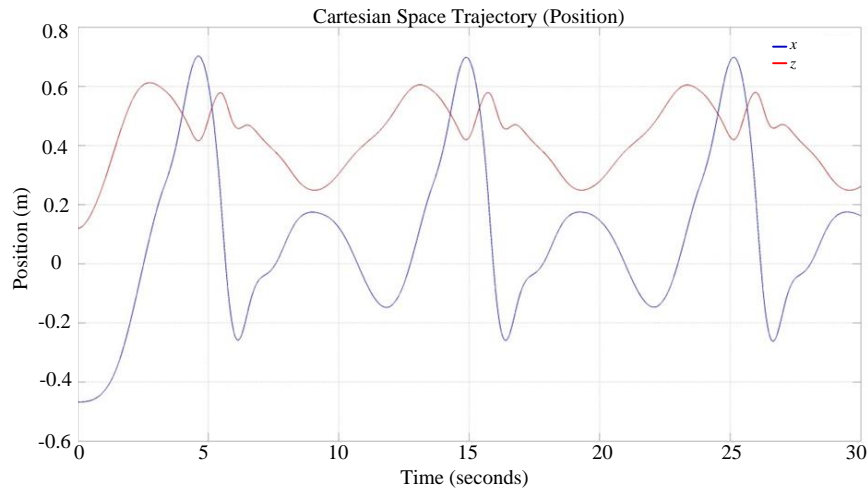


Figure 3.15: Actual Cartesian space position trajectories (with inverse dynamics using identified dynamic parameters and robust differentiator)

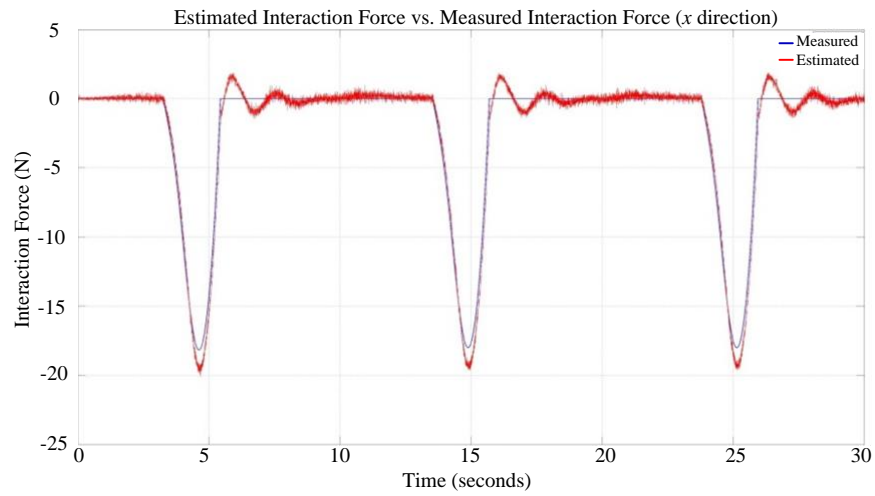


Figure 3.16: Estimated force vs. measured force (x direction with inverse dynamics using identified dynamic parameters and robust differentiator)

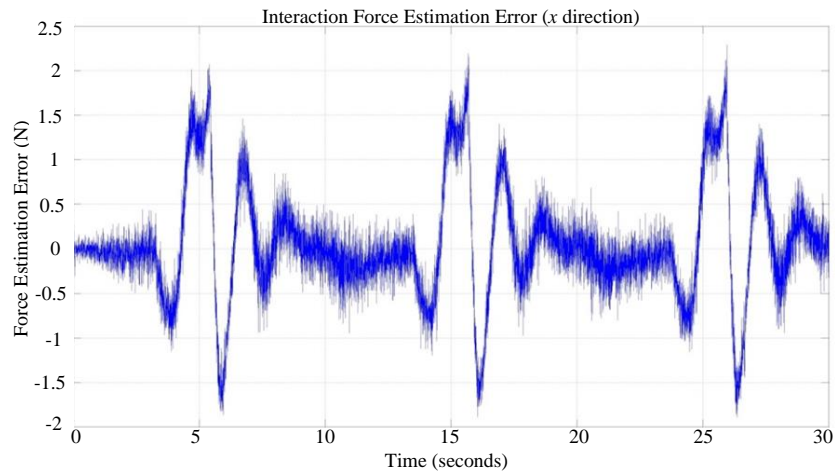


Figure 3.17: Force estimation error (x direction with inverse dynamics using identified dynamic parameters and robust differentiator)

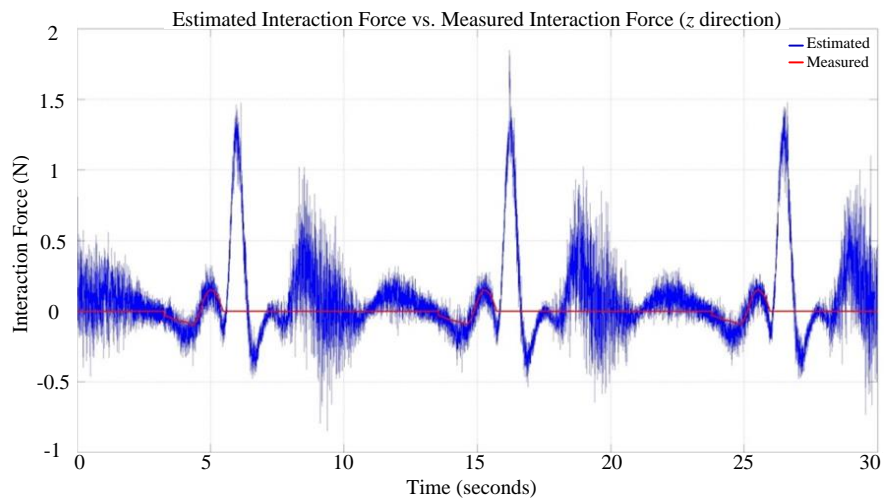


Figure 3.18: Estimated force vs. measured force (z direction with inverse dynamics using identified dynamic parameters and robust differentiator)

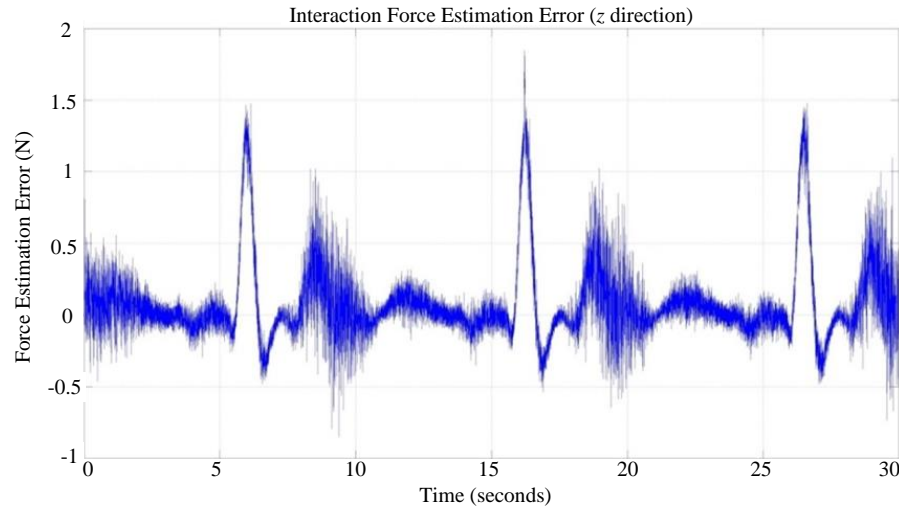


Figure 3.19: Force estimation error (z direction with inverse dynamics using identified dynamic parameters and robust differentiator)

3.1.3 Interaction Force Estimation Using Second-order Sliding Mode Observer (SOSMO)

The simultaneous reconstruction of the state and unknown inputs using a second-order sliding mode observers has been used, extended and evaluated by others. The most recent work is [109], in which the algorithms for state estimation and unknown input reconstruction has been extended to the multi-DOF case. This algorithm is implemented in the present thesis to act as a control group for other algorithms developed in the subsequent subsections. Further details are found in [109]. The robot dynamic model in the joint space, as described by equation (2.28), should be represented in a state-space form for the following development. For this purpose, let $x_1 = q$, $x_2 = \dot{q}$, $u = \tau$. Then, the joint space robot dynamic equation may be represented in state-space form as,

$$\dot{x}_1 = x_2 \quad (3.10)$$

$$\dot{x}_2 = f(t, x_1, x_2, u) + \xi(t, x_1, x_2, u) \quad (3.11)$$

$$y = x_1 \quad (3.12)$$

$$f(t, x_1, x_2, u) = -M^{-1}(x_1)[C(x_1, x_2)x_2 + F_v x_2 + F_c \text{sgn}(x_2) + G(x_1) - u] \quad (3.13)$$

$$\xi(t, x_1, x_2, u) = -M^{-1}(x_1)[\Delta \tau + J^T(x_1)F_e] \quad (3.14)$$

where $x_1 \in R^2$ is the joint position encoder reading; $x_2 \in R^2$ is the joint velocity vector of the manipulator; $f(t, x_1, x_2, u) \in R^2$ represents the nominal dynamics of the mechanical

system; $\xi(t, x_1, x_2, u)$ is the combination of the model inaccuracy induced terms $\Delta\tau$ and the external interaction force F_e . $f(t, x_1, x_2, u) \in R^2$ and $\xi(t, x_1, x_2, u)$ are given by

$$f(t, x_1, x_2, u) = \begin{bmatrix} f_1(t, x_1, x_2, u) \\ f_2(t, x_1, x_2, u) \end{bmatrix}, \quad \xi(t, x_1, x_2, u) = \begin{bmatrix} \xi_1(t, x_1, x_2, u) \\ \xi_2(t, x_1, x_2, u) \end{bmatrix} \quad (3.15)$$

The simultaneous state and unknown input observer for this mechanical system is given by

$$\dot{\hat{x}}_1 = \hat{x}_2 + z_1 \quad (3.16)$$

$$\dot{\hat{x}}_2 = f(t, x_1, \hat{x}_2, u) + z_2 \quad (3.17)$$

with $\hat{x}_1 \in R^2$, $\hat{x}_2 \in R^2$. The i -th components ($i = 1, 2$) of z_1 and z_2 are defined as

$$z_{1i} = \lambda_i \cdot |x_{1i} - \hat{x}_{1i}|^{1/2} \cdot \text{sign}(x_{1i} - \hat{x}_{1i}) \quad (3.18)$$

$$z_{2i} = \alpha_i \cdot \text{sign}(x_{1i} - \hat{x}_{1i}) \quad (3.19)$$

where $\hat{x}_1(0) = x_1(0)$, $\hat{x}_2(0) = x_2(0)$ is a reasonable assumption since the robot initial joint positions and velocities are known in practice. The trajectory of the observer is understood in the Filippov sense. λ_i is the observer gain, which is to be decided.

Define the state estimation errors as $\tilde{x}_1 = x_1 - \hat{x}_1$, $\tilde{x}_2 = x_2 - \hat{x}_2$, then the error dynamics are described by

$$\dot{\tilde{x}}_1 = \tilde{x}_2 - \lambda |\tilde{x}_1|^{1/2} \text{sign}(\tilde{x}_1) \quad (3.20)$$

$$\dot{\tilde{x}}_2 = F(t, x_1, x_2, \hat{x}_2, u) - \alpha \text{sign}(\tilde{x}_1) \quad (3.21)$$

$F(t, x_1, x_2, \hat{x}_2, u) \in R^2$ is defined as

$$F(t, x_1, x_2, \hat{x}_2, u) = f(t, x_1, x_2, u) - f(t, x_1, \hat{x}_2, u) + \xi(t, x_0, x_2, u) \quad (3.22)$$

This uncertainty item is bounded and describes the size of the uncertainty. It is given as

$$|F_i(t, x_1, x_2, \hat{x}_2, u)| < f_i^+ \quad (3.23)$$

It is seen that $|F_i(t, x_1, x_2, \hat{x}_2, u)|$ is configuration dependent. Proper control algorithms should be used to guarantee the validity of this inequality during experiments. After experimentation, this inequality should also be verified offline to check whether it is violated. We can select larger values for f_i^+ . However, a too large f_i^+ will result in poor performance, since the controllers and observers designed might be too conservative.

The observer gains in equation (3.18) and (3.19) should be chosen according to the following equations:

$$\alpha_i > f_i^+ \quad (3.24)$$

$$\lambda_i > \sqrt{\frac{2}{\alpha_i - f_i^+}} \cdot \frac{(\alpha_i + f_i^+)(1 + p_i)}{1 - p_i} \quad (3.25)$$

where $p_i \in (0,1)$. This second-order observer based on the super-twisting algorithm will drive states estimation error to converge in finite time T_i . Let $T = \max(T_i)$. Then after T , all the states will converge to the actual ones regardless of external disturbance or uncertainty. In this case, equation (3.21) may be rewritten as

$$\begin{aligned} \dot{\tilde{x}}_2 &= F(t, x_1, x_2, \hat{x}_2, u) - \alpha \text{sign}(\tilde{x}_1) \\ &= f(t, x_1, x_2, u) - f(t, x_1, \hat{x}_2, u) + \xi(t, x_0, x_2, u) - \alpha \text{sign}(\tilde{x}_1) = 0 \end{aligned} \quad (3.26)$$

When the state estimation errors converge to zero, $\hat{x}_2 \rightarrow x_2$, and

$$\xi(t, x_0, x_2, u) - \alpha \text{sign}(\tilde{x}_1) = 0 \quad (3.27)$$

Due to the limited bandwidth of the plant, the actual control action cannot switch infinitely fast. The plant will have some low-pass effect toward this ideally infinitely fast switching item. This item is called the equivalent output injection, which contains important information about system uncertainty. The unknown input may also be reconstructed as

$$\xi(t, x_1, x_2, u) = \alpha_i \cdot \text{sign}(\tilde{x}_{1i})_{eq} \quad (3.28)$$

which is obtained by passing \tilde{x}_{1i} through a low-pass filter. The filter may be represented as

$$T_i(\dot{\bar{x}}_{1i}) + (\bar{x}_{1i}) = \alpha_i \text{sign}(\tilde{x}_{1i})_{eq} \quad (3.29)$$

where T_i is the filter time constant, a critical parameter that has to be selected wisely. There has to be some compromise between the smoothness of the reconstructed interaction force signal and the time lag caused by this low-pass filtering process. It has been proved that

$$\lim_{T_i \rightarrow 0, T_s/T_i \rightarrow 0} (\bar{x}_{1i})_{eq} = (x_{1i})_{eq} \quad (3.30)$$

where T_s is the sampling period of the experimental system. In practice, T_i is selected as

$$T_s \ll T_i \ll 1 \quad (3.31)$$

The interaction force may be reconstructed as

$$F_e = -J^{-T}(x_1) \left[M^{-1}(x_1) \left[\alpha \cdot \text{sign}(\tilde{x}_1)_{eq} \right] + \Delta \tau \right] \quad (3.32)$$

Since the sampling time T_s of our experimental setup is 0.002 s, we can select $T_i = 0.01$ s. The corresponding results of interaction force estimation are given in Figure 3.20 – Figure 3.23. It is observed that the reconstructed interaction force is noisy and there is some chatter. However, the estimation drift is small, which means that the time lag induced by the low-pass filter is small.

With $T_i = 0.1$ s, the corresponding force reconstruction results are given in Figure 3.24 – Figure 3.27. It is seen that chattering is significantly reduced. However, there is clear drift in the force estimation. This is due to the compromise of the smoothness between the estimated force and the drift between the actual interaction force and the estimated one. It is seen from the Figure 3.25 that due to this drift, the force estimation error is not adequately small.

Based on the above observation, by aiming for a value between $T_i = 0.01$ s and $T_i = 0.1$ s, the value $T_i = 0.05$ s is selected. The force reconstruction results are given in Figure 3.28 – Figure 3.31. The smoothness and drift in the reconstructed force signal is between the corresponding ones in $T_i = 0.01$ s and $T_i = 0.1$ s. Still a worse performance is observed in the z direction than in the x direction. We will always observe a worse performance of the interaction force observation in the z direction. Another reason for this is that the force sensor mounting base is not adequately rigid. During interaction, the sliding between the cardboard and the force sensor may change the orientation of the force sensor. This may cause error in the interaction force estimation in the z direction.

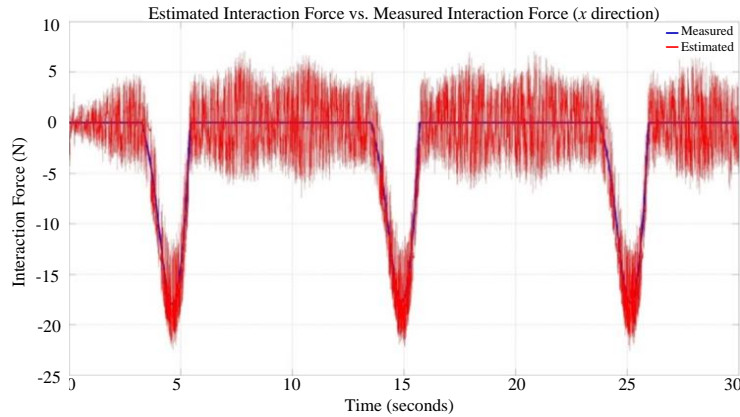


Figure 3.20: Estimated force vs. measured force (x direction with SOSMO, $T_i=0.01$)

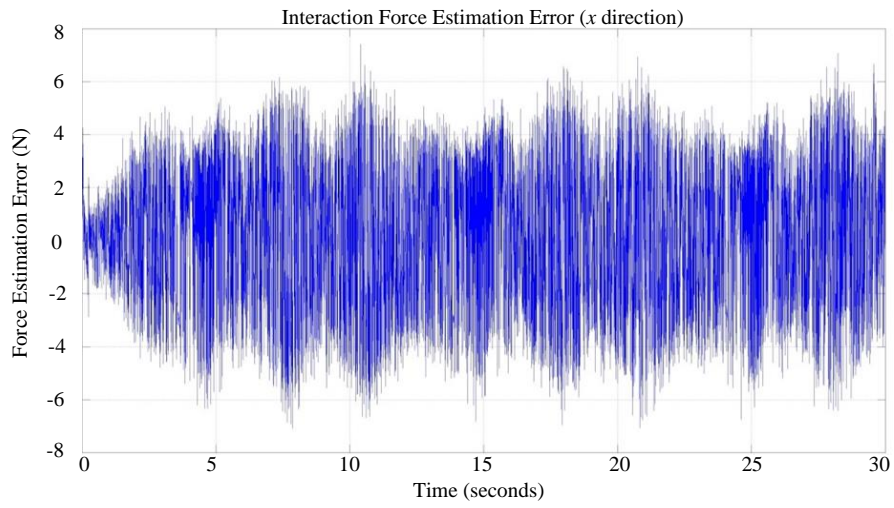


Figure 3.21: Force estimation error (x direction with SOSMO, $T_i=0.01$)

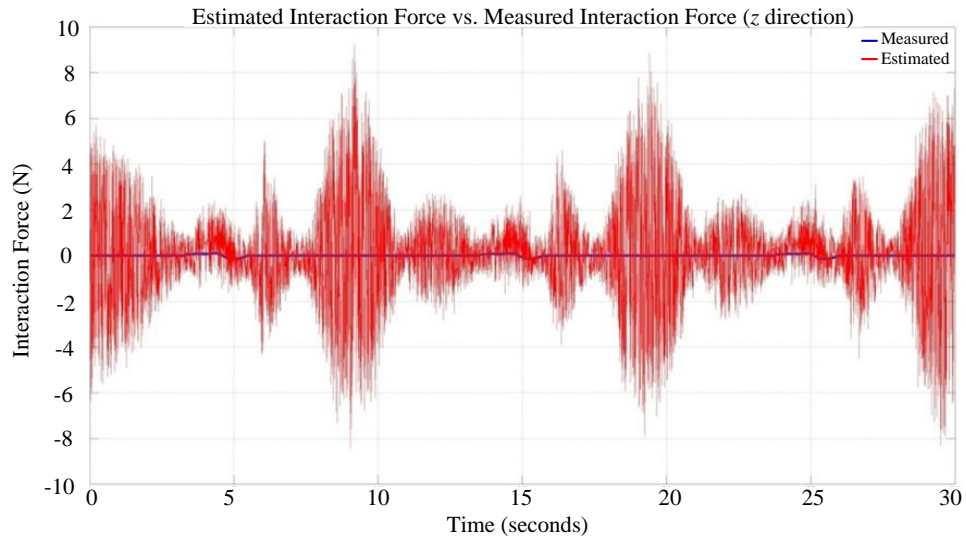


Figure 3.22: Estimated force vs. measured force (z direction with SOSMO, $T_i=0.01$)

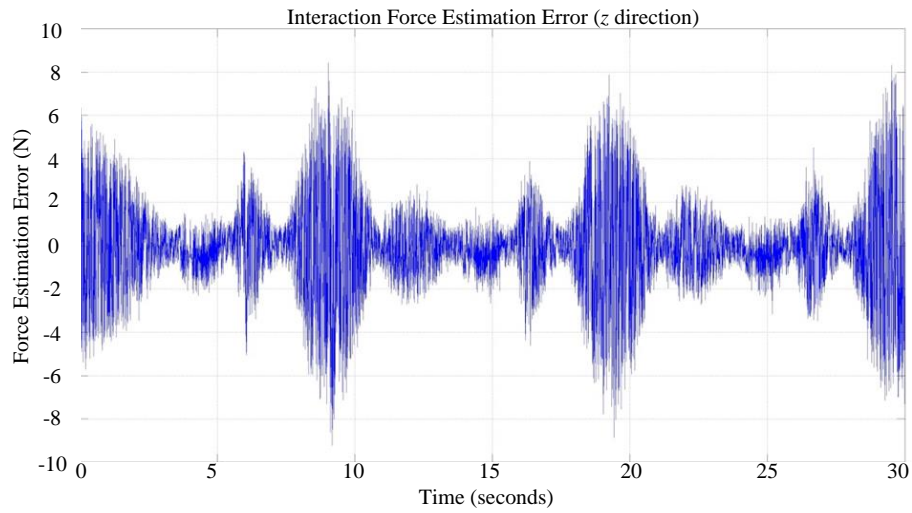


Figure 3.23: Force estimation error (z direction with SOSMO, $T_i=0.01$)

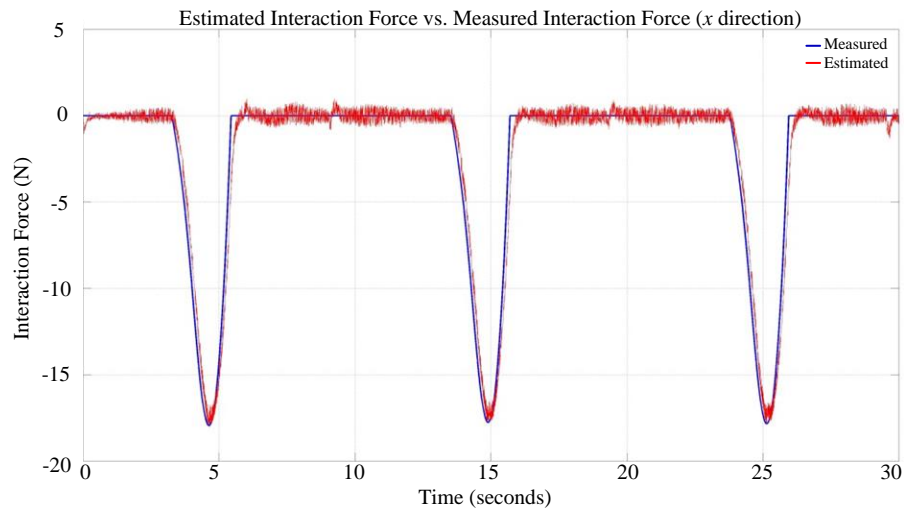


Figure 3.24: Estimated force vs. measured force (x direction with SOSMO, $T_i=0.1$)

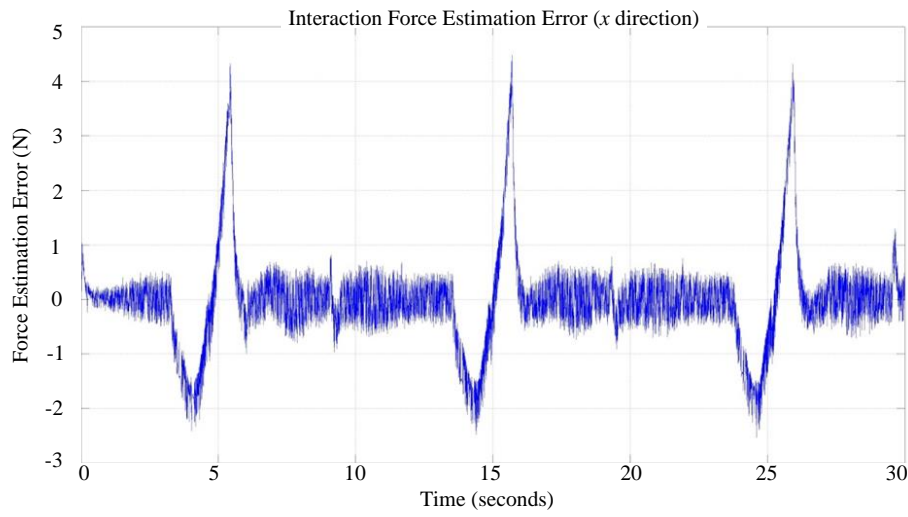


Figure 3.25: Force estimation error (x direction with SOSMO, $T_i=0.1$)

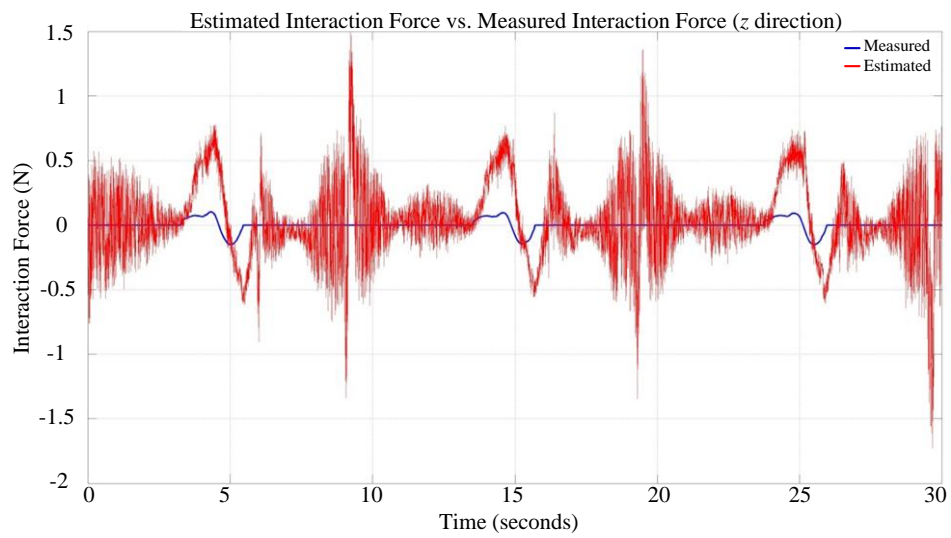


Figure 3.26: Estimated force vs. measured force (z direction with SOSMO, $T_i=0.1$)

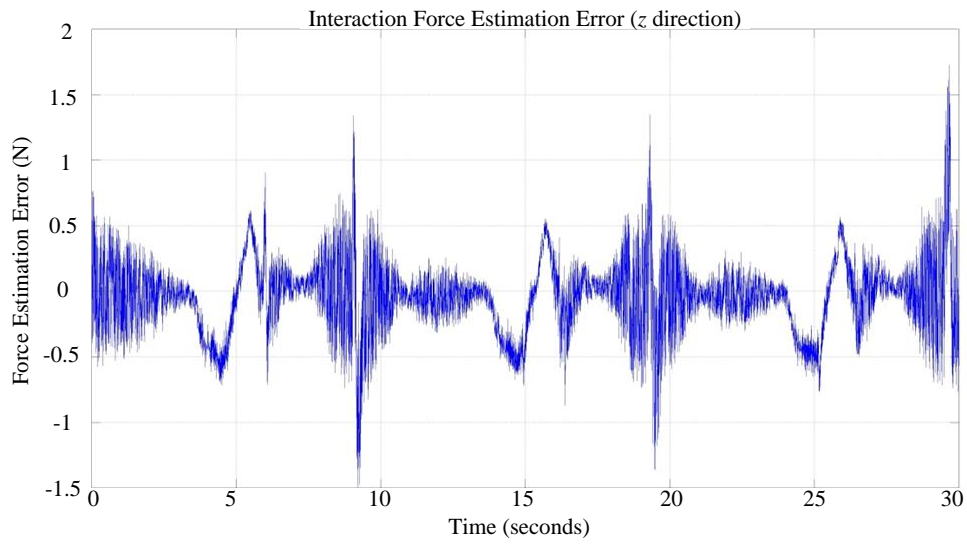


Figure 3.27: Force estimation error (z direction with SOSMO, $T_i=0.1$)

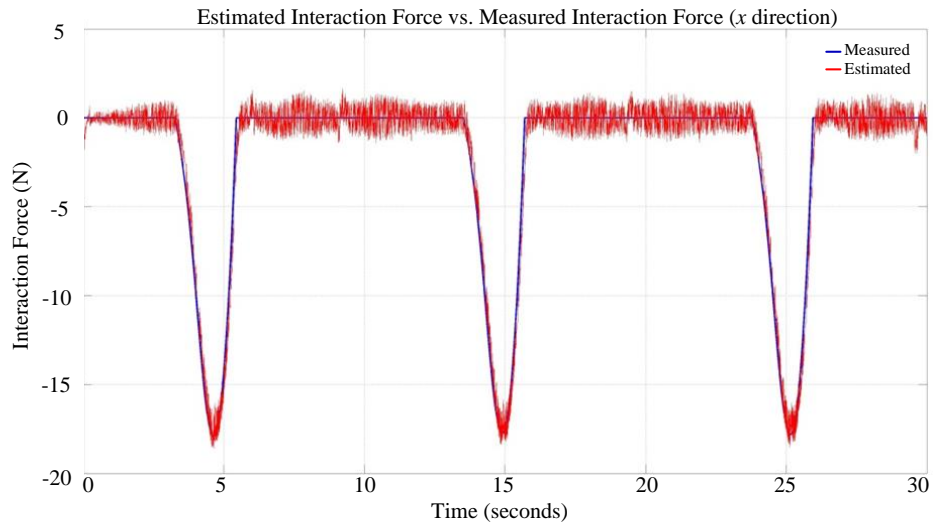


Figure 3.28: Estimated force vs. measured force (x direction with SOSMO, $T_i=0.05$)

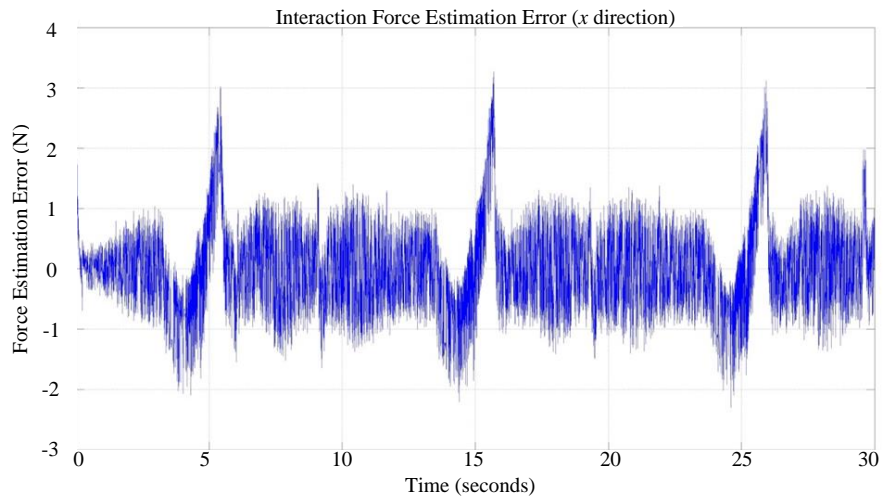


Figure 3.29: Force estimation error (x direction with SOSMO, $T_i=0.05$)

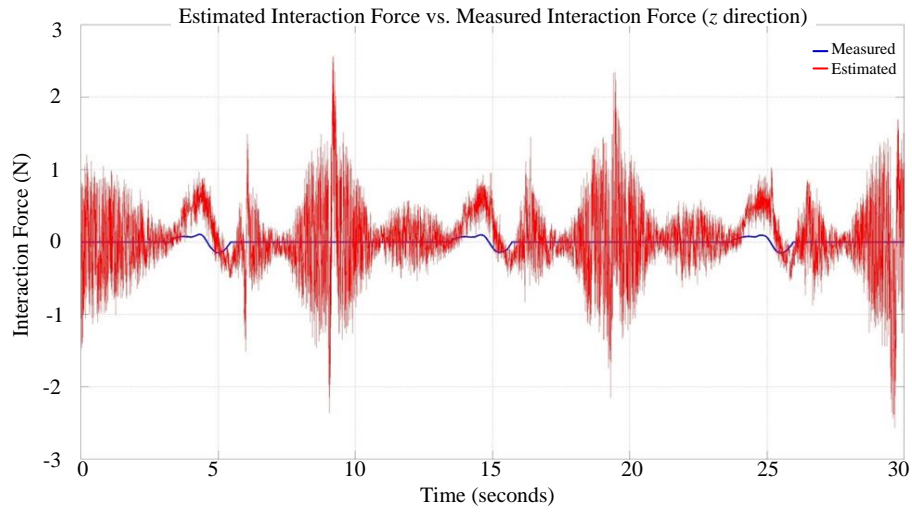


Figure 3.30: Estimated force vs. measured force (z direction with SOSMO, $T_i=0.05$)

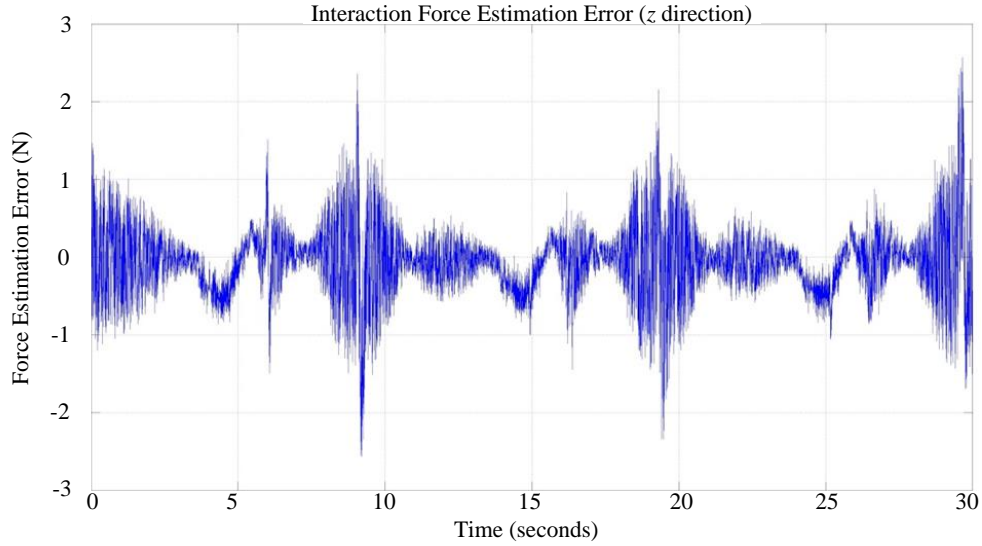


Figure 3.31: Force estimation error (z direction with SOSMO, $T_i=0.05$)

3.1.4 Adaptive High-order Sliding Mode Observer (AHOSMO) based Interaction Force Estimation

In the last subsection we noticed that there has to be some trade-off between the smoothness and the time lag of the reconstructed interaction force signal. This is due to the fact that we used a low-pass filter for the interaction force estimation based on the equivalent output injection concept. This trade-off may be eliminated if we use a high-order sliding mode observer:

$$\dot{\hat{x}}_{1i} = \hat{x}_{2i} + \lambda_{2i} |x_{1i} - \hat{x}_{1i}|^{2/3} \text{sign}(x_{1i} - \hat{x}_{1i}) \quad (3.33)$$

$$\dot{\hat{x}}_{2i} = f_i(x_1, \hat{x}_2, u) + \lambda_{1i} \left| \dot{\hat{x}}_{1i} - \hat{x}_{2i} \right|^{1/2} \text{sign}(\dot{\hat{x}}_{1i} - \hat{x}_{2i}) + \hat{z}_i \quad (3.34)$$

$$\dot{\hat{z}}_i = \lambda_{0i} \text{sign}(\dot{\hat{x}}_{1i} - \hat{x}_{2i}) \quad (3.35)$$

where λ_{ij} are the corresponding sliding mode observer gains, to be selected properly, and z_i is an auxiliary variable used for the reconstruction of the external interaction force. The state estimation error dynamics can be derived based on equation (3.33) – equation (3.35) by defining the state estimation error as $\tilde{x}_1 = x_1 - \hat{x}_1$, $\tilde{x}_2 = x_2 - \hat{x}_2$:

$$\dot{\tilde{x}}_{1i} = \tilde{x}_{2i} - \lambda_{2i} \left| \hat{x}_{1i} - \hat{x}_{1i} \right|^{2/3} \text{sign}(\tilde{x}_{1i} - \hat{x}_{1i}) \quad (3.36)$$

$$\dot{\tilde{x}}_{2i} = F_i(x_1, x_2, \hat{x}_2, u) - \lambda_{1i} \left| \dot{\hat{x}}_{1i} - \hat{x}_{2i} \right|^{1/2} \text{sign}(\dot{\hat{x}}_{1i} - \hat{x}_{2i}) - \hat{z}_i \quad (3.37)$$

$$F_i(x_1, x_2, \hat{x}_2, u) = [f(t, x_1, x_2, u) - f(t, x_1, \hat{x}_2, u) + \xi(t, x_1, x_2, u)]_i \quad (3.38)$$

$$\dot{\hat{z}}_i = \lambda_{0i} \text{sign}(\dot{\hat{x}}_{1i} - \hat{x}_{2i}) \quad (3.39)$$

When $\hat{x}_{2i} \rightarrow x_{2i}$, we have

$$\xi_i(t, x_1, x_2, u) = \hat{z}_i \quad (3.40)$$

The interaction force could be estimated as

$$F_e = -J^{-T}(x_1) [M^{-1}(x_1) \cdot \hat{z} + \Delta \tau] \quad (3.41)$$

Note that the process of interaction force reconstruction does not require the filtering of discontinuous equivalent output injection item. This will make this approach more desirable than the one discussed in the previous subsection. However, it is difficult to determine the observer gains λ . Only when the parameters are properly selected, the convergence can be guaranteed. Unfortunately, the determination of those parameters depends on the bound of uncertainty. When measurement noise is present in the system, it will make the determination of the gains even more difficult. Adaptive version of this algorithm is proposed below to guarantee the convergence without the knowledge of this uncertainty bound. It is given in equation (3.42) – equation (3.44):

$$\dot{\hat{x}}_{1i} = \hat{x}_{2i} + \hat{\lambda}_{2i} \left| \hat{x}_{1i} - \hat{x}_{1i} \right|^{2/3} \text{sign}(\hat{x}_{1i} - \hat{x}_{1i}) + k_{2i} (\hat{x}_{1i} - \hat{x}_{1i}) \quad (3.42)$$

$$\dot{\hat{x}}_{2i} = f_i(x_1, \hat{x}_2, u) + \hat{\lambda}_{1i} \left| \dot{\hat{x}}_{1i} - \hat{x}_{2i} \right|^{1/2} \text{sign}(\dot{\hat{x}}_{1i} - \hat{x}_{2i}) + k_{1i} (\dot{\hat{x}}_{1i} - \hat{x}_{2i}) + \hat{z}_i \quad (3.43)$$

$$\dot{\hat{z}}_i = \hat{\lambda}_{0i} \text{sign}(\dot{\hat{x}}_{1i} - \hat{x}_{2i}) \quad (3.44)$$

where $\hat{\lambda}_{2i}$, $\hat{\lambda}_{1i}$ and $\hat{\lambda}_{0i}$ are gains to be determined to guarantee the convergence of the estimation error.

The adaption laws for the gains are developed in this subsection. It is inspired by the work of [48] and [49]. However, the adaption laws derived there are for robust differentiators. In the present work, new adaption laws are designed for the super-twisting-based unknown input observers.

For the derivation simplicity, a matrix representation is used for the observer described by the following equations.

$$\dot{\hat{x}}_1 = \hat{x}_2 + \hat{\lambda}_2 |x_1 - \hat{x}_1|^{2/3} \text{sign}(x_1 - \hat{x}_1) + k_2(x_1 - \hat{x}_1) \quad (3.45)$$

$$\dot{\hat{x}}_2 = f(x_1, \hat{x}_2, u) + \hat{\lambda}_1 |\dot{\hat{x}}_1 - \hat{x}_2|^{1/2} \text{sign}(\dot{\hat{x}}_1 - \hat{x}_2) + k_1(\dot{\hat{x}}_1 - \hat{x}_2) + \hat{z} \quad (3.46)$$

$$\dot{\hat{z}} = \hat{\lambda}_0 \text{sign}(\dot{\hat{x}}_1 - \hat{x}_2) \quad (3.47)$$

The correspondence between these two representations is obvious and has been used in other work.

The variables in equations (3.45) – equation (3.47) are the corresponding vectors of equation (3.42) – equation (3.44). Before deriving the adaption law, two new variables are defined:

$$s_2 = x_1 - \hat{x}_1 \quad (3.48)$$

$$s_1 = \dot{\hat{x}}_1 - \hat{x}_2 \quad (3.49)$$

There should be a parameter λ_2^* such that

$$\dot{q} = \dot{x}_1 = \hat{x}_2 + \lambda_2^* |x_1 - \hat{x}_1|^{2/3} \text{sign}(x_1 - \hat{x}_1) \quad (3.50)$$

Comparing equation (3.45) with equation (3.50), the derivative of s_2 is

$$\dot{s}_2 = \dot{x}_1 - \dot{\hat{x}}_1 = -(\hat{\lambda}_2 - \lambda_2^*) |s_2|^{2/3} \text{sign}(s_2) - k_2 s_2 \quad (3.51)$$

Let $\tilde{\lambda}_2 = \hat{\lambda}_2 - \lambda_2^*$ and define the Lyapunov function V_2 :

$$V_2 = \frac{1}{2}(s_2^2 + \tilde{\lambda}_2^2) \quad (3.52)$$

The time derivative of V_2 along the system trajectory is

$$\begin{aligned}
\dot{V}_2 &= s_2 \dot{s}_2 + \tilde{\lambda}_2 \dot{\lambda}_2 \\
&= s_2 \left[-\tilde{\lambda}_2 |s_2|^{2/3} \text{sign}(s_2) - k_2 s_2 \right] + \tilde{\lambda}_2 \dot{\lambda}_2 \\
&= -k_2 s_2^2 + \tilde{\lambda}_2 \left[\dot{\lambda}_2 - s_2 |s_2|^{2/3} \text{sign}(s_2) \right]
\end{aligned} \tag{3.53}$$

Selecting the adaption law for $\hat{\lambda}_2$ as

$$\dot{\hat{\lambda}}_2 = s_2 |s_2|^{2/3} \text{sign}(s_2) \tag{3.54}$$

we have

$$\dot{V}_2 = -k_2 s_2^2 \tag{3.55}$$

In this manner the convergence of s_2 is concluded. Similarly, the derivative of s_1 may be represented as

$$\dot{s}_1 = \ddot{x}_1 - \dot{\hat{x}}_2 \tag{3.56}$$

There should be a parameter λ_1^* such that

$$\ddot{q} = \ddot{x}_1 = f(x_1, \hat{x}_2, u) + \lambda_1^* |s_1|^{1/2} \text{sign}(s_1) + \lambda_0^* \int_0^t \text{sign}(s_1) dt \tag{3.57}$$

By comparing equation (3.46) with equation (3.57), we have

$$\dot{s}_1 = \ddot{x}_1 - \dot{\hat{x}}_2 = (\lambda_1^* - \hat{\lambda}_1) |s_1|^{1/2} \text{sign}(s_1) + (\lambda_0^* - \hat{\lambda}_0) \int_0^t \text{sign}(s_1) dt - k_1 s_1 \tag{3.58}$$

Defining $\tilde{\lambda}_1 = \hat{\lambda}_1 - \lambda_1^*$ and $\tilde{\lambda}_0 = \hat{\lambda}_0 - \lambda_0^*$, we have

$$\dot{s}_1 = -\tilde{\lambda}_1 |s_1|^{1/2} \text{sign}(s_1) - \tilde{\lambda}_0 \int_0^t \text{sign}(s_1) dt - k_1 s_1 \tag{3.59}$$

A new Lyapunov function V_1 is chosen as

$$V_1 = \frac{1}{2} (s_1^2 + \tilde{\lambda}_1^2 + \tilde{\lambda}_0^2) \tag{3.60}$$

The derivative of V_1 along the system trajectory is given by

$$\begin{aligned}
\dot{V}_1 &= s_1 \dot{s}_1 + \tilde{\lambda}_1 \dot{\lambda}_1 + \tilde{\lambda}_0 \dot{\lambda}_0 \\
&= s_1 \left[-\tilde{\lambda}_1 |s_1|^{1/2} \text{sign}(s_1) - \tilde{\lambda}_0 \int_0^t \text{sign}(s_1) dt - k_1 s_1 \right] + \tilde{\lambda}_1 \dot{\lambda}_1 + \tilde{\lambda}_0 \dot{\lambda}_0 \\
&= -k_1 s_1^2 + \tilde{\lambda}_1 \left[\dot{\lambda}_1 - s_1 |s_1|^{1/2} \text{sign}(s_1) \right] + \tilde{\lambda}_0 \left[\dot{\lambda}_0 - s_1 \int_0^t \text{sign}(s_1) dt \right]
\end{aligned} \tag{3.61}$$

Selecting the adaption law for $\hat{\lambda}_1$ and $\hat{\lambda}_0$ as

$$\dot{\hat{\lambda}}_1 = s_1 |s_1|^{1/2} \text{sign}(s_1) \quad (3.62)$$

$$\dot{\hat{\lambda}}_0 = s_1 \int_0^t \text{sign}(s_1) dt \quad (3.63)$$

we have

$$V_1 = -k_1 s_1^2 \quad (3.64)$$

Thus the convergence of s_1 is concluded.

The results of interaction force estimation of this AHOSMO-based algorithm are given in Figure 3.32 – Figure 3.35. It is seen that the performance is improved when compared with the SOSMO interaction force estimation algorithm. With regard to implementation, this approach is more applicable since it is able to reconstruct the interaction force without filtering. Also, it does not need the knowledge of the uncertainty bound.

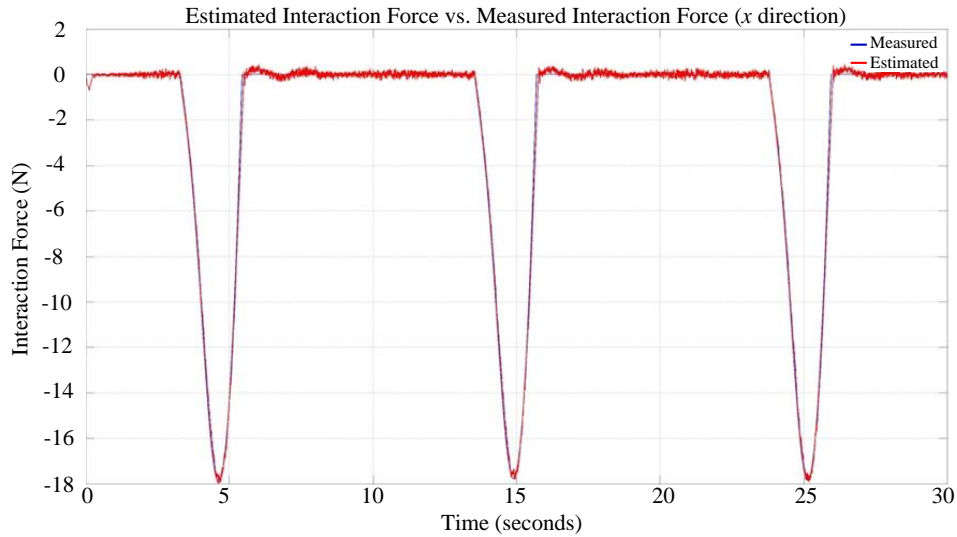


Figure 3.32: Estimated force vs. measured force (x direction with AHOSMO)

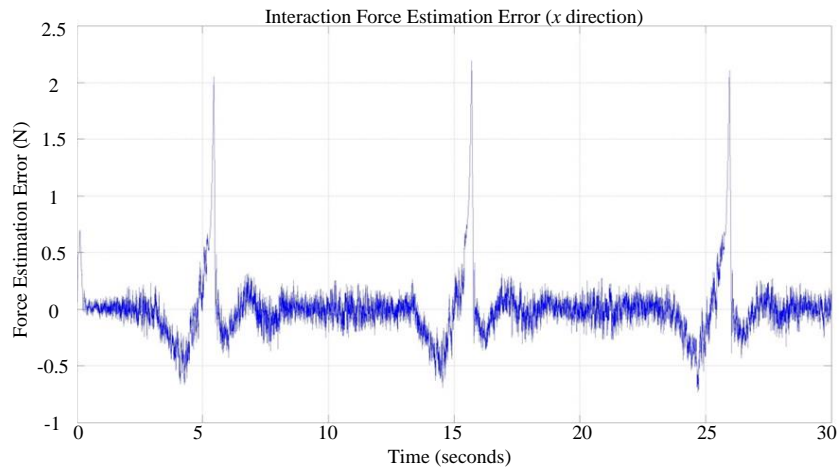


Figure 3.33: Force estimation error (x direction with AHOSMO)

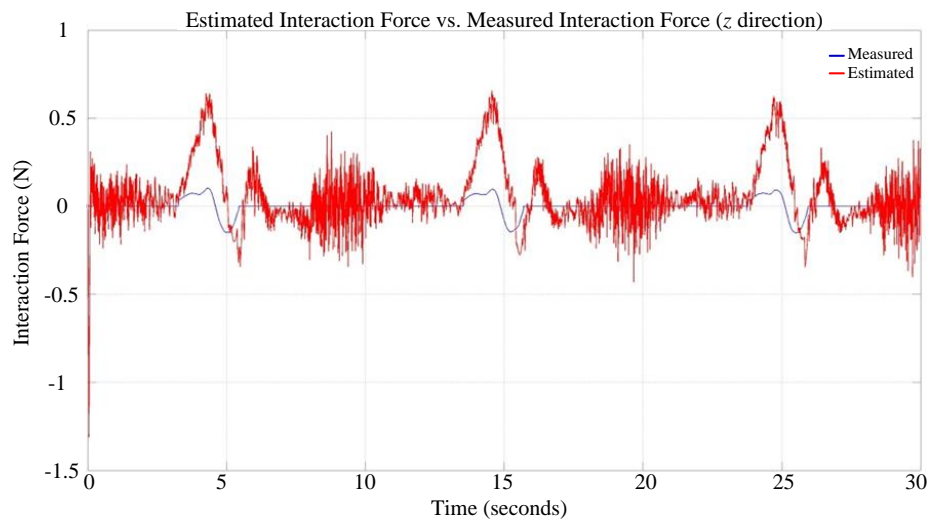


Figure 3.34: Estimated force vs. measured force (z direction with AHOSMO)

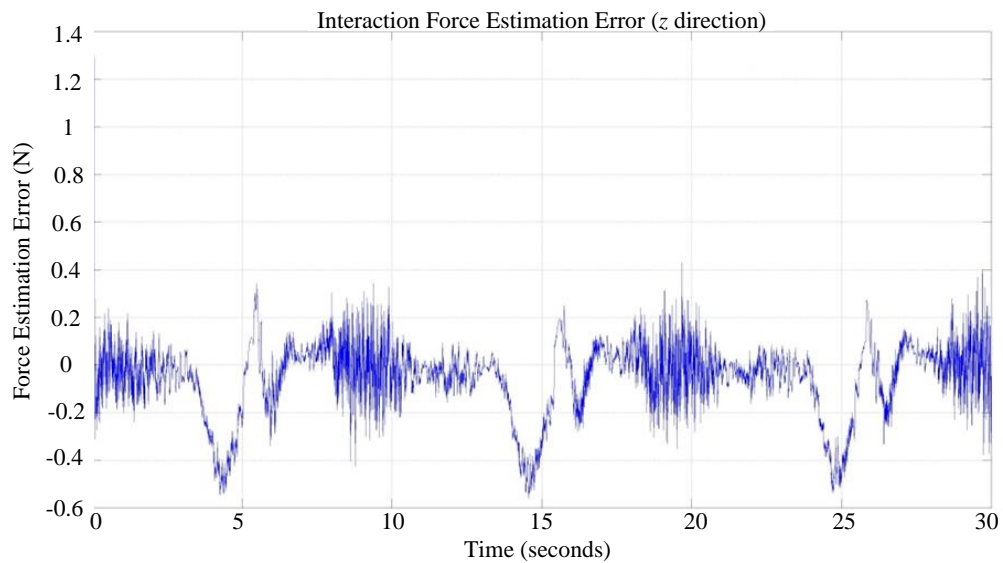


Figure 3.35: Force estimation error (z direction with AHOSMO)

3.2 Summary

In this chapter, the observer based interaction force reconstruction algorithms are proposed and compared. Due to the high cost and the unavoidable limitations with force sensors, the interaction force which is required in the impedance control algorithms is obtained using observers. Sliding mode-based interaction force observers show their effectiveness in interaction force reconstruction. The AHOSMO-based interaction force observer designed in this chapter is used in the following chapters to form an observer-controller framework for impedance control.

Chapter 4: Impedance Control without Direct Force Sensing

Mechanical impedance is defined as the interaction force over the corresponding velocity in a mechanical system. It can arise due to inertia, flexibility (stiffness) and damping in a mechanical system, and is analogous to the concept of electrical impedance (strictly, its inverse—admittance) resulting from the dynamic effects of resistors, inductors, and capacitors. To ensure safe interaction between a human body and a robot manipulator, during robot-assistive tasks, impedance control is developed and implemented for interaction control in the present project of homecare robotics. Impedance control is an indirect force control approach which seeks to control the impedance properties instead of the actual position or force in the manipulator-object interface during interaction. Since its introduction in the 1980s, it has been promoted for applications of dynamic interaction control because of its desirable properties in task execution and robustness.

To control the mechanical impedance of a task arbitrarily, the interaction force between the manipulator and the environment should be determined. The interaction force determined through observers (Chapter 3) is used here to form a controller-observer scheme to realize impedance control. This kind of impedance control scheme is called sensor-less impedance control since it does not need direct sensing of force. In order to show the advantage of impedance control over conventional position-based interaction control, a comparison between them is carried out as well in this chapter.

An accuracy measure for impedance control is defined. Different impedance controllers are proposed, and the corresponding impedance control algorithms are compared with regard to their accuracy of impedance shaping. This chapter serves as the basis for the next chapter where linear-time-invariant teleoperation system is studied. The impedance control algorithms that are developed in this chapter will be used in the teleoperation control algorithms discussed in the next chapter. The finite-time convergence properties of the proposed impedance control algorithms will significantly simplify the analysis of the teleoperation system.

4.1 Introduction

4.1.1 Impedance Control in Interaction Applications

Mechanical impedance is defined as the transfer function, exerted force over the resulting velocity, as given by

$$Z(s) = \frac{F(s)}{V(s)} \quad (4.1)$$

where F is the applied force and V is the resulting velocity. In the case of a block of mass moving against frictional resistance, as shown in Figure 4.1, the mechanical impedance is

$$Z(s) = \frac{F(s)}{V(s)} = Ms + B \quad (4.2)$$

where M and B are the mass of the mass block and viscous friction coefficient between the mass and the ground, respectively.

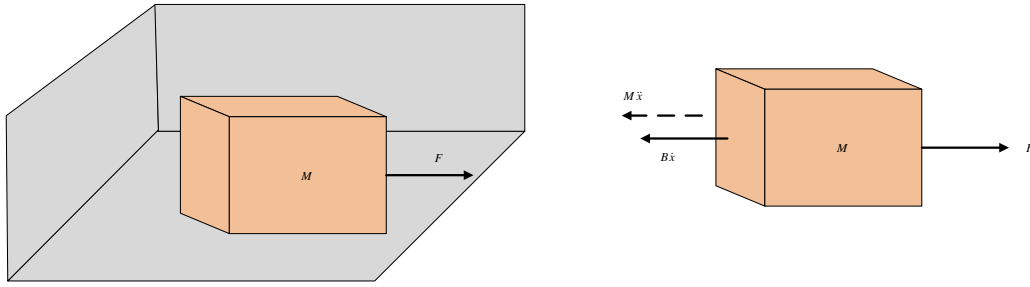


Figure 4.1: Original mechanical impedance of a mass block

This impedance is called the original impedance of this mechanical system. It depends on the natural properties of the mass, and the smoothness of the surface on which it moves. The use of the original impedance might be inappropriate in some situations. For example, suppose that we use the mass block to interact with some environment, and the mass is too large. As a result, an excessive interaction force is expected during interaction. This justifies the need to tune the mechanical impedance to meet the task requirements.

Mechanical impedance can be actively tuned by feedback control. The same example as before is taken now to show the process of actively tuning the mechanical impedance of a block of mass. Now, there is a propeller installed on the original block, as shown in Figure 4.2. It is able to exert a desired force to the block. The exerted force by the propeller is calculated by the feedback control law given in equation (4.3). It is a function

of the current states of the mass block. When the desired impedance characteristics at the mass block are known, the feedback control law will calculate the corresponding external force. The calculated external force will be delivered by the propeller. In this way, the mechanical impedance can be tuned actively to meet the specific requirements of the application, as given in equation (4.4).

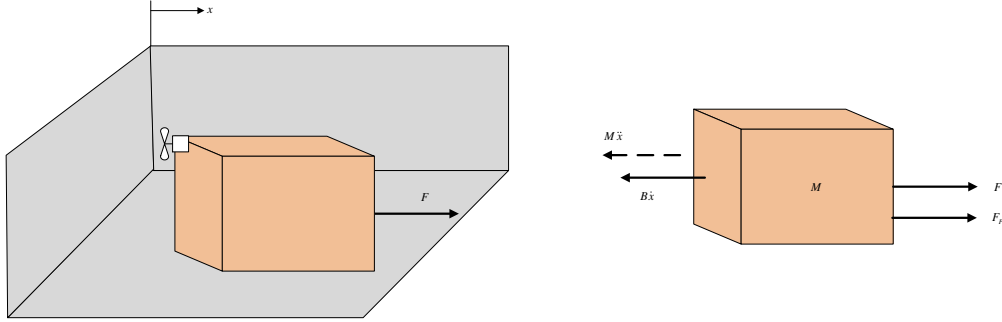


Figure 4.2: Actively controlled mechanical impedance

$$F_p(s) = \Delta M \cdot \ddot{x} + \Delta B \cdot \dot{x} + \Delta K \cdot x \quad (4.3)$$

$$Z(s) = \frac{F(s)}{V(s)} = (M + \Delta M)s + (B + \Delta B) + \frac{K}{s} \quad (4.4)$$

Generally, in robot interaction control, the objective is to control the interaction force and resulting motion to satisfy

$$\frac{F_z(s)}{V_z(s)} = M_d s + B_d + \frac{K_d}{s} \quad (4.5)$$



Figure 4.3: Impedance control in robot interaction applications

The robot dynamics in the Cartesian space is given in equation (2.44), and is repeated here.

$$M_c(X)\ddot{X} + C_c(X, \dot{X})\dot{X} + F_{vc}(X)\dot{X} + F_{cc}(X) + G_c(X) = F_\tau(X) - F_e \quad (2.44)$$

When the robot manipulator is commanded to follow a desired Cartesian space trajectory, the dynamic equation describing the manipulator under impedance control could be described by equation (1.31), as repeated below

$$M_d(\ddot{X} - \ddot{X}_d) + B_d(\dot{X} - \dot{X}_d) + K_d(X - X_d) = F_e \quad (1.31)$$

where M_d , B_d , and K_d are the desired inertia, damping and stiffness matrix, respectively, of the impedance model.

If an accurate dynamic model is available, by the conventional inverse dynamics-based control formulation, the actuator torque is given by

$$\begin{aligned} \tau = G(q) + J^T(q) \Big[M_c(X) \ddot{X}_d + C_c(X, \dot{X}) \dot{X} + F_{vc}(X) \dot{X} \Big] \\ - J^T(q) M_c(X) M_d^{-1}(X) (K_d \tilde{X} + D_d \dot{\tilde{X}}) + J^T(q) \Big[M_c(X) M_d^{-1}(X) - I \Big] F_e \end{aligned} \quad (4.6)$$

It is clear that, if we want to tune the impedance parameters of the manipulator arbitrarily, the interaction force F_e should be available. The intuitive way is to use a force sensor to sense the interaction force. However, there are some inherent limitations when using a force sensor as discussed in Chapter 3. Observer-based impedance control algorithms may be applied to overcome these limitations and achieve impedance control in the absence of force sensors.

4.1.2 Incomplete Impedance Control without Force Sensors

One way to avoid the force sensing requirement in impedance control is to select

$$M_d(x) = M_c(x) \quad (4.7)$$

In this case, the corresponding coefficient relating to interaction force in equation (4.6) is zero, which means the interaction force information is not needed. The limitation of this approach is that the inertia matrix in the desired impedance model in equation (1.31) cannot be selected arbitrarily. This is why it is called incomplete impedance control.

The desired inertia of the manipulator should always be selected to be equal to the original impedance of the manipulator. In case that the original impedance of the manipulator is not appropriate for the application, this approach cannot be applied.

The desired impedance matrix is preferred to be diagonal. Another limitation with this approach is that the desired inertia component is not a diagonal matrix. The desired inertia component in this case is configuration dependent and time varying. Because the

matrix is non-diagonal, the impedances between various DOFs are coupled, which complicates the controller design and stability analysis.

Finally, an accurate dynamic model is not available in practice. By selecting $M_d(x) = M_c(x)$, we actually cannot make the coefficient of interaction force in equation (4.6) to be zero. This will further complicate the subsequent controller design and system stability analysis.

When direct force sensing is unavailable, the resulting limitations of incomplete impedance control can be eliminated by using observers to estimate the interaction force. In that approach, the interaction force is observed using a dynamic model, the input, and the output. Conventional observers are used to reconstruct the system state, or the unknown input. Most observers converge to the actual value asymptotically. It is desirable if the system states or unknown inputs that are reconstructed by an observer will converge in finite time. AHOSMO, which is proposed in chapter 3, converges in finite time. It is used in this chapter for impedance control.

4.2 Impedance Control Using AHOSMO-estimated Interaction Force

4.2.1 Problem Formulation

As we can see from equation (3.6), the Cartesian space velocity and interaction force are required for impedance control in order to arbitrarily shape the mechanical impedance of the manipulator. They should be reconstructed individually or simultaneously by observers. Also, it is desirable that the estimated variables converge to the actual ones in finite time. Then the separation principle, which is only applicable in linear systems, is also valid.

The Cartesian space velocity can be obtained by the differential kinematics, as a product of the Jacobian and the joint velocity. The joint space velocity can be obtained by differentiating the joint space position with respect to time directly or by using the robust differentiator proposed in Chapter 2. In Chapter 2 we find that robust differentiator-based velocity calculation outperforms direct differentiation. The proposed robust differentiator can also be applied to a Cartesian space dynamic model. In this way, the Cartesian space velocity can be obtained directly.

In Chapter 3 we proposed different interaction force estimation algorithms. The force estimation accuracy of them was compared. AHOSMO showed best interaction force

estimation results. Fortunately, AHOSMO proposed in chapter 3 is able to estimate not only the interaction force but also the velocity. The velocity information is required in the impedance controllers proposed in this chapter.

Now we have two approaches to estimate the Cartesian space velocity. The first one is to use the robust differentiator introduced in Chapter 2. It does not need a dynamic model of the manipulator. The second approach is to use AHOSMO. Since AHOSMO is able to reconstruct velocity and interaction force simultaneously, it will be used in this chapter for impedance control.

4.2.2 Cartesian Space Reference Trajectory Selection

To show the effectiveness of impedance control as an interaction control algorithm, a common Cartesian space trajectory is planned for the end-effector to follow. The end-effector will interact with the obstacle by following this trajectory. The derivatives of this reference trajectory are directly used in the control algorithms. Hence, it is desirable to have smooth reference trajectories for both velocity and acceleration.

In the reference trajectory selection, the specific application should also be considered. In our homecare project, the robot end-effector is commanded to follow the reference trajectory to come in contact with the human body. In addition, the initial Cartesian space position of the robot end-effector is $(-0.4691, 0.1193)$. The interacted object is located at $(0.08, 0.6)$ in the Cartesian space. After that, the end-effector is made to exert some periodic motion on the object.

Taking all the factors discussed above into consideration, the reference trajectories are divided into two phases. The switching between the two phases happens at $t = 2.0\text{sec}$. In the first phase, the end-effector in the Cartesian space is commanded to move to some adjacent region of the designated position. When the end-effector is in the designated location, it is commanded to execute a desired periodical motion. Since there are six known conditions (initial position, initial velocity, initial acceleration, final position, final velocity, and final acceleration) in phase 1, a fifth-order polynomial is used to describe the trajectory in this phase.

For the x direction, assume that

$$x(t) = a_{x5} \cdot t^5 + a_{x4} \cdot t^4 + a_{x3} \cdot t^3 + a_{x2} \cdot t^2 + a_{x1} \cdot t + a_{x0} \quad (4.8)$$

with $x(0) = -0.4691$, $\dot{x}(0) = 0$, $\ddot{x}(0) = 0$, $x(2) = 0.1$, $\dot{x}(2) = 0$, $\ddot{x}(2) = 0$, which may be used to determine the polynomial coefficients. Then,

$$x(t) = 0.1067 \cdot t^5 - 0.5335 \cdot t^4 + 0.7114 \cdot t^3 - 0.4691, \quad t \in [0, 2] \quad (4.9)$$

Similarly, the reference trajectory for the z direction is determined as

$$z(t) = 0.0651 \cdot t^5 - 0.3381 \cdot t^4 + 0.4759 \cdot t^3 + 0.1193, \quad t \in [0, 2] \quad (4.10)$$

After determining the phase 1 polynomial, the velocity and acceleration at $t = 2\text{sec}$ are known. There will be three known conditions for phase 2. A sinusoidal waveform is used to describe the periodic motion in both directions. However, there should be four known conditions in order to completely describe a sinusoidal signal. We can assign any of these four to a specified value in order to have a closed form solution. In the z direction, it is reasonable to set the amplitude of the sinusoidal wave to be 0.1. For the x direction, it is assumed to be the constant location 0.1.

Then the sinusoidal waveforms for x and z directions, when $t \in [2.0, 10]$, may be determined as

$$x(t) = 0.1 \quad (4.11)$$

$$z(t) = 0.1 \sin(1.272t - 1.8780) + 0.5382 \quad (4.12)$$

The designed Cartesian space trajectories are shown in Figure 4.4. The corresponding joint space position trajectories are shown in Figure 4.5. We can see from the joint space reference trajectories that the position of joint 2 will never be 0 or π , which means the manipulator will not be in the singularity configurations if the trajectory tracking performance is good.

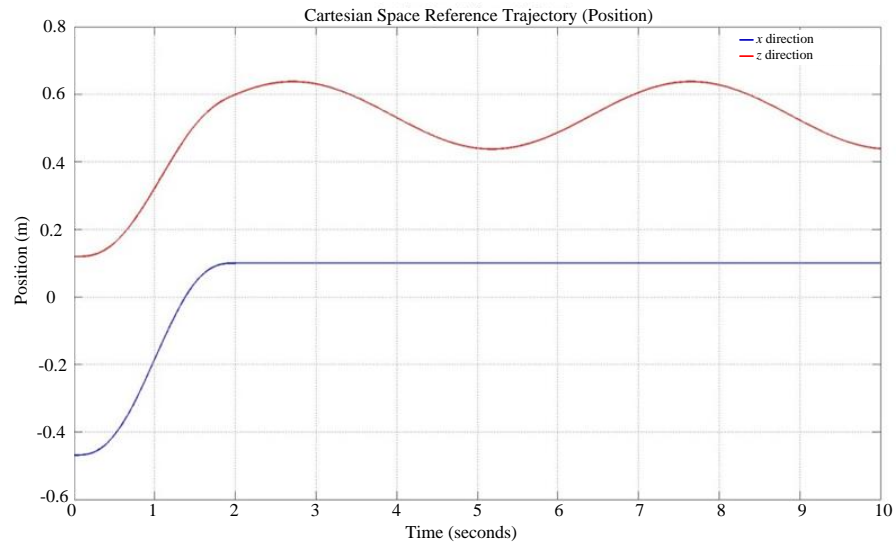


Figure 4.4: Cartesian space reference trajectories (for interaction control)

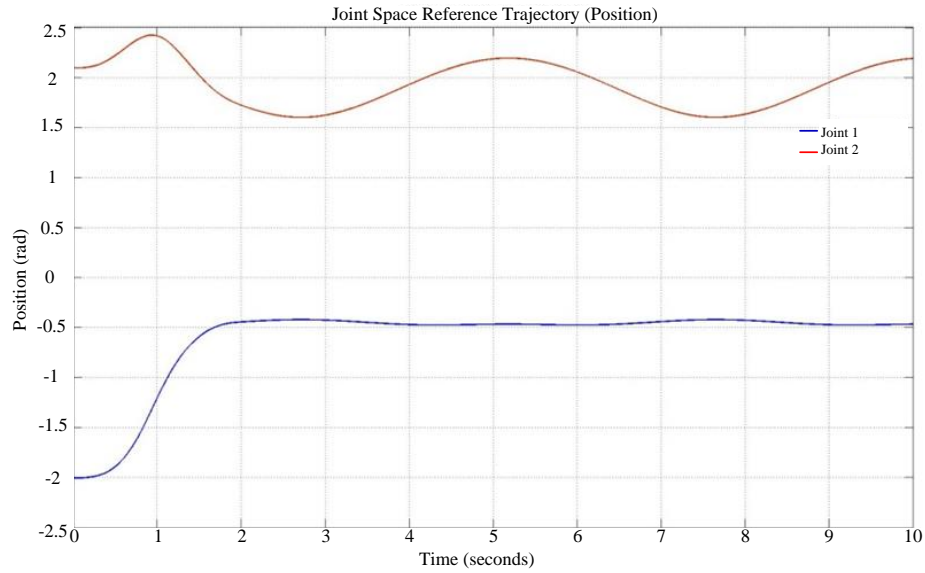


Figure 4.5: Joint space reference trajectories (for interaction control)

4.2.3 Interaction Control through Position Control

Interaction force can be controlled through exclusive position control approaches. Then, interaction force is taken as disturbances. The position control system will calculate the desired actuator torque so that the reference trajectory will be tracked. It may generate an excessive interaction force during interaction, which has to be avoided.

If the interaction force has to be modulated, the mechanical properties of the environment have to be accurately known. However, the mechanical properties of the

environment are difficult to obtain. It is shown that an excessive interaction force may be present when pure position control strategies are used in interaction control. This will act as the control group for the following subsections, to show the advantage of impedance control when it is used as the interaction control strategy in the present project.

The Cartesian space inverse dynamics-based trajectory tracking controller is used here. This control algorithm is given in equation (4.13). After simplification, the trajectory tracking problem can be described by equation (4.14).

$$\tau = M(q)J^{-1}(q)\left[-\dot{J}(q)\dot{q} + \ddot{X}_d + K_D(\dot{X}_d - \dot{X}) + K_P(X_d - X)\right] + C(q, \dot{q})\dot{q} + F_c \cdot \text{sign}(\dot{q}) + F_v \cdot \dot{q} + G(q) + J^T(q)\hat{F}_e \quad (4.13)$$

$$\ddot{X} - \ddot{X}_d + K_D(\dot{X} - \dot{X}_d) + K_P(X - X_d) = 0 \quad (4.14)$$

Here K_p and K_D are the corresponding proportional and derivative gains.

Note that in this case, the interaction control problem is transformed into a pure trajectory tracking scheme, which means the controller renders the manipulator infinitely stiff. In this case, the external interaction force is considered as the external disturbance to be rejected by the Cartesian space trajectory tracking algorithm.

The selected values of K_p and K_D are given in equation (4.15). The choice of these two parameters is based on the selection of the desired impedance parameters in the following subsection. This will lay the foundation for performance comparison between pure position-based interaction control and explicit impedance control.

$$K_D = \begin{bmatrix} 0.2 & \\ & 0.4 \end{bmatrix}, \quad K_P = \begin{bmatrix} 25 & \\ & 50 \end{bmatrix} \quad (4.15)$$

The interaction forces in the two directions are shown in Figure 4.6 and Figure 4.7. We can see that this control scheme can lead to a large interaction force which is unacceptable in the present application. This is due to the large stiffness of this control scheme. Then, a small position tracking error will make the manipulator to command an extra control torque to drive the end-effector to the desired position regardless of the stiffness of the objects that the manipulator is interacting with.

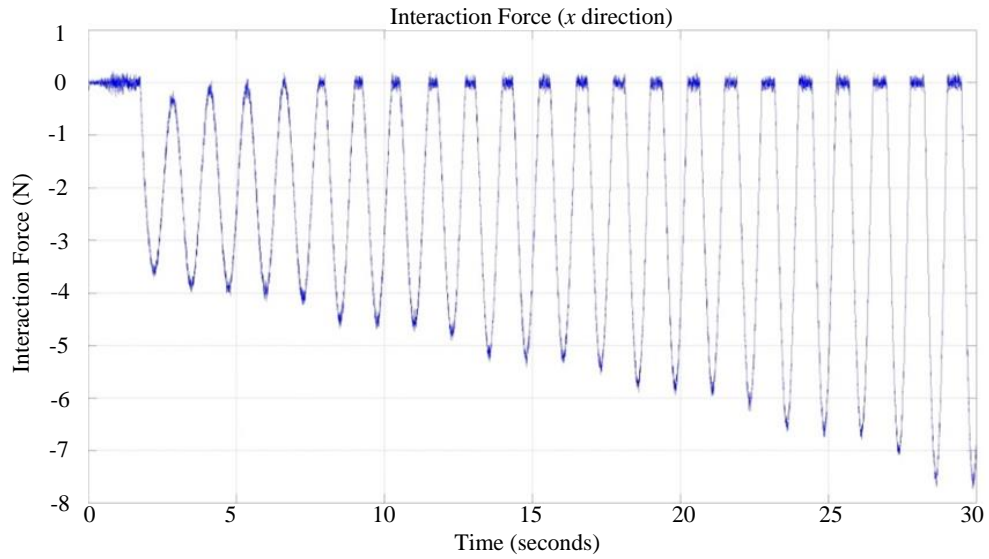


Figure 4.6: Interaction force under pure position control (x direction)

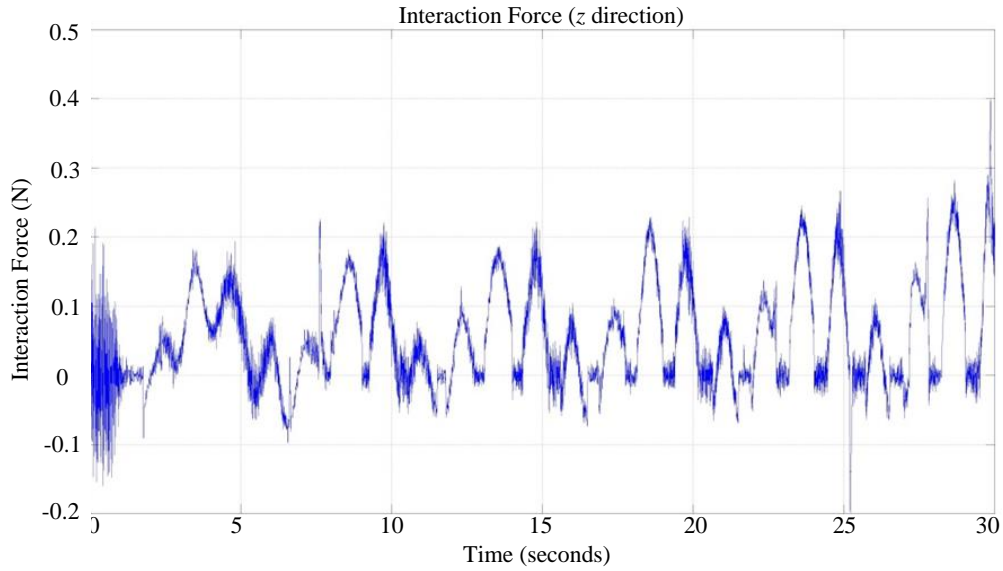


Figure 4.7: Interaction force under pure position control (z direction)

We can reduce this problem by using the control law given in equation (4.16). Here, the interaction force is not included in the control law, which will make the manipulator have some compliance toward the interaction force.

$$\begin{aligned} \tau = M(q)J^{-1}(q) \left[-\dot{J}(q)\dot{q} + \ddot{X}_d + K_D(\dot{X}_d - \dot{X}) + K_P(X_d - X) \right] \\ + C(q, \dot{q})\dot{q} + F_c \cdot \text{sign}(\dot{q}) + F_v \cdot \dot{q} + G(q) \end{aligned} \quad (4.16)$$

This controller will lead to the position tracking error given by

$$\ddot{X} - \ddot{X}_d + K_D(\dot{X} - \dot{X}_d) + K_P(X - X_d) = -J(q)M^{-1}(q)J^T(q)F_e \quad (4.17)$$

It is seen that unless the right-hand side of equation (4.17) is zero, the trajectory tracking error will not go to zero even though positive definite matrices K_p and K_d are selected.

Strictly, this is a type of impedance control. We can rewrite equation (4.17) as equation (4.18). The inertia component, damping component, and the stiffness component corresponding to this type of impedance controllers are given by

$$J^{-T}(q)M(q)J^{-1}(q)(\ddot{X} - \ddot{X}_d) + J^{-T}(q)M(q)J^{-1}(q)K_d(\dot{X} - \dot{X}_d) + J^{-T}(q)M(q)J^{-1}(q)K_p(X - X_d) = -F_e \quad (4.18)$$

$$M_d(q) = J^{-T}(q)M(q)J^{-1}(q) \quad (4.19)$$

$$B_d(q) = J^{-T}(q)M(q)J^{-1}(q)K_d \quad (4.20)$$

$$K_d(q) = J^{-T}(q)M(q)J^{-1}(q)K_p \quad (4.21)$$

Here $M_d(q)$, $B_d(q)$, and $K_d(q)$ are configuration-dependent since they are functions of the manipulator's present position. These matrices are not always diagonal. Impedance parameters in each DOF cannot be assigned independently. This is not desirable since each DOF has a different requirement regarding the impedance parameters to be assigned. This justifies the need for explicit impedance control schemes.

4.2.4 Accuracy of Impedance Control

The accuracy of impedance control corresponds to the extent to which the ideal impedance model is realized. This is discussed here particularly to compare the performance of different impedance control algorithms that are proposed in the following subsections. The desired impedance model is described by

$$M_d(\ddot{X} - \ddot{X}_d) + B_d(\dot{X} - \dot{X}_d) + K_d(X - X_d) = F_e \quad (1.31)$$

Let $\tilde{X} = X - X_d$. Then equation (1.31) can be rewritten as

$$M_d\ddot{\tilde{X}} + B_d\dot{\tilde{X}} + K_d\tilde{X} - F_e = 0 \quad (4.22)$$

Define the following matrix variable

$$I_e = M_d\ddot{\tilde{X}} + B_d\dot{\tilde{X}} + K_d\tilde{X} - F_e \quad (4.23)$$

If $I_e = \text{diag}(0, 0)$, then the desired impedance is realized.

However, since a force sensor is not used in the present project, F_e in the right-hand side of equation (4.23) will be replaced with the estimated interaction force \hat{F}_e . Thus, a modified version of equation (4.23) may be given by

$$\hat{I}_e = M_d \ddot{\tilde{X}} + B_d \dot{\tilde{X}} + K_d \tilde{X} - \hat{F}_e \quad (4.24)$$

We have noticed that the estimated interaction forces have some residue with respect to the measured ones. This is unavoidable no matter what type of system identification algorithm is used to identify the dynamic model of the manipulator. Hence, the estimated interaction force may be represented as

$$F_e = \hat{F}_e + \Delta_F \quad (4.25)$$

where Δ_F is the estimation error in the interaction force. Then equation (1.31) may be rewritten as

$$M_d(\ddot{X} - \ddot{\tilde{X}}_d) + B_d(\dot{X} - \dot{\tilde{X}}_d) + K_d(X - \tilde{X}_d) = \hat{F}_e + \Delta_F \quad (4.26)$$

ΔF may be represented as a function of the manipulator's current Cartesian space configuration, as

$$\Delta F = \Delta M_C(X) \ddot{X} + \Delta F(\dot{X}) \dot{X} + \Delta G_C(X) \quad (4.27)$$

Substituting equation (4.27) into equation (4.26), we notice that the desired impedance model will not be realized, but the uncertain items are bounded.

As we have noticed in Chapter 3, the interaction force estimation with AHOSMO can make the estimation error very small. We may treat the estimation error as the sensor noise since they are in the same level. Hence, we will ignore the estimation error in the analysis of impedance control accuracy. This assumption is made in all the algorithms in this chapter, and it makes the comparison of the proposed impedance control algorithms more meaningful. In fact, because neither a sensor nor necessary filtering of force sensing is required, the assumption made here is reasonable.

The past work, such as [109], did not consider the force estimation error. Also, the force estimation results were not verified with real measurements using force sensors. The estimated interaction force was taken as accurate and directly used in the impedance controller.

In the following subsections, different impedance control algorithms are proposed, and the accuracy of those impedance algorithms is compared.

The state-space representation of the Cartesian space dynamics of a robot manipulator is given here as it will be used in the impedance control algorithms. From equation (2.44), we have

$$H(X, \dot{X}) = C_C(X, \dot{X})\dot{X} + F_{VC}(X)\dot{X} + F_{CC}(X) + G_C(X) \quad (4.28)$$

Then equation (2.44) may be rewritten as

$$M_C(X)\ddot{X} + H(X, \dot{X}) = F_\tau(X) - F_e \quad (4.29)$$

Let $X_1 = X \in R^2$. The state-space representation is given as

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= -M_C^{-1}(X_1)H(X_1, X_2) + M_C^{-1}(X_1)F_\tau - M_C^{-1}(X_1)F_e \\ Y &= X_1 \end{aligned} \quad (4.30)$$

As discussed previously, only the position information is available to us, and the velocity information should be observed. Even though the controller is derived in the Cartesian space, it is eventually transformed to determine the joint torque command to be sent to the actuator in each joint.

4.2.5 Inverse Dynamics-based Impedance Controller

According to the Cartesian space dynamic model and the desired impedance model, the torque command to be sent to the joint actuator is described by

$$\begin{aligned} \tau = & \left[J^T(q) + M(q)J(q)^{-1}M_d^{-1} \right] \hat{F}_e + C(q, \dot{q})\dot{q} + F_c \cdot \text{sign}(\dot{q}) + F_v \cdot \dot{q} + G(q) \\ & + M(q)J(q)^{-1}\ddot{X}_d - M(q)J(q)^{-1} \left[M_d^{-1}B_d(J(q)\dot{q} - \dot{X}_d) + M_d^{-1}K_d(X(q) - \dot{X}_d) \right] \\ & + M(q)J(q)^{-1} - M(q)J(q)^{-1}\dot{J}(q)\dot{q} \end{aligned} \quad (4.31)$$

where $X(q)$ is the Cartesian space position of the manipulator end-effector, calculated by the forward kinematics equation. Joint space AHOSMO is used here to estimate the joint velocity and interaction force simultaneously.

The desired impedance parameters are selected as

$$M_d = \begin{bmatrix} 1 & \\ & 0.5 \end{bmatrix}, B_d = \begin{bmatrix} 0.2 & \\ & 0.2 \end{bmatrix}, K_d = \begin{bmatrix} 2.5 & \\ & 2.5 \end{bmatrix} \quad (4.32)$$

The actual Cartesian and joint space trajectories are given in Figure 4.8 and Figure 4.9, respectively. We can see that joint 2 did not pass through the singularity points. Thus, the controller is implemented properly, without needing a robust singularity handling technique.

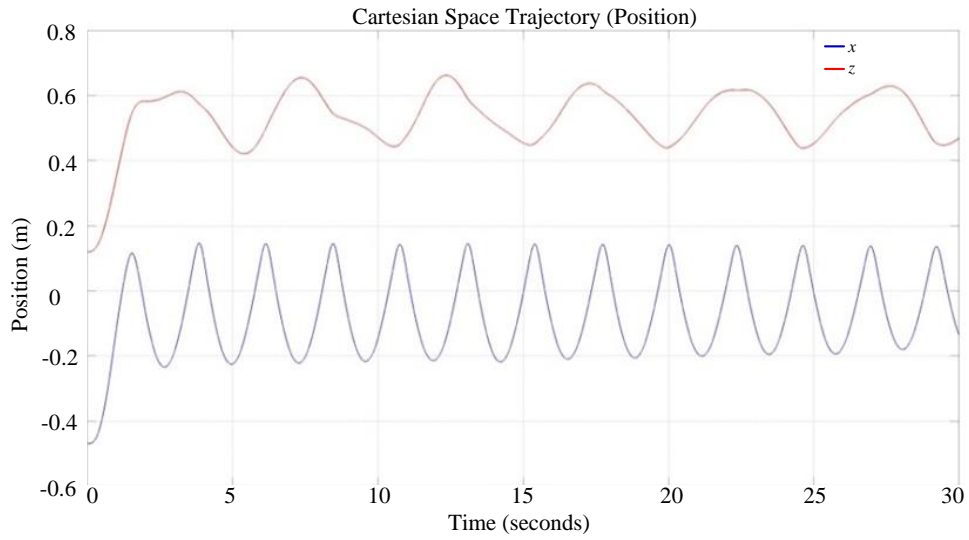


Figure 4.8: Cartesian space trajectories (with inverse dynamics-based impedance control algorithms)

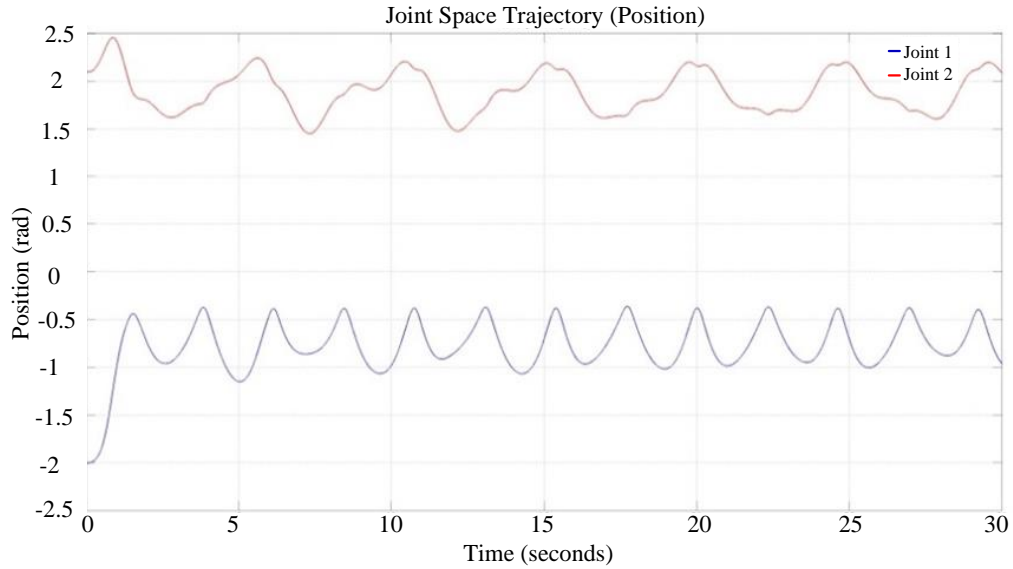


Figure 4.9: Joint space trajectories (with inverse dynamics-based impedance control algorithms)

The estimated interaction forces in the x and z directions are given in Figure 4.10 and Figure 4.11, respectively. The x direction interaction force estimated by AHOSMO appears to be noisy. Even a positive interaction force was present which is not possible. The reason may be that the controller used here directly uses the nominal dynamic parameters.

Compared with the pure position control-based interaction control, impedance control can guarantee compliance during interaction so that no excessive interaction force will be

present between the end-effector and the environment. This is an advantage of the impedance controller applied in the present project. The joint actuator torque is given in Figure 4.12.

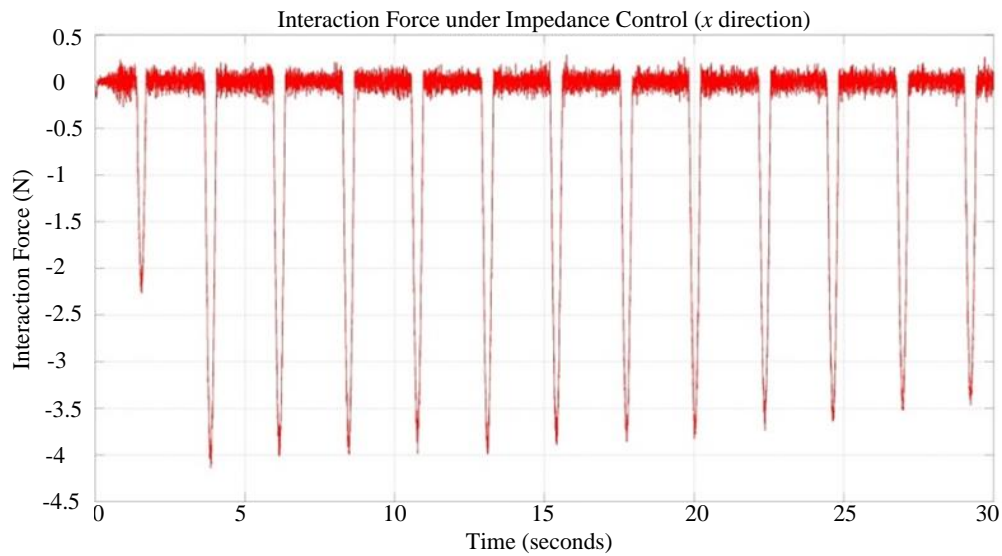


Figure 4.10: Interaction force (x direction with inverse dynamics-based impedance control algorithms)

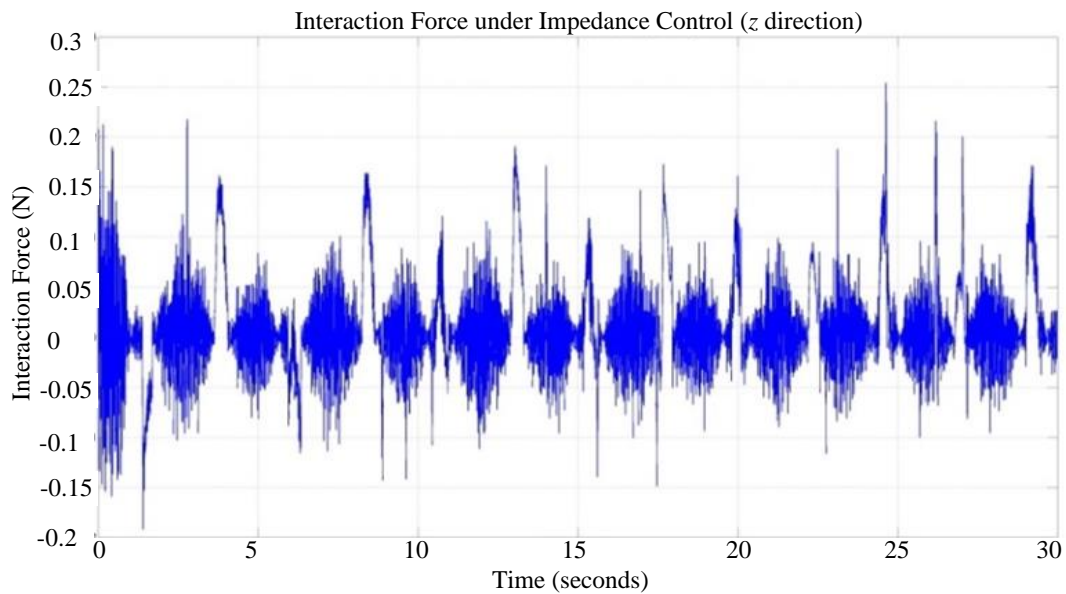


Figure 4.11: Interaction force (z direction with inverse dynamics-based impedance control algorithms)

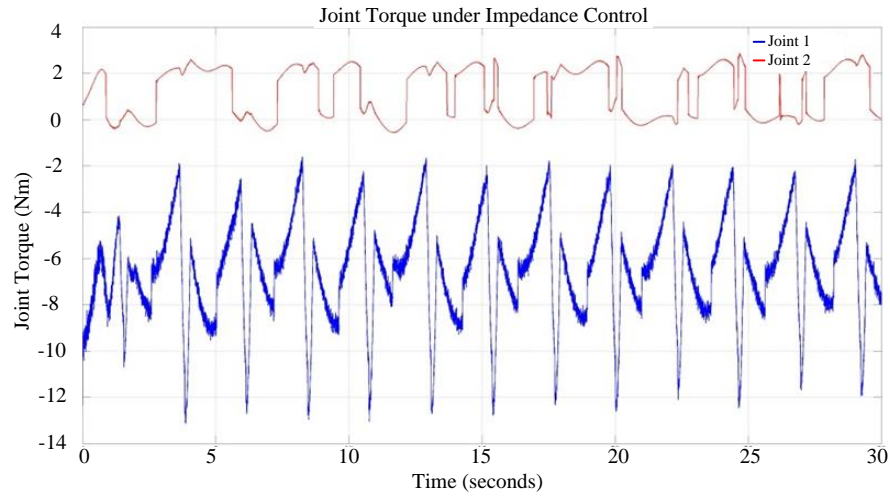


Figure 4.12: Joint torque trajectories (with inverse dynamics-based impedance control algorithms)

The impedance control accuracy of this algorithm is analyzed now. The Cartesian space acceleration that is required in the accuracy validation process is reconstructed by using

$$\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (2.32)$$

The joint acceleration may be calculated by using the robust differentiator introduced in Section 2.5.2. Also, the Cartesian space results may be calculated by directly applying the HOSMO in Section 2.5.2 to the manipulator Cartesian space dynamic model.

The impedance control accuracy is given in Figure 4.13. It is seen that the impedance control accuracy is not satisfactory due to the unavoidable uncertainty in the dynamic model.

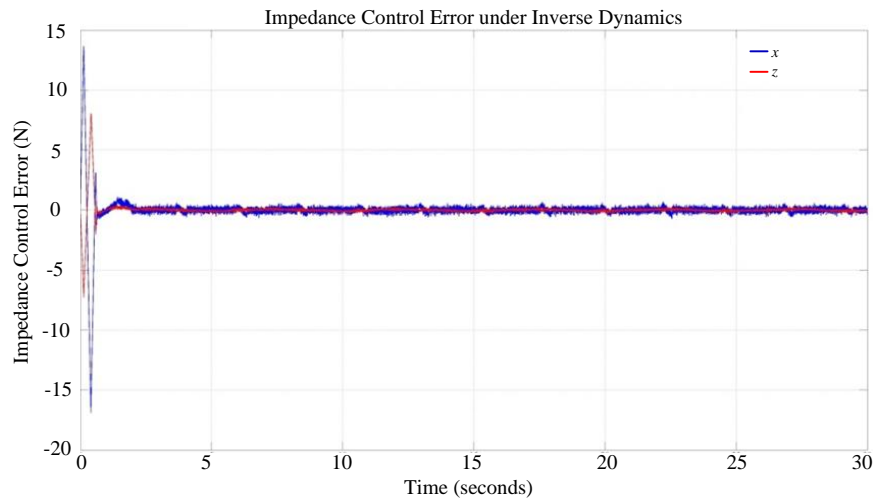


Figure 4.13: Impedance control accuracy (with inverse dynamics-based impedance control algorithms)

4.2.6 First-order Sliding Mode Controller-based Impedance Control

In the previous section, we have noticed that impedance control accuracy is not satisfactory. The robustness of the impedance controller could not be guaranteed due to measurement noise or disturbance in the plant. Sliding mode control is good at dealing with those uncertainties, to enhance the performance.

A sliding surface is defined as

$$s = \int_0^t M_d^{-1} \cdot \hat{I}(\tau) d\tau \quad (4.33)$$

The finite-time convergent AHOSMO proposed in Chapter 3 is used here for the estimation of the external interaction force and velocity. For representation simplicity, AHOSMO about the Cartesian space model is given in equation (4.34), even though the joint-space representation is adequate for experimental implementation.

$$\begin{aligned} \dot{\hat{X}}_1 &= \hat{X}_2 + \hat{\lambda}_2 |X_1 - \hat{X}_1|^{2/3} \cdot \text{sign}(X_1 - \hat{X}_1) + K_2(X_1 - \hat{X}_1) \\ \dot{\hat{X}}_2 &= M_C^{-1}(X_1)F_\tau - M_C^{-1}(X_1)H(X_1, \hat{X}_2) \\ &\quad + \hat{\lambda}_1 |\dot{\hat{X}}_1 - \hat{X}_2|^{1/2} \cdot \text{sign}(\dot{\hat{X}}_1 - \hat{X}_2) + K_1(X_1 - \hat{X}_1) + \hat{Z} \\ \dot{\hat{Z}} &= \hat{\lambda}_0 \text{sign}(\dot{\hat{X}}_1 - \hat{X}_2) \end{aligned} \quad (4.34)$$

Comparing this with the original system dynamics described in equation (4.30), the state estimation error dynamics can be described by

$$\begin{aligned} \dot{\tilde{X}}_1 &= \tilde{X}_2 - \lambda_2 |X_1 - \hat{X}_1|^{2/3} \cdot \text{sign}(X_1 - \hat{X}_1) - K_2(X_1 - \hat{X}_1) \\ \dot{\tilde{X}}_2 &= -M_C^{-1}(X_1)H(X_1, X_2) + M_C^{-1}(X_1)H(X_1, \hat{X}_2) - M_C^{-1}(X_1)F_e \\ &\quad - \lambda_1 |\dot{\tilde{X}}_1 - \hat{X}_2|^{1/2} \cdot \text{sign}(\dot{\tilde{X}}_1 - \hat{X}_2) - K_1(X_1 - \hat{X}_1) - \hat{Z} \end{aligned} \quad (4.35)$$

When the state estimation errors converge to zero, we have

$$\hat{F}_e = -M_C(X_1)\hat{Z} \quad (4.36)$$

Since the observed states are used, the sliding surface in equation (4.24) is redefined as

$$\hat{I}_e = M_d(\dot{\hat{X}}_2 - \ddot{X}_d) + B_d(\hat{X}_2 - \dot{X}_d) + K_d(\hat{X}_1 - X_d) - \hat{F}_e \quad (4.37)$$

where \hat{X}_2 , \hat{X}_1 and $\dot{\hat{X}}_2$ are obtained from the AHOSMO proposed in Chapter 3.

The sliding surface should be a function of the system states. In the state-space representation of the robot manipulator, the state variables should be position and velocity. Thus acceleration is not a state in the state-space representation. The sliding variable is represented in the integral form given by

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \int_0^t M_d^{-1} \cdot \hat{I}(\tau) d\tau = 0 \quad (4.38)$$

The derivative of this sliding surface is given as

$$\dot{s} = M_d^{-1} \cdot \hat{I}(\tau) = \dot{\hat{X}}_2 - \ddot{X}_d + M_d^{-1} B_d (\hat{X}_2 - \dot{X}_d) + M_d^{-1} K_d (\hat{X}_1 - X_d) - M_d^{-1} \hat{F}_e \quad (4.39)$$

Substituting the equation for the external interaction force observer into equation (4.39), we have

$$\begin{aligned} \dot{s} = & M_c^{-1}(X_1) F_\tau - M_c^{-1}(X_1) H(X_1, \hat{X}_2) + \lambda_1 \left| \dot{\hat{X}}_1 - \hat{X}_2 \right|^{1/2} \cdot \text{sign}(\dot{\hat{X}}_1 - \hat{X}_2) \\ & + K_1(X_1 - \hat{X}_1) + \hat{Z} - \ddot{X}_d + M_d^{-1} B_d (\hat{X}_2 - \dot{X}_d) + M_d^{-1} K_d (\hat{X}_1 - X_d) - M_d^{-1} \hat{F}_e \end{aligned} \quad (4.40)$$

The control law, which drives the system state onto this manifold, is given by

$$\begin{aligned} F_\tau = & H(X_1, \hat{X}_2) - M_c(X_1) \lambda_1 \left| \dot{\hat{X}}_1 - \hat{X}_2 \right|^{1/2} \cdot \text{sign}(\dot{\hat{X}}_1 - \hat{X}_2) - M_c(X_1) K_1(X_1 - \hat{X}_1) \\ & - M_c(X_1) \hat{Z} + M_c(X_1) \ddot{X}_d - M_c(X_1) M_d^{-1} B_d (\hat{X}_2 - \dot{X}_d) \\ & - M_c(X_1) M_d^{-1} K_d (\hat{X}_1 - X_d) + M_c(X_1) M_d^{-1} \hat{F}_e - M_c(X_1) \cdot K_g \cdot \text{sign}(s) \end{aligned} \quad (4.41)$$

$$K_g = \begin{bmatrix} k_{g1} \\ k_{g2} \end{bmatrix} \quad (4.42)$$

where $k_{g1} > 0$, $k_{g2} > 0$ are parameters to be tuned to guarantee the stabilizing feature of this controller.

This control law will lead to the representation of \dot{s} as

$$\dot{s} = -K_g \cdot \text{sign}(s) \quad (4.43)$$

Define

$$k_{\min} = \min(k_{g1}, k_{g2}) \quad (4.44)$$

The finite time convergence of this controller may be proved by using the Lyapunov function given by

$$V = \frac{1}{2} s^T s = \frac{1}{2} \|s\|_2^2 \quad (4.45)$$

Taking the time derivative of V along the trajectories of the system, we have

$$\begin{aligned}
\dot{V} &= s^T \dot{s} = -s^T \cdot K_g \cdot \text{sign}(s) \\
&= -k_{g1}s_1 \cdot \text{sign}(s_1) - k_{g2}s_2 \cdot \text{sign}(s_2) \\
&= -k_{g1}|s_1| - k_{g2}|s_2| \\
&\leq -k_{\min}(|s_1| + |s_2|) = -k_{\min} \|s\|_1
\end{aligned}$$

In view of real vector norms given in equation (4.46), and using it in the derivatives of the Lyapunov function, we have equation (4.47) for the derivative of the Lyapunov function.

$$\|s\|_2 \leq \|s\|_1 \quad (4.46)$$

$$\dot{V} \leq -k_{\min} \|s\|_1 \leq -k_{\min} \|s\|_2 \quad (4.47)$$

Equation (4.47) can be further simplified into

$$\dot{V} \leq -\sqrt{2}k_{\min} V^{1/2} \quad (4.48)$$

Stability of the plant under the proposed impedance controller is easy to verify. The finite-time convergence is further proved here, which is an important feature of the proposed controller.

Integrating both sides of equation (4.48) over time interval $[0, t]$, we get

$$V^{1/2}(t) \leq -\sqrt{2}k_{\min}t + V^{1/2}(0) \quad (4.49)$$

Thus, $V(t)$ reaches zero in finite time T_r bounded by

$$T_r \leq \frac{\sqrt{2}V^{1/2}(0)}{k_{\min}} \quad (4.50)$$

Note that if we want faster convergence, we should increase the value of k_{\min} . However, the side-effect with this increase is the increased magnitude of the discontinuous term in the control input described by equation (4.41). Chattering in this case will be more problematic. This is the tradeoff between the convergence rate and the side-effect of chattering. The impedance control results are shown in Figure 4.14 – Figure 4.16.

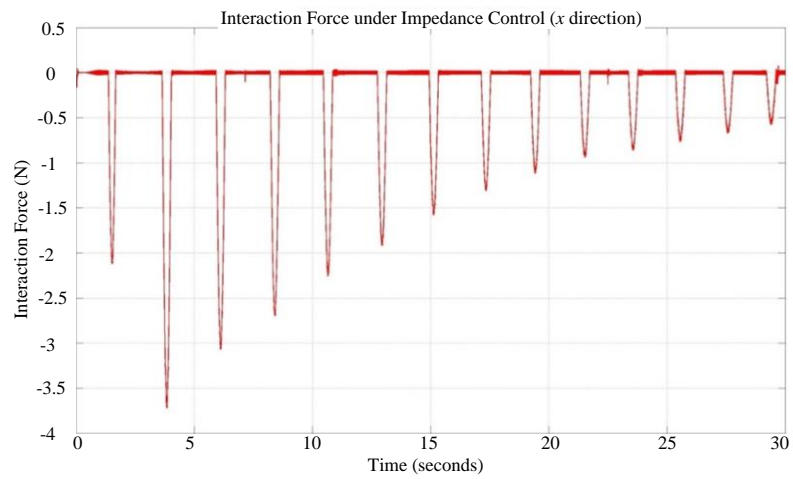


Figure 4.14: Interaction Force (x direction with SMC-based impedance control algorithms)

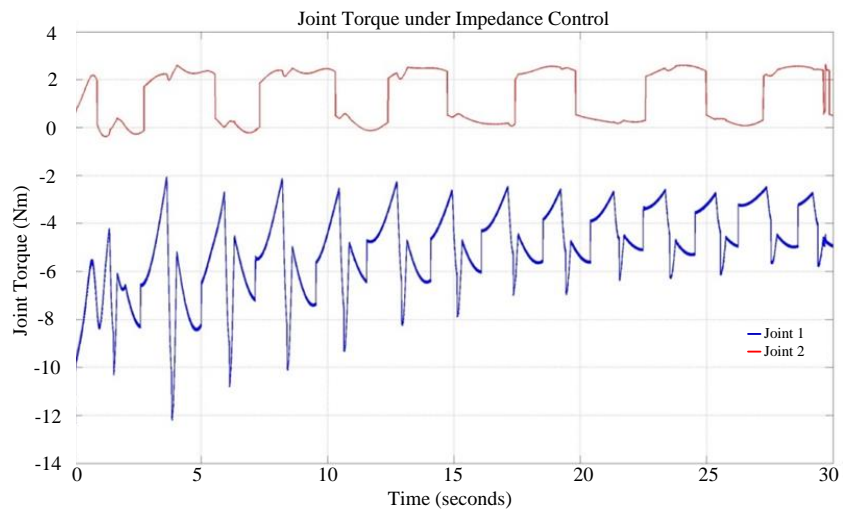


Figure 4.15: Joint torque trajectories (with SMC-based impedance control algorithms)

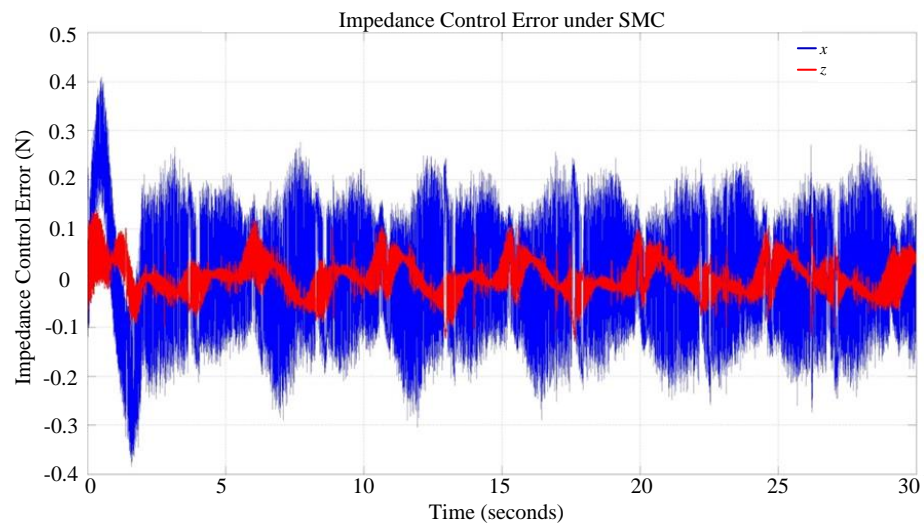


Figure 4.16: Impedance control accuracy (with SMC-based impedance control algorithms)

From these figures we can conclude that the sliding mode controller-based impedance control algorithm can slightly improve the impedance control accuracy. The application of AHOSMO given here results in a low-cost impedance control setup. High order sliding mode controllers may be used in order to further enhance the accuracy of impedance control.

4.3 Summary

In this chapter, impedance control algorithms using the observer-reconstructed interaction force are presented. In interaction control, impedance control as an indirect interaction control approach has shown advantages over pure motion control. Two impedance control algorithms are discussed and compared in terms of the accuracy of impedance control. The sliding mode control-based impedance control algorithm outperforms the inverse dynamics-based impedance controller.

Chapter 5: Applications in Bilateral Teleoperation

The present homecare robotics project is devoted to the development of a homecare system that can provide service and assistance to the elderly and the disabled people in a home environment. The homecare robots can operate in the autonomous mode when the task is routine and not highly professional. When the task is critical that needs professional assistance, then the intervention of a human operator with the necessary expertise is required. In such a situation, the homecare robots will operate in the teleoperation mode.

The impedance control algorithm that does not use force sensor, as proposed in the previous chapter is used when the robots operate in the autonomous mode. An available sensor network is used to command the robot manipulator to go to a desired Cartesian space position. Impedance control is used to ensure safe interaction between the manipulator and the human body.

In the present chapter the second mode, the teleoperation mode is briefly discussed. When the manipulator is operated in the teleoperation mode, the master side and the slave side both are impedance controlled for different purposes. The slave side impedance control will be used to ensure compliant interaction between the robot manipulator and the objects. Transparency of a teleoperation system is defined as the matching of the operator sensed impedance at the master side and the actual impedance of the environment. Master side impedance control will be used to enhance the transparency of the teleoperation system. When both sides are under impedance control, each of them may be described as a linear decoupled system, which significantly simplifies the analysis of the teleoperation system.

This chapter considers two applications of sensorless impedance control algorithms in teleoperation. The first application in section 5.1 concerns bilateral impedance control with task-dependent impedance parameters. The desired impedance parameters are selected based on the environmental properties. Stability should be a requirement in selecting these impedance parameters. This application is an improvement over [7]. In the present thesis the desired impedance parameters are selected based on a less conservative criterion as introduced in [106].

In section 5.2, the second application is presented. It formulates the impedance controlled teleoperation system in a LPV framework. Robust controllers are designed to guarantee robust performance under uncertainties of human operator and environment. The second application is an extension to the work proposed in [110]–[112] to include impedance parameter uncertainties of human operator into the LPV framework.

5.1 Bilateral Impedance Control with Task-dependent Impedance Parameters

Both master side and slave side are impedance controlled. The desired impedance parameters for the two sides are selected according to the task at hand.

The desired impedance parameters selection for the slave side is task-dependent. If the interacted object is stiff, the desired stiffness of the robot manipulator should be more compliant to avoid large interaction forces. This justifies the need to estimate the impedance parameters of the environment. An online algorithm is used to estimate the environmental impedance parameters.

Selection of the desired impedance parameters is done under the stability constraint described by the Bounded Impedance Absolute Stability (BIAS) criterion. A data base is built off-line for the selection of the desired impedance parameters under the stability constraint.

The teleoperation system in this subsection has two channels. It is usually called the position-force architecture. In the operation of the teleoperation system, the master side trajectory is sent to the slave side, while the slave side interaction force is sent back to the master side. When sensorless impedance control algorithms are used here, the estimated master side trajectory is sent to the slave side, while the AHOSMO estimated interaction force in the slave side is sent back to the master side.

The overall system architecture of this task-dependent bilateral impedance control scheme is given in Figure 5.1. Each component of the teleoperation system will be modeled. Finally, numerical simulation results will be given to show the effectiveness of the algorithm proposed in this subsection.

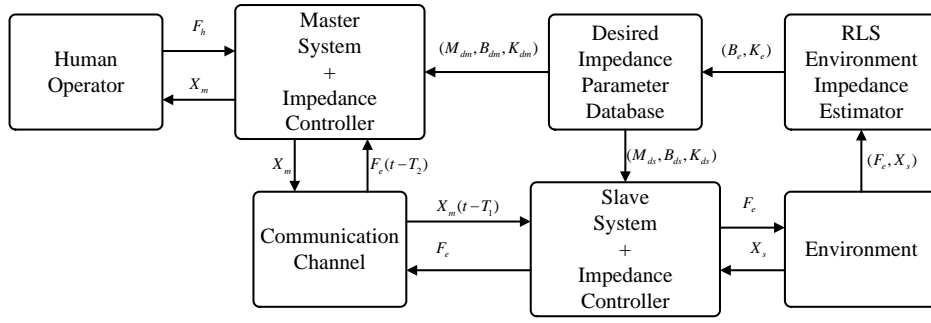


Figure 5.1: Task-dependent bilateral impedance control scheme

5.1.1 System Modeling

As indicated in Figure 1.9, the teleoperation system has five parts. The model for each part is presented in this subsection.

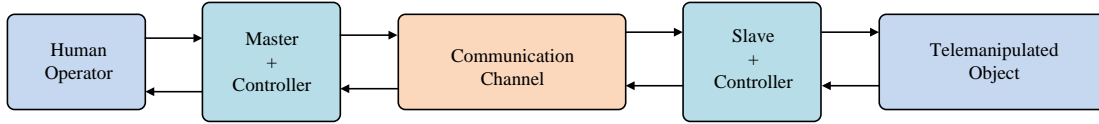


Figure 1.9: A teleoperation system

5.1.1.1 Human Operator Modeling

The dynamics of a human operator is described by the second-order linear-time-varying (LTI) equation

$$F_h = F_h^* - Z_h V_h \quad (5.1)$$

where F_h^* is the force generated by the human muscle after it has received the command from the operator's central nervous system. This force cannot be determined explicitly. However, research has been done to indicate that the EMG signals of different locations of the human operator's arm could be used to estimate F_h^* . Here $Z_h = M_h s + B_h + \frac{K_h}{s}$ is the human operator impedance in the Laplace form, with M_h , B_h , K_h denoting the inertia, damping and stiffness components, respectively; V_h is the velocity of human operator arm; and F_h is the interaction force between the human operator and the master system. Since only a numerical simulation is performed, we assume that all these parameters are known.

5.1.1.2 Master Side Impedance Control

Conventionally, the master side has a haptic device which is able to provide force feedback to the human operator. Haptic devices are delicately designed light-weight robot manipulators. In the present project, the PhantomTM Premium haptic device is used, which allows the user to directly command the interaction force, which would be felt by the human operator.

Unfortunately, the dynamic model of haptic device was embedded in the device driver and the calibration information was hidden from users. One way is to identify the dynamic model of the haptic device through experimentation. Instead, we will use the same slave side manipulator dynamic model, in simulation, to represent the master side dynamics in order to verify our algorithm. In the Cartesian space, the dynamic model of the master device may be described by

$$M_m(X_m)\ddot{X}_m + C_m(X_m, \dot{X}_m)\dot{X}_m + F_m(X_m, \dot{X}_m) + G_m(X_m) = F_h - F_m \quad (5.2)$$

where X_m is the position of the master device in the Cartesian space; $M_m(X_m)$ is the inertia matrix, which is positive definite; $C_m(X_m, \dot{X}_m)$ is the Cartesian space Centrifugal/Coriolis matrix; $F_m(X_m, \dot{X}_m)$ is the Cartesian space matrix related to friction; $G_m(X_m)$ is the Cartesian space matrix related to gravity; and F_m is the control input. Inverse dynamics-based controller given in Section 4.2.5 is used to shape the desired impedance in the master side. After using the impedance controller, the dynamics of the master side can be described by the desired impedance model

$$M_{dm}\ddot{X}_m(t) + B_{dm}\dot{X}_m(t) + K_{dm}X_m(t) = F_h(t) - F_e(t - T_2) \quad (5.3)$$

where M_{dm} , B_{dm} and K_{dm} are the master side desired inertia matrix, damping matrix and stiffness matrix, respectively; and T_2 is the latency when the interaction force is fed back to the master side from slave side. In the frequency domain, equation (5.3) may be rewritten using the Laplace transformation, as

$$(M_{dm}s + B_{dm} + \frac{K_{dm}}{s})V_m = F_h - F_e e^{-T_2 s} \quad (5.4)$$

5.1.1.3 Slave Side Impedance Control

Sliding mode-based impedance controller proposed in Section 4.2.6 is applied in the slave side. Under ideal impedance control, the slave side dynamic model can be described by its impedance model given below.

$$\begin{aligned}
& M_{ds} [\ddot{X}_s(t) - \ddot{X}_m(t - T_1)] + B_{ds} [\dot{X}_s(t) - \dot{X}_m(t - T_1)] \\
& + K_{dm} [X_s(t) - X_m(t - T_1)] = F_e
\end{aligned} \tag{5.5}$$

where M_{ds} , B_{ds} and K_{ds} are the slave side desired inertia matrix, damping matrix and stiffness matrix, respectively; and T_1 is the latency when the master side position information is sent to the slave side. In the s -domain, equation (5.5) may be rewritten as

$$(M_{ds}s + B_{ds} + \frac{K_{ds}}{s})(V_s - V_m e^{-T_1 s}) = F_e \tag{5.6}$$

5.1.1.4 Environmental Impedance Estimation

The desired impedance model of the master system and the slave system can be achieved in finite time. The desired impedance parameters are selected based on two aspects. The first one is the environmental impedance, which is estimated by the algorithm proposed in this subsection.

The environmental impedance is important for tuning of the desired impedance parameters on both sides. An online estimation algorithm should be applied to have real-time updating of the identified environmental impedance parameters.

Since the teleoperated object is a human body, the inertia parameter is not as important as the stiffness and damping properties in the environmental impedance. This may be the reason why most teleoperation schemes with environmental impedance estimation only consider the stiffness component while ignoring the other two. We will consider the damping and stiffness properties here. Then the interaction force can be described by

$$\hat{F}_e = B_e \dot{X}_s + K_e X_s = \begin{bmatrix} \dot{X}_s & X_s \end{bmatrix} \begin{bmatrix} B_e \\ K_e \end{bmatrix} = \varphi^T \theta \tag{5.7}$$

where \hat{F}_e is the measured interaction force between the slave manipulator and the environment, and B_e and K_e are the estimated damping and stiffness components, respectively, of the environmental impedance. It is reformulated into the linear regression form. Recursive least squares estimation is applied to obtain the online estimation of these two parameters. The RLS algorithm is given by:

$$\hat{\theta}_n = \hat{\theta}_{n-1} + L_n [\hat{F}_e - \varphi_n^T \hat{\theta}_{n-1}] \tag{5.8}$$

$$L_n = P_{n-1} \varphi_n (1 + \varphi_n^T P_{n-1} \varphi_n)^{-1} \quad (5.9)$$

$$P_n = [I - L_n \varphi_n^T P_{n-1}] \quad (5.10)$$

where n is the time stamp, L_n is the update gain, and P_n is the covariance matrix.

The initial values of the estimated parameters are set to be some reasonable ones within the range of typical human body impedance. Online estimation of the environment impedance is used in the tuning of the desired impedance parameters, as discussed in the subsequent subsections.

5.1.2 Selection of Desired Impedance Parameters

The desired impedance parameters of the master and slave sides should be tuned by taking stability and performance into consideration. With regard to performance, to have compliant interaction with the teleoperated object, the desired stiffness of the slave side, should be inversely proportional to the estimated environmental stiffness. For the master side, to enhance the transparency of the teleoperation system, the desired stiffness of the master system should be proportional to the estimated environmental stiffness.

Once the stiffness components in the desired impedance of the master and slave sides have been selected, the other two components are selected in the following manner. For the master side, the desired damping component is selected as the estimated environmental damping to enhance the transparency. In the slave side, to avoid the setting phase of an underdamped second-order system, the desired damping and mass component are selected to satisfy

$$B_{ds} = 2\sqrt{K_{ds}M_{ds}} \quad (5.11)$$

This provides a critically damped system. Based on this relationship, we have two free variables to select, namely M_{dm} and B_{ds} , which can be adjusted. They are constrained by the robust stability of the teleoperation system under time delay. Bounded Impedance Absolute Stability (BIAS) criterion can be used to tune these two free parameters to guarantee robust stability of the system under time delay.

The concept of BIAS is briefly reviewed here. Details are found in [106]. The network representation of the teleoperation system is shown in Figure 5.2.

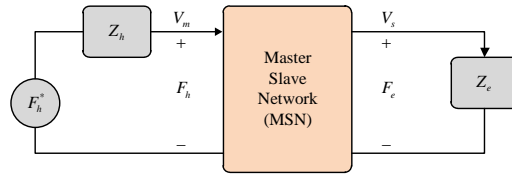


Figure 5.2: Network representation of a teleoperation system

The lumped master-communication block-slave system can be described by the hybrid representation:

$$\begin{bmatrix} F_h \\ -\dot{X}_s \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ F_e \end{bmatrix} \quad (5.12)$$

In BIAS, the power variables are transformed into wave variables as follows:

$$\begin{bmatrix} F_h - V_m \\ F_e + V_s \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} F_h + V_m \\ F_e - V_s \end{bmatrix} \quad (5.13)$$

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \left(\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \left(\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \quad (5.14)$$

It is known that the passivity of a two-port network is determined by the maximum singular value of the scattering matrix; thus,

$$\|S\|_{\infty} = \bar{\sigma}(S^T(j\omega)S(j\omega)) \leq 1 \quad (5.15)$$

where S is the scattering matrix with its elements defined in equation (5.15).

In the scattering domain, the positive realness property is transformed into a unit disk, as shown below in Figure 5.3 [106].

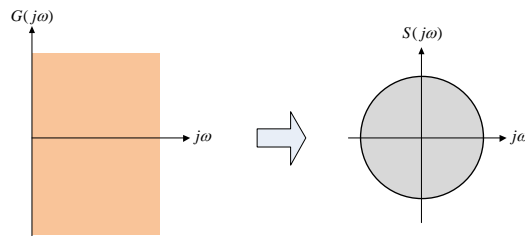


Figure 5.3: Transformation between power variables and scattering variables

Following the same procedure, the human operator and the environmental dynamics can be represented in the scattering domain. We need to determine the values of M_{dm} and B_{ds} so that the termination of the one-port network with the two-port network will result in a one-port network with scattering coefficient located within a unit circle.

The location of the impedance circles of human operator in the scattering domain are determined by the parameters of the two-port network. Some parameters will result in a passive one-port network when observed from the environment side. Since the parameters of the two-port network are functions of the desired free impedance parameters M_{dm} and B_{ds} . Finally, it turns out that the desired impedance parameters should be tuned under robust stability constraints.

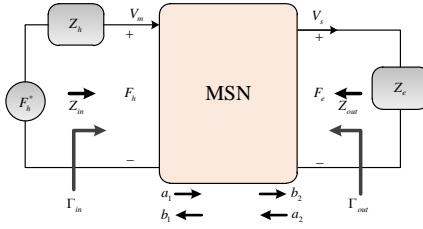


Figure 5.4: Network representation of a teleoperation system in scattering domain

Wave variables are defined as

$$a_1 = \frac{F_h + bV_m}{2\sqrt{b}}, \quad a_2 = \frac{F_e - bV_s}{2\sqrt{b}} \quad (5.16)$$

$$b_1 = \frac{F_h - bV_m}{2\sqrt{b}}, \quad b_2 = \frac{F_e + bV_s}{2\sqrt{b}} \quad (5.17)$$

where a_1 and a_2 are input wave variables from the master side and slave side, respectively; and b_1 and b_2 are output wave variables from the master side and slave side, respectively.

Three possible cases are shown in Figure 5.5 – Figure 5.7.

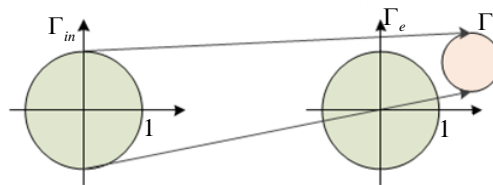


Figure 5.5: Potentially unstable in scattering domain

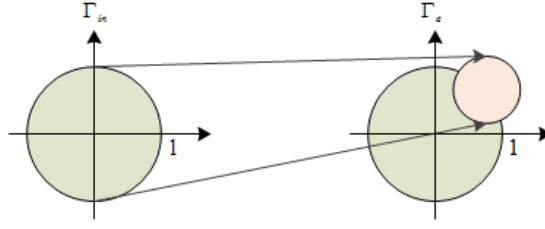


Figure 5.6: Potentially unstable in scattering domain (overlap)

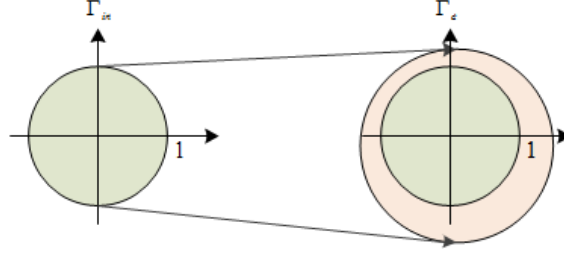


Figure 5.7: Absolutely stable in scattering domain

The desired impedance parameters should be selected so that the input passive impedance circles are located as shown in Figure 5.7 or in the shaded area of Figure 5.6 so that robust stability can be guaranteed. A database of the desired impedance parameters can be formed offline. Desired impedance parameters can be retrieved from the database so that robust stability is guaranteed and the teleoperation system transparency is optimized.

5.1.3 Transparency Analysis of the Teleoperation System

The hybrid matrix in this teleoperation scheme is given as

$$\begin{bmatrix} F_h \\ -V_s \end{bmatrix} = H(s) \begin{bmatrix} V_m \\ F_e \end{bmatrix} \quad (5.18)$$

$$\begin{bmatrix} F_h \\ -V_s \end{bmatrix} = H(s) \begin{bmatrix} V_m \\ F_e \end{bmatrix} = \begin{bmatrix} M_{md}s + B_{md} + \frac{K_{md}}{s} & -e^{-T_2s} \\ e^{-T_1s} & \frac{1}{M_{ms}s + B_{ms} + \frac{K_{ms}}{s}} \end{bmatrix} \begin{bmatrix} V_m \\ F_e \end{bmatrix} \quad (5.19)$$

With reference to the ideal hybrid matrix under time delay, we notice that the ideal transparency is not achieved. However, ideal force tracking is achieved. As discussed in [113], imperfect transparency leads to robust stability. This is a tradeoff between robust stability and performance (transparency).

5.1.4 Numerical Simulation Results

Numerical simulation results are given in Figure 5.8 – Figure 5.11. We observe that both the position tracking and force tracking of the proposed algorithm are satisfactory.

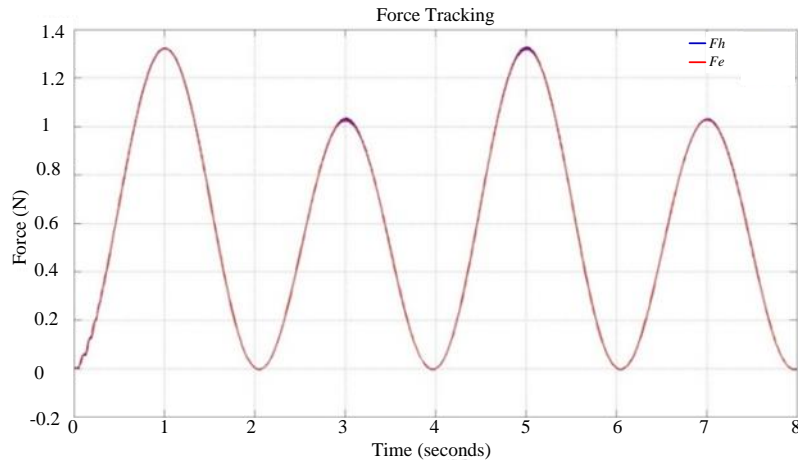


Figure 5.8: Force tracking performance

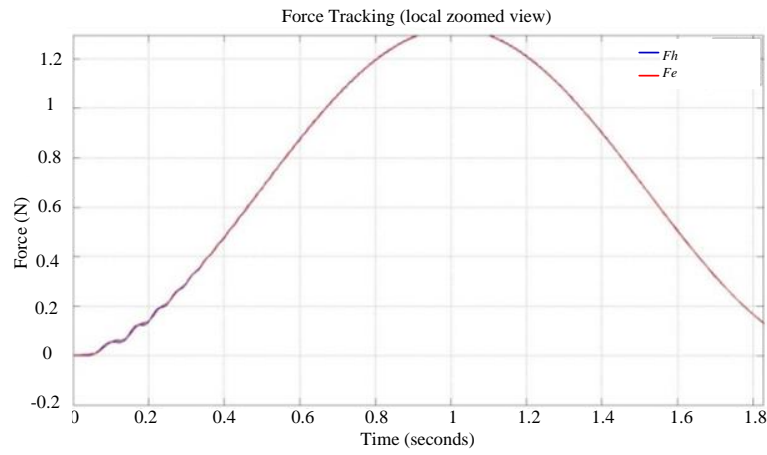


Figure 5.9: Force tracking performance (local zoomed view)

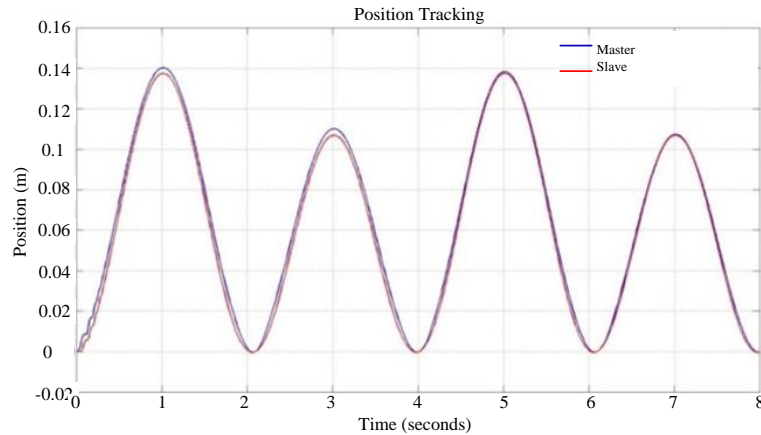


Figure 5.10: Position tracking performance

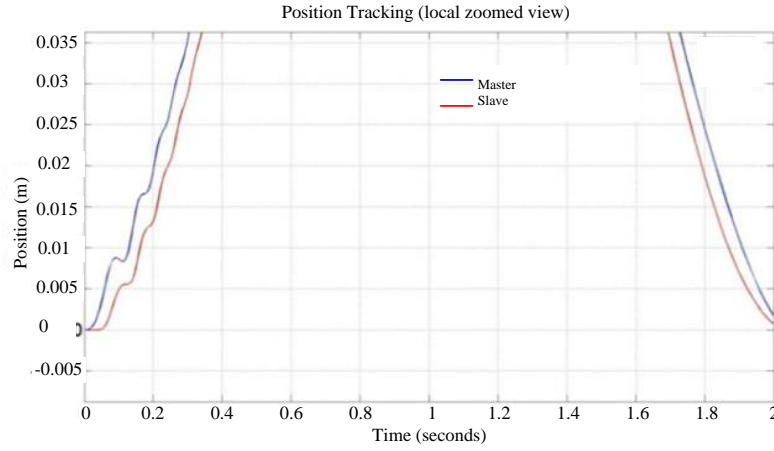


Figure 5.11: Position tracking performance (local zoomed view)

5.1.5 Conclusion & Discussion

In this subsection, a bilateral impedance controller with varying desired impedance parameters is proposed. In order to enhance the performance, the desired impedance parameters for the master side and the slave side are selected based on different principles. Also, the selected desired impedance parameters should satisfy the stability constraint. A less conservative approach to determine the robust stability of the teleoperation system—BIAS was used to form the constraint in the selection of the desired impedance parameters. Numerical results show the effectiveness of the proposed teleoperation algorithms.

5.2 Robust Performance by Gain-scheduling Control

In a teleoperation system, the operator and environment are commonly assumed to be linear time invariant (LTI), which is not entirely practical. The dynamic model of a human operator and environment are nonlinear in general. Linearization about an operating point of a nonlinear system is a common strategy in the design and analysis of a nonlinear system.

The impedance parameters of the operator and environment are time-varying. Adaptive control algorithms have been proposed to remove this limitation by estimating the impedance of the human operator and environment as it varies. After linearizing the nonlinear dynamic model, and the time-varying impedance parameters estimated, a group of controllers may be used for the estimated impedance parameters. If an impedance

parameter changes, the corresponding controller is adapted accordingly. However, stability cannot be guaranteed in general during the adaptation of a controller. The situation will become more serious when there is time delay in the communication channel.

Gain-scheduling control for linear time-varying system has found its application in this situation. A gain-scheduling controller has been used in [111] in a teleoperation system. The time-varying parameters in the linear teleoperation system are the force and position scale between the master and slave sides. The work in [112] discusses the gain-scheduling controller design about the time-varying parameters in the environmental impedance. However, in [112], the impedance of the human operator is assumed to be linear and time invariant. This is a limitation of the approach. Also, the impedance identification approach proposed there had a slow convergence rate. The present section of this thesis seeks to remove the limitations of [112]. Time-varying properties of the human operator impedance parameters are considered in the design of the gain-scheduling controller.

5.2.1 Preliminaries for Gain-scheduling Control of LPV Systems

Gain-scheduling control is widely used in the control of time-varying systems. Also, it is used in the control of a linearized nonlinear system around a time-varying operating point as the operating conditions change. If the operating points are known in advance, controllers can be designed separately for each point. When the operating point changes, then the applied controller has to be switched. Stability during the phase of controller switching cannot be guaranteed in general. This problem was solved in [114][115] by a LMI based approach where robust stability during operating point switch is guaranteed. A special case of linear time-varying system is reviewed here. The detailed proof is found in [114] and [115].

A Linear Parameter-Varying (LPV) system can be described by the following state space representation:

$$\dot{x}(t) = A(\theta(t))x(t) + B(\theta(t))w(t) \quad (5.20)$$

$$z(t) = C(\theta(t))x(t) + D(\theta(t))w(t) \quad (5.21)$$

where x is the state variable vector, z is the output vector, and w is the input vector. State-space matrices are dependent on a time-varying vector $\theta(t)$ in an affine manner. It

can represent a linear time-varying system with varying parameters. An LPV system is called polytopic if its state-space matrices $A(\theta(t))$, $B(\theta(t))$, $C(\theta(t))$, and $D(\theta(t))$ satisfy:

(i) The parameters range over a fixed polytope;

$$\theta(t) \in \Phi = Co\{w_1 \quad w_2 \quad \cdots \quad w_r\} \quad (5.22)$$

where Φ is the convex hull spanned by the extreme values of the time varying parameters.

(ii) $A(\theta(t))$, $B(\theta(t))$, $C(\theta(t))$, $D(\theta(t))$ depend on $\theta(t)$ in an affine manner.

An LPV system having quadratic performance means that the systems is asymptotically stable, and the L_2 norm of the system satisfies

$$\sup_{\omega} \frac{\|z\|_2}{\|\omega\|_2} < \gamma \quad (5.23)$$

The linear fractional transformation representation of an LPV system is given in Figure 5.12.

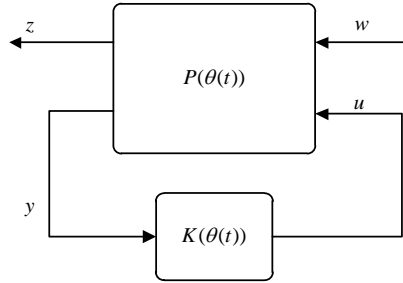


Figure 5.12: Linear fractional transformation representation of a general system

Here $P(\theta(t))$ is the generalized plant including the weighting functions and $K(\theta(t))$ is the controller. The state-space representation of the LPV generalized plant is given by

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A(\theta(t)) & B_1(\theta(t)) & B_2 \\ C_1(\theta(t)) & D_{11}(\theta(t)) & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \omega(t) \\ u(t) \end{bmatrix} \quad (5.24)$$

We wish to find a time varying controller that guarantees quadratic H_∞ performance from w to z . It has the state-space representation:

$$\begin{bmatrix} \dot{x}_k(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} A_k(\theta(t)) & B_k(\theta(t)) \\ C_k(\theta(t)) & D_k(\theta(t)) \end{bmatrix} \begin{bmatrix} x_k(t) \\ y(t) \end{bmatrix} \quad (5.25)$$

An LPV system has quadratic H_∞ performance if there exists a single matrix $X > 0$ such that

$$\begin{bmatrix} A^T X + XA & XB & C^T \\ B^T X & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0 \quad (5.26)$$

This transfers the robust performance problem into a feasibility problem of an LMI.

A polytopic LPV system having quadratic H_∞ performance is equivalent to the existence of a single matrix $X > 0$ such that equation (5.26) is satisfied for each vertex formed by the time varying parameter vector. This gives a sufficient condition for the existence of the controller (5.25). Theorem 2 given here is frequently used for the controller reconstruction and provides a sufficient and necessary condition [114][115].

Theorem 2: For a polytopic LPV system with r vertices, the sufficient and necessary condition for the existence of gain scheduling controllers boils down to the existence of two $n \times n$ matrix R and S which satisfy the following $2r+1$ LMIs:

$$\begin{bmatrix} N_R & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} A_i R + R A_i^T & R C_{1i}^T & B_{1i} \\ C_{1i} R & -\gamma I & D_{11i} \\ B_{1i}^T & D_{11i}^T & -\gamma I \end{bmatrix} \begin{bmatrix} N_R & 0 \\ 0 & I \end{bmatrix} < 0 \quad (5.27)$$

$$\begin{bmatrix} N_S & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} A_i^T S + S A_i & S B_{1i} & C_{1i}^T \\ B_{1i}^T S & -\gamma I & D_{11i}^T \\ C_{1i} & D_{11i} & -\gamma I \end{bmatrix} \begin{bmatrix} N_S & 0 \\ 0 & I \end{bmatrix} < 0 \quad (5.28)$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \quad (5.29)$$

N_R and N_S are the null spaces of $\begin{bmatrix} B_2^T & D_{12}^T \end{bmatrix}$ and $\begin{bmatrix} C_2 & D_{21} \end{bmatrix}$, respectively. ($i=1,2,3,\dots,r$)

The controller can be obtained by the following reconstruction steps:

(i) Singular value decomposition (SVD)

$$M N^T = I - R S \quad (5.30)$$

(ii) Solve a linear equation with variable X_{cl}

$$\Pi_2 = X_{cl}\Pi_1 \quad (5.31)$$

$$\text{with } \Pi_1 = \begin{bmatrix} S & I \\ N^T & 0 \end{bmatrix}, \Pi_2 = \begin{bmatrix} I & R \\ 0 & M^T \end{bmatrix}$$

(iii) Find the controller for each corresponding vertex which is indicated as

$$\begin{bmatrix} A_{ki} & B_{ki} \\ C_{ki} & D_{ki} \end{bmatrix} \text{ satisfying} \quad \begin{bmatrix} A_{ki}^T X_{cl} + X_{cl} A_{ki} & X_{cl} B_{ki} & C_{ki}^T \\ B_{ki}^T X_{cl} & -\gamma I & D_{ki}^T \\ C_{ki} & D_{ki} & -\gamma I \end{bmatrix} < 0 \quad (5.32)$$

Once the controller for each operating point is found, when the operating point changes, the corresponding gain-scheduling controller is applied for the new operating point within the convex hull. This can be done by simple interpolation, and the coefficient of interpolation for the operating point with respect to the convex hull can be directly transformed into the controller state-space matrix. Stability during switching between different controllers is guaranteed since the controller for each vertex is designed under the same set of LMI variables. The reconstruction process of the gain-scheduling controller is done in Matlab, and the derived controllers are then implemented in Simulink.

In related literature considering time varying parameters, the variation of the parameters is assumed to be infinitely fast. This is not practical and the designed controller is conservative. To obtain less conservative controllers and better performance, the rates of variation of the parameters should be considered in the controller design using the corresponding Matlab command.

5.2.2 Human Operator Modelling

As discussed in [116], the human operator model can be obtained practically through experimentation. The mathematical model is just an approximation of the experimental results. The human operator arm is modeled as a second-order mass-spring-damper system. In [117], it is pointed out that if the human operator muscle contraction level is low, and the teleoperation system is operating in a low-frequency range, the mass

component can be ignored. Only the stiffness and damping components are used to describing the dynamic model of human operator as given by

$$f_{op} - f_m = b_{op}\dot{x}_m + k_{op}x_m \quad (5.33)$$

where f_{op} is the intended force generated by the human central nervous system; b_{op} and k_{op} are damping and stiffness parameters, respectively, of the operator. These two parameters are assumed to be time varying, and are considered in the robust gain-scheduling controller design, which is given next. Also, f_m is the interaction force between the master side manipulator and the human operator.

5.2.3 Environment Modeling

The environment can be modeled by

$$f_{dist} + f_s = b_{env}\dot{x}_s + k_{env}x_s \quad (5.34)$$

Here f_s is the interaction force between the slave side manipulator and the environment; and b_{env} and k_{env} are damping and stiffness, respectively, of the environment. They are time-varying and should be considered in the gain scheduling controller design. Also, f_{dist} is the disturbance force due to the estimation error in the environmental impedance parameter.

5.2.4 Modeling of Master Slave System

For the master side, we assume that ideal impedance control is applied. The desired impedance equation is used to describe the dynamics of the master side as

$$f_m + u_m = M_m\ddot{x}_m + B_m\dot{x}_m + K_mx_m \quad (5.35)$$

Similarly, for the slave side under ideal impedance control, the dynamics may be described by

$$u_s - f_s = M_s\ddot{x}_s + B_s\dot{x}_s + K_sx_s \quad (5.36)$$

Here, M_m , B_m , and K_m are the desired impedance parameters of the master side; M_s , B_s , and K_s are the desired impedance parameters of the slave side; and u_m and u_s are

the control efforts of the master side and slave side actuators, respectively. They are determined by the controller proposed here. In this subsection we use constant desired impedance parameters.

The performance of a teleoperation system can be evaluated by the tracking error of the position and force in the two sides. In the present application, the control effort generated by the actuators in the master and slave sides is also monitored in order not to trigger the actuators saturation. The weighing functions are applied at the performance regulation channels.

The overall system is represented in Figure 5.13 following the conventional formulation of robust control.

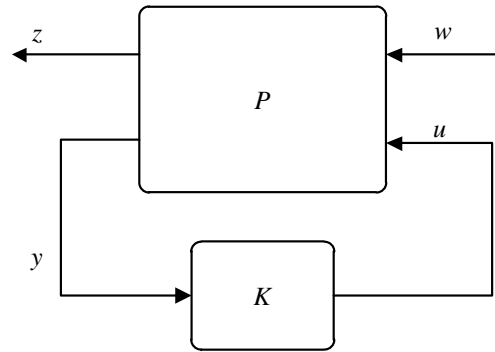


Figure 5.13: Linear fractional transformation representation of a teleoperation system

Here

$$w = \begin{bmatrix} f_{op} \\ f_{dist} \end{bmatrix}, u = \begin{bmatrix} u_m \\ u_s \end{bmatrix}, y = \begin{bmatrix} F_m \\ x_m \\ F_s \\ x_s \end{bmatrix}, z = \begin{bmatrix} W_p(x_m - x_s) \\ W_f(f_m - f_s) \\ W_m u_m \\ W_s u_s \end{bmatrix}$$

The state-space representation is as follows:

$$\begin{bmatrix} \dot{X} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} X \\ w \\ u \end{bmatrix} \quad (5.37)$$

$$\begin{bmatrix} \dot{X}_k \\ u \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \begin{bmatrix} X_k \\ y \end{bmatrix} \quad (5.38)$$

Considering the bandwidth of the human operator during teleoperation, the weighting functions for position tracking and force tracking are selected as low-pass filters, while the weighting functions for the master and slave side actuators are selected to be inversely proportional to the reciprocal of actuator saturation value. The values selected here are the same as those in [111]. They are given below.

$$W_p = \frac{20}{s+0.1}, W_f = \frac{0.04}{s+0.1}, W_m = W_s = \frac{1}{50} \quad (5.39)$$

Define $e_p = W_p(x_m - x_s)$ and $e_f = W_f(f_m - f_s)$, then we have

$$\dot{e}_p + 0.1e_p = 20(x_m - x_s) \quad (5.40)$$

$$\dot{e}_f + 0.1e_f = 0.04(f_m - f_s) \quad (5.41)$$

The state of the generalized plant is

$$X = [\dot{x}_m \quad x_m \quad \dot{x}_s \quad x_s \quad e_p \quad e_f]^T \quad (5.42)$$

The time varying parameters form the vector

$$\theta(t) = [b_{op} \quad k_{op} \quad b_{env} \quad k_{env}]^T \quad (5.43)$$

$$A(\theta) = \begin{bmatrix} -\frac{B_m + b_{op}}{M_m} & -\frac{K_m + k_{op}}{M_m} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{B_s + b_{env}}{M_s} & -\frac{K_s + k_{env}}{M_s} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 20 & 0 & -20 & -0.1 & 0 \\ -0.04b_{op} & -0.04k_{op} & -0.04b_{env} & -0.04k_{env} & 0 & -0.1 \end{bmatrix} \quad (5.44)$$

$$B_1(\theta) = \begin{bmatrix} \frac{1}{M_m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{M_s} \\ 0 & 0 \\ 0 & 0 \\ 0.04 & 0.04 \end{bmatrix}, B_2(\theta) = \begin{bmatrix} \frac{1}{M_m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{M_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.45)$$

$$C_1(\theta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, D_{11}(\theta) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, D_{12}(\theta) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{50} & 0 \\ 0 & \frac{1}{50} \end{bmatrix} \quad (5.46)$$

$$C_2(\theta) = \begin{bmatrix} -b_{op} & -k_{op} & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{env} & k_{env} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, D_{21}(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}, D_{22}(\theta) = 0 \quad (5.47)$$

We can find that each matrix given above is affine in $\theta(t)$, and the system is formulated as an LPV one with time-varying parameters $\theta(t)$. However, as discussed previously, in order to control an LPV plant by a gain-scheduling controller, the following three conditions should be satisfied:

- (1) $D_{22}(\theta) = 0$ (Satisfied in this formulation)
- (2) $B_2(\theta)$, $C_2(\theta)$, $D_{12}(\theta)$, $D_{21}(\theta)$ are independent of $\theta(t)$ (Not satisfied in this formulation)
- (3) $(A(\theta), B_2(\theta))$ are quadratically stabilizable over $B_2(\theta)$ and $(A(\theta), C_2(\theta))$ are quadratically detectable over $C_2(\theta)$ (To be determined by checking the LMI feasibility).

We see that condition (2) is not satisfied. To resolve this problem, filters are applied about the control signal u and the measured signal y [111]. This is shown in Figure 5.14.

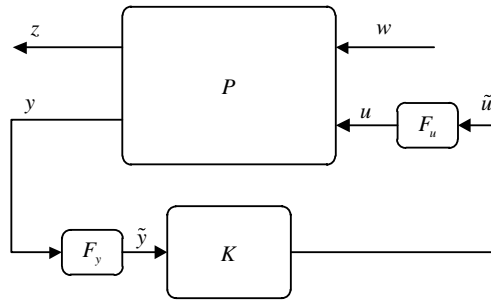


Figure 5.14: Linear fractional transformation representation of a teleoperation system with filters

Suppose that the state-space representations of the filters are:

Filter for u :

$$\begin{aligned}\dot{X}_u &= A_u X_u + B_u \tilde{u} \\ u &= C_u X_u\end{aligned}\tag{5.48}$$

Filter for y :

$$\begin{aligned}\dot{X}_y &= A_y X_y + B_y y \\ \tilde{y} &= C_y X_y\end{aligned}\tag{5.49}$$

Absorbing the filter state-space realizations into the original system, we have

$$\begin{bmatrix} \dot{X} \\ \dot{X}_u \\ \dot{X}_y \end{bmatrix} = \begin{bmatrix} A(\theta) & B_2(\theta)C_u & 0 \\ 0 & A_u & 0 \\ B_y C_2(\theta) & 0 & A_y \end{bmatrix} \begin{bmatrix} X \\ X_u \\ X_y \end{bmatrix} + \begin{bmatrix} B_1(\theta) \\ 0 \\ B_y D_{21}(\theta) \end{bmatrix} w + \begin{bmatrix} 0 \\ B_u \\ 0 \end{bmatrix} \tilde{u}\tag{5.50}$$

$$z = \begin{bmatrix} C_1(\theta) & D_{12}(\theta)C_u & 0 \end{bmatrix} \begin{bmatrix} X \\ X_u \\ X_y \end{bmatrix} + D_{11}(\theta)w\tag{5.51}$$

$$\tilde{y} = \begin{bmatrix} 0 & 0 & C_y \end{bmatrix} \begin{bmatrix} X \\ X_u \\ X_y \end{bmatrix}\tag{5.52}$$

By checking each component, we notice that the system with filters satisfies the requirement for a gain-scheduling controller. The filter designed here should take the signal bandwidth into consideration. We use first order low-pass filters here, with cut-off frequency 20 rad/s.

In application of a gain-scheduling controller to the LPV teleoperation system, each component in the time varying vector has two bounds (the lower bound and the upper bound of human operator and environment impedance parameters). Therefore, there are up to $2^4=16$ combinations. The teleoperation system is expressed as a convex combination of polytopes since the parameter dependence of the teleoperation system is affine.

5.2.5 Numerical Simulation Results

The parameters given in equation (5.53) are used in the following numerical simulation.

$$f_{op}(t) = 0.6\cos(\pi t) + 0.15\sin\left(\frac{\pi t}{2}\right)\tag{5.53}$$

$$b_{op} \in [3, 8], k_{op} \in [500, 550] \quad (5.54)$$

$$b_{env} \in [20, 25], k_{env} \in [100, 150] \quad (5.55)$$

$$Z_m = Z_s = 0.072s^2 + 0.5s + 5 \quad (5.56)$$

The disturbance signal f_{dist} is assumed to be Gaussian white noise as shown in Figure 5.15. During simulation, the impedance parameters of the operator and the environment are selected as time-varying with the impedance range given in equation (5.54) and equation (5.55). The force tracking error and position tracking error are shown in Figure 5.16 and Figure 5.17, respectively.

From these results we observe that the position tracking performance and the force tracking performance are satisfactory when the operator and environment impedances are varying within the convex hull. Also, we see that the system is stable during the switching between the gain scheduling controllers.

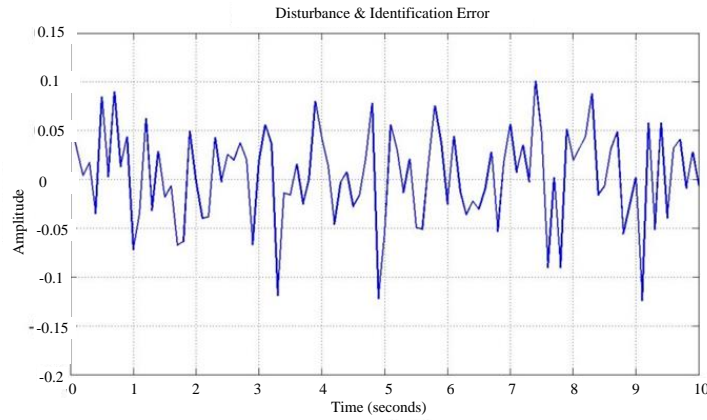


Figure 5.15: Signal for environment disturbance with identification error

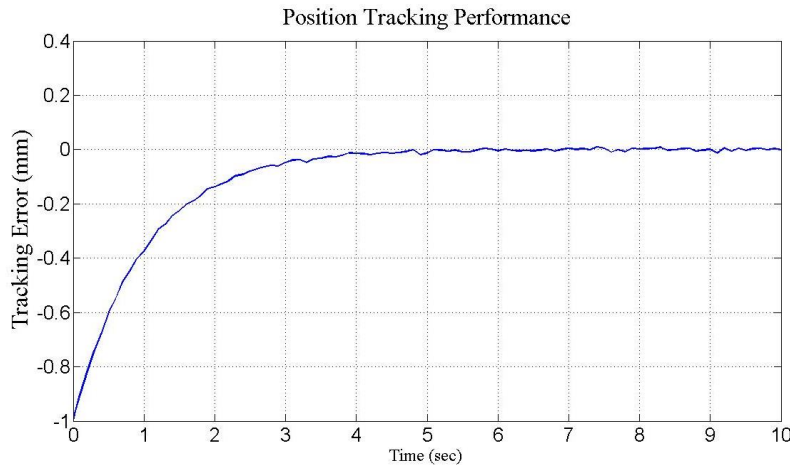


Figure 5.16: Position tracking error

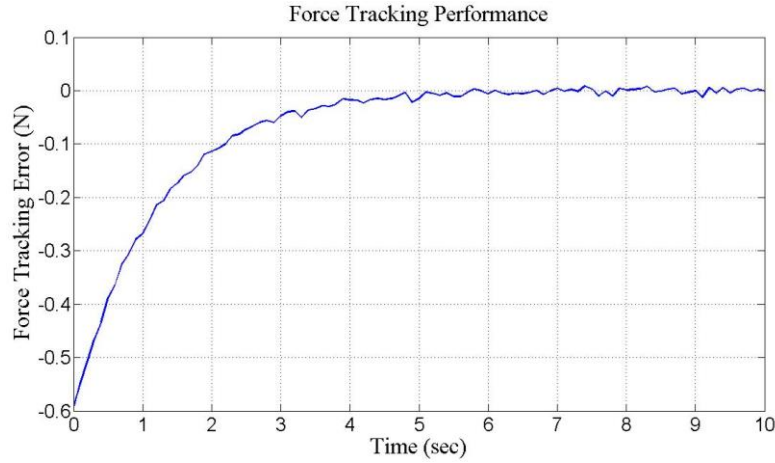


Figure 5.17: Force tracking error

5.2.6 Summary

A teleoperation system with both time-varying operator impedance and environmental impedance is analyzed in this subsection. The transparency of the teleoperation system is optimized. The control efforts by the manipulator actuators are also considered as the parameters to be monitored.

The teleoperation system is formulated as an LPV system with time-varying parameters, which are the impedances of human operator and environment. To design a gain-scheduling controller, filters are applied to the control signal and to the measured output of the generalized plant. The teleoperation system with filters is shown to satisfy the solvability condition and a controller is designed based on gain scheduling.

The gain-scheduling controller designed here is conservative since it assumes the variation of the time-varying parameters to be infinitely fast. A less conservative controller that takes the rate of change of the parameters into consideration is preferred. Also, time delay should be considered in the controller design.

Chapter 6: Conclusions and Future Work

This thesis analyzed the challenges in the design of a homecare robotic system. The work focused on the development of effective strategies for the control of robot interaction when a homecare robot manipulator assists a care-receiver. Two modes of operation of a homecare robot were studied—autonomous operation and haptic teleoperation. Modeling and model identification (experimental modeling) were investigated because a robot model was needed in the developed control schemes. In particular, interaction control with the estimation of the interaction force in the absence of direct force sensing was studied and associated methodologies were developed. Impedance control in the teleoperation mode was investigated and methodologies were developed. The developed techniques were implemented in a homecare system and evaluated using both simulation and experimentation.

In the present chapter, a summary of the research activities of the present thesis is given and conclusions are made on the overall investigation. The key research contributions and the importance of them are summarized. Finally, the limitations of the present research are pointed out. Possible future work that may make further contributions in the present area and may improve the performance of a homecare robotic system are suggested.

6.1 Conclusions

In this thesis dynamic control algorithms that would ensure the safe interaction between a robot manipulator and a care-receiver in a homecare robotic system were investigated, designed, analyzed, and evaluated in a systematic manner. Impedance control as a strategy of indirect force control was used for interaction control. It could avoid exertion of excessive interaction force on human body during a care-giving task. The developed impedance control algorithm showed important advantages when compared with a position-based interaction control algorithm.

As a prerequisite for the subsequent development, the manipulator used in the present homecare project was described. In order to conveniently verify the developed methodologies, the hardware system and software system of the laboratory robot were redesigned based on Matlab xPc Target technology. Two out of the four joints of the

manipulator were locked to convert it into a two-DOF system for convenience of the experimental studies. Specifically, the resulting planar two-DOF manipulator significantly simplified the dynamic control problem.

The manufacturer of the robotic system provided dynamic parameters of the robot, which were tested using inverse dynamics-based joint space trajectory tracking experiments, and were found to be inaccurate. The dynamic parameters from the manufacturer were derived from CAD models before fabrication, and errors are unavoidable during the manufacturing process. Since an accurate dynamic model was required for the subsequent investigation, dynamic model identification using experiments was carried out.

Interaction force was found to be a key parameter in representing the state of interaction between a robot manipulator and a manipulated object (care-receiving human). Many limitations and drawbacks were observed in using a force sensor for direct measurement of the interaction force. These included high cost, difficulty of mounting the sensor at the interface of the end-effector and manipulated object, variance of working conditions, and reduction of the mechanical bandwidth of the original system due to the compliant structure of the force sensor. Consequently, the interaction force was estimated using an adaptive third-order sliding mode observer. The estimated interaction force was compared with the measurements from a force sensor. The effectiveness of the adaptive third-order sliding mode observer in interaction force estimation was verified in this manner.

The interaction force estimated by using the observers was applied in the impedance control algorithm to form an observer-controller framework. It was found that the sliding mode-based impedance control algorithm outperformed the inverse dynamics-based impedance control algorithm. In particular, the accuracy of impedance control was improved when compared with the inverse dynamics based impedance control algorithm.

The proposed impedance control algorithm was used in two teleoperation applications. The first one was teleoperation through bilateral impedance control with task-dependent desired impedance parameters. The master side impedance control was needed for transparency enhancement. The slave side impedance control was important for safe interaction between the robot manipulator and the work environment (care-receiver).

The second application of the sliding mode-based impedance control algorithms was in the controller synthesis with robust performance under uncertainties in the human operator and the environment. To include these uncertainties, the overall teleoperation system was formulated into an LPV system. Transparency was optimized by finding a feasible controller for the LPV system. The effectiveness of impedance control algorithms in teleoperation tasks was verified by numerical simulation.

6.2 Contributions

The primary contribution of the present investigation was the development and evaluation of a robust impedance control algorithm based on identification (AHOSMO). The developed algorithm could ensure finite-time realization of the desired impedance behavior. This greatly simplified the analysis of the teleoperation system with bilateral impedance control, since the separation principle in a linear system played a role here. In summary, the following contributions were made in this thesis

- (i) The dynamic model of a reconfigured robot manipulator was identified using different dynamic model identification algorithms. In order to obtain accurate estimation of velocity and acceleration, adaptive sliding mode based differentiators were used. Off-line dynamic parameter identification was proposed. However, in the validation process, the accuracy was found to be unsatisfactory. On-line experimental identification of dynamic parameters indicated some uncertainty in the friction parameters. A Neural Network-based torque compensator was proposed to resolve this problem. In the validation phase, the Neural Network-based compensator together with the off-line identified dynamic parameters showed better performance in terms of accuracy.
- (ii) An adaptive high-order sliding mode observer (AHOSMO) was developed to simultaneously estimate the interaction force and the states of the robot. This was shown to eliminate the unavoidable drawbacks and limitations of using a force sensor. The observed interaction force results were compared with the force sensor measurements and the validity was illustrated. The estimated states would be useful in the controller design.
- (iii) A sliding mode-based impedance control algorithm was proposed and developed. It showed better impedance control accuracy when compared with an inverse

dynamics-based impedance control algorithm. The observed interaction force was used in the impedance control algorithm.

- (iv) The proposed impedance controllers were implemented in a teleoperation system. Improvements were made to two teleoperation applications that had been proposed. Numerical simulations were given to show the improvements in the teleoperation systems.
- (v) A new setup for dynamic control of a robot manipulator was developed. It was based on Matlab/Simulink xPC Target platform. This platform takes advantage of the numerous functions and interfaces of Matlab/Simulink. It will facilitate the development and validation of different control algorithms.

6.3 Significance of the Work

The algorithms proposed in the current work may be directly used in a robotic homecare system. It will ensure safe interaction between the manipulator and the care-receiver. What is more, the developed impedance control strategy, which does not use direct force sensing, would find applications in other fields as well.

There are applications where mounting a force sensor in the end-effector is impossible. For example, in tele-surgery the surgical robot interacts with the organs of the patient, and the interaction force should be carefully monitored. However, due to sanitary, mechanical, and other issues, mounting a force sensor at the surgical interface would not be feasible. Currently in this field, ultrasound and medical imaging technologies are applied to monitor the tissue deformation which will eventually lead to estimation of the interaction force. The research presented in this thesis may be used to estimate the interaction force without needing expensive medical imaging equipment.

In conclusion, this thesis proposed a low-cost impedance control strategy which could be used by other interaction control tasks of robot manipulators.

6.4 Limitations and Suggested Future Work

Several assumptions were made in this thesis, which in fact correspond to the limitations of the work. Some further work should be done to release some of the assumptions and to make it more broadly acceptable. Some of these are listed below.

- (i) A more complex and realistic friction model may be applied instead of the Coulomb and viscous friction models. Identification of and compensation for robot friction is a complex problem. An improved friction model would provide a more accurate description and quantification of the frictional torque and force in a robot.
- (ii) More advanced learning algorithms other than a simple Neural Network may be applied to compensate for uncertainty in the dynamic model of the manipulator.
- (iii) Once the dynamic model of a manipulator is identified, extended Kalman filter (EKF) may be applied to reconstruct the joint velocity and acceleration. The velocity and acceleration based on EKF could be compared with the results obtained from sliding mode-based observer/differentiators to determine which approach is more effective.
- (iv) The HOSMO used for simultaneous estimation of system states and interaction force is based on the homogeneity theory of differential equations. The proof of its adaptive version is based on the assumption that the gains are available, even though they are unknown. A uniform framework to design the HOSMO will make the proof of convergence more concise and elegant. It is likely this can be done in a framework of homogeneity theory of differential equations. A systematic way to propose a Lyapunov function for a high-order sliding mode system is still a difficult problem.
- (v) The sliding mode controller that was used in Chapter 4 for impedance control is a first-order one. Chattering is unavoidable in the system. High-order sliding mode controller for impedance control would improve the performance. However, the proof of convergence of system dynamics to the sliding surface will be more complex. This may be investigated in a framework of homogeneity theory. This will unify the problems of controller and observer design.
- (vi) In the present work, the application of the impedance control algorithms in teleoperation were only verified in numerical simulation, due to the absence of an accurate dynamic model for the haptic device. Also, the assumptions made about human operators and the environments are not quite practical, and accurate dynamic models of them can be very complex. More general methodologies of

stability analysis and controller synthesis may be investigated to extend the applicability of the developed algorithms.

Bibliography

- [1] Y. Wang, H. Lang, and C. W. de Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 5, pp. 757–769, 2010.
- [2] M.T. Khan and C. W. de Silva, "Autonomous fault tolerant multi-robot coordination for object transportation based on Artificial Immune System," *Proc. 2nd Int. Conf. Robot Commu. Coordi.*, pp.1-6, 2009.
- [3] Y. Matsumoto, S. Katsura, and K. Ohnishi, "An analysis and design of bilateral control based on disturbance observer," *Proc. IEEE Int. Conf. Ind. Technol.*, pp. 802–807, 2003.
- [4] N. Hogan, "Impedance control: An approach to manipulation," *Proc. Am. Control Conf.*, pp. 304-313, 1985.
- [5] B. Technology, "Whole Arm Manipulation (WAM) User Manual" Barrett Technology, <http://support.barrett.com/wiki/WAM>, 2009.
- [6] Y. Wang and C. W. de Silva, "An experimental setup for dynamic control of robot manipulators," *Proc. IEEE Int. Conf. Comp. Sci. Edu.*, pp. 123-128, 2014.
- [7] H. Cho, J. Park, K. Kim, and J. Park, "Sliding-mode-based impedance controller for bilateral teleoperation under varying time-delay," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1025–1030, 2001.
- [8] M. Walker, "Efficient dynamic computer simulation of robotic mechanisms," *ASME J. Dyn. Syst. Meas.*, vol. 104, no. 3, pp. 205-211, 1982.
- [9] K. D. Young, V. I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 3, pp. 328–342, 1999.
- [10] E. Eng, "Sliding mode: basic theory and new perspectives," *"5th Workshop on Structural Dynamical Systems: Computational Aspect"*, 2008.
- [11] R. DeCarlo, S. Zak, and G. Matthews, "Variable structure control of nonlinear multivariable systems: a tutorial," *Proc. IEEE*, vol. 76, no. 3, pp. 212–232, 1988.
- [12] M. Duarte-Mermoud and P. La Rosa, "Designing SISO observers with unknown input," *IMA J. Math. Control*, vol. 20, pp. 387–391, 2003.

- [13] P. Menold, "Finite time convergent observers for nonlinear systems," *Proc. 11th Mediterranean Conf. Control. Autom.*, pp. 5673–5678, 2003.
- [14] T. Floquet and J. P. Barbot, "A sliding mode approach of unknown input observers for linear systems," *Proc. 43rd IEEE Conf. Decis. Control*, Vol.2, pp. 1724–1729, 2004.
- [15] S. K. Spurgeon, "Sliding mode observers: a survey," *Int. J. Syst. Sci.*, vol. 39, no. 8, pp. 751–764, 2008.
- [16] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, 1998.
- [17] J. Y. Hung, W. Gao, and J. C. Hung, "Variable structure control: a survey," *IEEE Trans. Ind. Electron.*, vol. 40, no. 1, pp. 2–22, 1993.
- [18] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding mode control and observation*. New York, NY: Springer New York, 2014.
- [19] V. I. Utkin and A. S. Poznyak, "Adaptive sliding mode control with application to super-twist algorithm: equivalent control method," *Automatica*, vol. 49, no. 1, pp. 39–47, 2013.
- [20] V. Utkin, J. Guldner, and M. Shijun, *Sliding mode control in electro-mechanical systems*. Philadelphia, CA, Taylor & Francis, 1999.
- [21] M. Chen, "A state-dependent boundary layer design for sliding mode control," *IEEE Trans. Automat. Contr.*, vol. 47, no. 10, pp. 1677–1681, 2002.
- [22] H. Lee, E. Kim, H.-J. Kang, and M. Park, "A new sliding-mode control with fuzzy boundary layer," *Fuzzy Sets Syst.*, vol. 120, no. 1, pp. 135–143, 2001.
- [23] L. Fridman and Y. Shtessel, "Higher-order sliding-mode observer for state estimation and input reconstruction in nonlinear systems," *Int. J. Robust Nonlinear Control*, vol.18, no.4, pp. 399–412, 2008.
- [24] S. Emel'yanov, S. Korovin, and A. Levant, "High-order sliding modes in control systems," *Comput. Math. Modeling*, vol. 7, no. 3, pp. 294–318, 1996.
- [25] M. Taleb, a. Levant, and F. Plestan, "Twisting algorithm adaptation for control of electropneumatic actuators," *Proc. 12th Int. Work. Var. Struct. Syst.*, no. 2, pp. 178–183, 2012.
- [26] A. Ferrara and M. Rubagotti, "A sub-optimal second order sliding mode controller for systems with saturating actuators," *IEEE Trans. Automat. Contr.*, vol. 54, no. 5, pp. 1082–1087, 2009.

- [27] S. Mohammed, P. Fraisse, D. Guiraud, P. Poignet, and H. El Makssoud, "Robust control law strategy based on high order sliding mode: towards a muscle control," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 2644–2649, 2005.
- [28] A. Estrada and L. Fridman, "Quasi-continuous HOSM control for systems with unmatched perturbations," *Proc. Int. Work. Var. Struct. Syst.*, pp. 179–184, 2008.
- [29] A. Polyakov and A. Poznyak, "Lyapunov function design for finite-time convergence analysis: twisting controller for second-order sliding mode realization," *Automatica*, vol. 45, no. 2, pp. 444–448, 2009.
- [30] A. Levant, "Construction principles of output-feedback 2-sliding mode design," *Proc. 41st Conf. Decis. Control*, pp. 317–322, 2002.
- [31] J. Rivera and A. Loukianov, "Integral nested sliding mode control: application to the induction motor," *Proc. Int. Work. Var. Struct. Syst.*, pp. 110–114, 2006.
- [32] A. Levant, "Homogeneity approach to high-order sliding mode design," *Automatica*, vol. 41, no. 5, pp. 823–830, 2005.
- [33] A. Levant, "Finite differences in homogeneous discontinuous control," *IEEE Trans. Automat. Contr.*, vol. 52, no. 7, pp. 1208–1217, 2007.
- [34] A. Levant and L. M. Fridman, "Accuracy of homogeneous sliding modes in the presence of fast actuators," *IEEE Trans. Automat. Contr.*, vol. 55, no. 3, pp. 810–814, 2010.
- [35] A. Levant, "Universal single-input-single-output (SISO) sliding-mode controllers with finite-time convergence," *IEEE Trans. Automat. Contr.*, vol. 46, no. 9, pp. 1447–1451, 2001.
- [36] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993.
- [37] I. Boiko, L. Fridman, and R. Iriarte, "Analysis of chattering in continuous sliding mode control," *IEEE Trans. Automat. Contr.*, vol. 50, no. 9, pp. 1442–1446, 2005.
- [38] G. David, "An introduction to observers," *IEEE Trans. Automat. Contr.*, vol. 16, no. 6, pp. 596–602, 1971.
- [39] V. Utkin, *Sliding modes in control and optimization*, Springer-Verlag, 1992.
- [40] F. Bejarano, "Exact state estimation for linear systems with unknown inputs based on hierarchical super-twisting algorithm," *Int. J. Robust Nonlinear Control*, vol. 17, no. 18, pp. 1734–1753, 2007.

- [41] T. Floquet and J. P. Barbot, "Super twisting algorithm-based step-by-step sliding mode observers for nonlinear systems with unknown inputs," *Int. J. Syst. Sci.*, vol. 38, no. 10, pp. 803–815, 2007.
- [42] J. Davila, L. Fridman, and A. Levant, "Second-order sliding-mode observer for mechanical systems," *IEEE Trans. Automat. Contr.*, vol. 50, no. 11, pp. 794–797, 2005.
- [43] M. Iqbal and A. Bhatti, "Robust parameter estimation of nonlinear systems using sliding-mode differentiator observer," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 680–689, 2011.
- [44] J. Davila, M. Basin, and L. Fridman, "Finite-time parameter identification via high-order sliding mode observer," *Proc. Am. Control Conf.*, no. 1, pp. 2960–2964, 2010.
- [45] J. Davila, L. Fridman, and A. Poznyak, "Observation and identification of mechanical systems via second order sliding modes," *Int. J. Control*, vol. 79, no. 10, pp. 232–237, 2006.
- [46] Y. B. Shtessel, J. a. Moreno, F. Plestan, L. M. Fridman, and A. S. Poznyak, "Super-twisting adaptive sliding mode control: A Lyapunov design," *Proc. 49th IEEE Conf. Decis. Control*, pp. 5109–5113, 2010.
- [47] R. Sameni and M. B. Shamsollahi, "Filtering noisy ECG signals using the extended Kalman filter based on a modified dynamic ECG model," *Proc. 32nd Annu. Int. Conf. Comput. Cardiology.*, pp. 1017–1020, 2005.
- [48] L. Thomasset and X. Bideaux, "Adaptive higher order sliding modes for two-dimensional derivative estimation," *Proc. 18th IFAC World Congress.*, vol. 18., no.1, pp.3062-3071, 2011.
- [49] L. Sidhom and M. Pham, "Identification of a robot manipulator based on an adaptive higher order sliding modes differentiator," *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, pp. 1093-1098, 2010.
- [50] A. Levant, "Quasi-continuous high-order sliding-mode controllers," *IEEE Trans. Automat. Contr.*, vol. 50, no. 11, pp. 1812–1816, 2005.
- [51] M. Cavusoglu and D. Feygin, "Kinematics and dynamics of PHANToM (TM) model 1.5 haptic interface," *University of California at Berkeley, Electronics Research Laboratory Memo M1.*, 2001.
- [52] A. Tahmasebi and B. Taati, "Dynamic parameter identification and analysis of a PHANToM haptic device," *Proc. IEEE Conf. Control Applications*, pp. 1251–1256, 2005.

- [53] A. Janot, P. O. Vandanjon, and M. Gautier, "Identification of robots dynamics with the Instrumental Variable method," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1762–1767, 2009.
- [54] M. Gautier, P.O. Vandanjon, and A. Janot, "Dynamic identification of a 6 dof robot without joint position data," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 234–239, 2011.
- [55] Z. Bingül and O. Karahan, "Dynamic identification of Staubli RX-60 robot using PSO and LS methods," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4136–4149, 2011.
- [56] M. Iwasaki, M. Miwa, and N. Matsui, "GA-based evolutionary identification algorithm for unknown structured mechatronic systems," *IEEE Trans. Ind. Electron.*, vol. 52, no. 1, pp. 300–305, 2005.
- [57] K. Ahn and H. Huy Anh, "System modeling and identification the two-link pneumatic artificial muscle (PAM) manipulator optimized with genetic algorithms," *Proc. Int. Jt. Conf. SICE-ICASE*, pp. 4744–4749, 2006.
- [58] B. Bukkems and D. Kostić, "Online identification of a robot using batch adaptive control," *Proc. 13th IFAC Symp. System Identification*, pp. 953–958, 2003.
- [59] G. J. Garcia, J. a Corrales, J. Pomares, and F. Torres, "Survey of visual and force/tactile control of robots for physical interaction in Spain," *Sensors*, vol. 9, no. 12, pp. 9689–733, 2009.
- [60] J. De Schutter and H. Bruyninckx, "Force control: a bird's eye view," *Control Probl. Robotics Automation*, pp. 1–17, Springer Berlin Heidelberg, 1998.
- [61] H. Sage, "Robust control of robot manipulators: a survey," *Int. J. Control*, vol. 72, no.16, pp. 1498–1522, 1999.
- [62] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: a tutorial," *Automatica*, vol. 25, no. 6, pp. 877–888, 1988.
- [63] R. Seifabadi, S. M. Rezaei, S. Shiry, and M. Saadat, "Robust impedance control of a delayed telemanipulator considering hysteresis nonlinearity of the piezo-actuated slave robot," *Haptics: Perception, Devices and Scenarios*, pp. 63–72. Springer Berlin Heidelberg, 2008.
- [64] H. C. Cho and J. H. Park, "Stable bilateral teleoperation under a time delay using a robust impedance control," *Mechatronics*, vol. 15, no. 5, pp. 611–625, 2005.
- [65] J. Garcia and A. Robertsson, "Sensor fusion for compliant robot motion control," *IEEE Trans. Robot. Autom.*, vol. 24, no. 2, pp. 430–441, 2008.

- [66] S. Katsura, Y. Matsumoto, and K. Ohnishi, "Modeling of force sensing and validation of disturbance observer for force control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 530–538, 2007.
- [67] P. J. Hacksel and S. E. Salcudean, "Estimation of environment forces and rigid-body velocities using observers," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 931–936, 1994.
- [68] K. S. Eom, I. H. Suh, W. K. Chung, and S.-R. Oh, "Disturbance observer based force control of robot manipulator without force sensor," *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, pp. 3012–3017, 1998.
- [69] K. Ohishi, M. Miyazaki, M. Fujita, and Y. Ogino, " H_∞ observer based force control without force sensor," *Proc. Int. Conf. Ind. Electron. Control, Intrtum.*, no. 4, pp. 1049–1054, 1991.
- [70] A. Alcocera and A. Robertssona, "Force estimation and control in robot manipulators," *Proc. 7th Symp. Robot Control*, pp. 31–36, 2003.
- [71] M. Van Damme, P. Beyl, B. Vanderborght, V. Grosu, R. Van Ham, I. Vanderniepen, a. Matthys, and D. Lefeber, "Estimating robot end-effector force from noisy actuator torque measurements," *Proc. IEEE Int. Conf. Robot. Autom.*, no. 4, pp. 1108–1113, 2011.
- [72] T. Murakami, F. Yu, and K. Ohnishi, "Torque sensorless control in multidegree-of-freedom manipulator," *IEEE Trans. Ind. Electron.*, vol. 40, no. 2, pp. 259–265, 1993.
- [73] S. Katsura, Y. Matsumoto, and K. Ohnishi, "Realization of 'law of action and reaction' by multilateral control," *IEEE Trans. Ind. Electron.*, vol. 52, no. 5, pp. 1196–1205, 2005.
- [74] T. Shimono, S. Member, S. Katsura, and K. Ohnishi, "Abstraction and reproduction of force sensation from real environment by bilateral control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 907–918, 2007.
- [75] S. Tungpataratanawong, "Force sensorless workspace impedance control considering resonant vibration of industrial robot," *Proc. IEEE Annual Conf. Ind. Electron.*, pp. 1878–1883, 2005.
- [76] K. Ohnishi, "Online identification and compensation of the force estimating error in sensor-less bilateral control system," *Proc. IEEE 15th Conf. Emerg. Technol. Fact. Autom.*, pp. 1–4, 2010.
- [77] T. Bhattacharjee, H. Il Son, and D. Y. Lee, "Haptic control with environment force estimation for telesurgery.," *Proc. IEEE Conf. Eng. Med. Biol. Soc.*, pp. 3241–3244, 2008.

- [78] J. W. Jeong, P. H. Chang, and K. Bin Park, "Sensorless and modelless estimation of external force using time delay estimation: application to impedance control," *J. Mech. Sci. Technol.*, vol. 25, no. 8, pp. 2051–2059, 2011.
- [79] S. H. Kang, M. Jin, and P. H. Chang, "A solution to the accuracy/robustness dilemma in impedance control," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 3, pp. 282–294, 2009.
- [80] K. Youcef-Toumi and S. Wu, "Input/output linearization using time delay control," *J. Dyn. Sys., Meas., Control*, vol. 114, no. 1, pp. 10–19, 1992.
- [81] C. Mitsantisuk, K. Ohishi, S. Member, and S. Katsura, "Estimation of action/reaction forces for the bilateral control using Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4383–4393, 2012.
- [82] G. Liu, "On velocity estimation using position measurements," *Proc. Am. Control Conf.*, vol. 2, pp. 1115–1120, 2002.
- [83] J. P. J. Kolhe, M. Shaheed, T. S. Chandar, and S. E. Talole, "Robust control of robot manipulators based on uncertainty and disturbance estimation," *Int. J. Robust Nonlinear Control*, vol. 23, no. 1, pp. 104–122, 2013.
- [84] L. Hsu and F. Lizarralde, "Variable structure adaptive tracking control of robot manipulators without joint velocity measurement," *Proc. IFAC World Congr.*, no. 12, pp. 2–5, 1993.
- [85] R. J. E. Merry, M. J. G. van de Molengraft, and M. Steinbuch, "Velocity and acceleration estimation for optical incremental encoders," *Mechatronics*, vol. 20, no. 1, pp. 20–26, 2010.
- [86] M. Namvar, "A class of globally convergent velocity observers for robotic manipulators," *IEEE Trans. Automat. Contr.*, vol. 54, no. 8, pp. 1956–1961, 2009.
- [87] Y. Zhang, "A comparative study of Luenberger observer, sliding mode observer and extended Kalman filter for sensorless vector control of induction motor drives," *Proc. IEEE Energy Convers. Congr. Expo.*, pp. 2466–2473, 2009.
- [88] B. Xian, M. S. de Queiroz, D. M. Dawson, and M. L. McIntyre, "A discontinuous output feedback controller and velocity observer for nonlinear mechanical systems," *Automatica*, vol. 40, no. 4, pp. 695–700, 2004.
- [89] F. Celani, "A Luenberger-style observer for robot manipulators with position measurements," *Proc. 14th Mediterr. Conf. Control Autom.*, pp. 1–6, 2006.
- [90] S. Salcudean, "A globally convergent angular velocity observer for rigid body motion," *IEEE Trans. Automat. Contr.*, vol. 36, no. 12, pp. 1493–1497, 1991.

- [91] L. Chen, M. Laffranchi, N. G. Tsagarakis, and D. G. Caldwell, "A novel curve fitting based discrete velocity estimator for high performance motion control," *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, pp. 1060–1065, 2012.
- [92] Y. Su, P. Muller, and C. Zheng, "A simple nonlinear observer for a class of uncertain mechanical systems," *IEEE Trans. Automat. Contr.*, vol. 52, no. 7, pp. 1340–1345, 2007.
- [93] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: an historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.
- [94] P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," *Rob. Auton. Syst.*, vol. 38, no. 1, pp. 49–64, 2002.
- [95] X. Wang and P. Liu, "Improvement of haptic feedback fidelity for telesurgical applications," *Electron. Lett.*, vol. 42, no. 6, 2006.
- [96] S. Wang, J. Ding, J. Yun, and Q. Li, "A robotic system with force feedback for micro-surgery," *Proc. IEEE Int. Conf. Robot. Autom.*, pp.199–204, 2005.
- [97] H. Il Son, T. Bhattacharjee, and H. Hashimoto, "Effect of impedance-shaping on perception of soft tissues in macro-micro teleoperation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3273–3285, 2011.
- [98] M. Mitsuishi, N. Sugita, and P. Pitakwatchara, "Force-feedback augmentation modes in the laparoscopic minimally invasive telesurgical system," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 4, pp. 447–454, 2007.
- [99] H. Il Son, T. Bhattacharjee, and H. Hashimoto, "Effect of scaling on the performance and stability of teleoperation systems interacting with soft environments," *Adv. Robot.*, vol. 25, no. 11–12, pp. 1577–1601, 2011.
- [100] G. De Gersem, "Reliable and enhanced stiffness perception in soft-tissue telemanipulation," *Int. J. Rob. Res.*, vol. 24, no. 10, pp. 805–822, 2005.
- [101] M. C. Cavusoglu, A. Sherman, and F. Tendick, "Design of bilateral teleoperation controllers for haptic exploration and telemanipulation of soft environments," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 641–647, 2002.
- [102] X. Wang, P. Liu, and D. Wang, "Design of bilateral teleoperators for soft environments with adaptive environmental impedance estimation," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1127–1132, 2005.
- [103] S. Salcudean, K. Hashtrudi-Zaad, S. Tafazoli, S.P. DiMaio, and C. Reboulet, "Bilateral matched impedance teleoperation with application to excavator control," *IEEE Control Syst. Mag.*, vol. 19, no. 6, pp.29 -37, 1999.

- [104] H. Cho and J. Park, "Design and stability analysis of impedance controller for bilateral teleoperation under a time delay," *KSME Int. J.*, vol. 18, no. 7, pp. 1131–1139, 2004.
- [105] H. Cho and J. Park, "Impedance control with variable damping for bilateral teleoperation under time delay," *JSME Int. J. Ser. C*, vol. 48, no. 4, pp. 695–703, 2005.
- [106] A. Haddadi and K. Hashtrudi-Zaad, "Bounded-impedance absolute stability of bilateral teleoperation control systems," *IEEE Trans. Haptics*, vol. 3, no. 1, pp. 15–27, 2010.
- [107] P. Corke, "A robotics toolbox for MATLAB," *IEEE Robot. Autom. Mag.*, vol. 3, no. 1, pp. 24–32, 1996.
- [108] C. Presse and M. Gautier, "New criteria of exciting trajectories for robot identification," *Proc. IEEE Conf. Robot. and Autom.*, vol. 3, pp. 907–912, 1993.
- [109] J. Daly and D. Wang, "Time-delayed output feedback bilateral teleoperation with force estimation for-DOF nonlinear manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 1, pp. 299–306, 2014.
- [110] J. Cho and D. Lee, "Gain-scheduling control of a teleoperation system," *Proc. Conf. Syst. Man Cybern.*, pp. 1489–1495, 2009.
- [111] E. Vander Poorten, "Robust variable-scale bilateral control for micro teleoperation," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 655–662, 2008.
- [112] J. Cho, H. Son, D. Lee, T. Bhattacharjee, and D. Lee, "Gain-scheduling control of teleoperation systems interacting with soft tissues," *IEEE Trans. Ind. Electron.*, vol. 60, no. 3, pp. 946–957, 2013.
- [113] J. M. Daly and D. W. L. Wang, "Output feedback sliding mode control in the presence of unknown disturbances," *Syst. Control Lett.*, vol. 58, no. 3, pp. 188–193, 2009.
- [114] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.
- [115] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled H_∞ controllers," *IEEE Trans. Automat. Contr.*, vol. 40, no. 5, pp. 853–864, 1995.
- [116] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Trans. Rob. Auton.*, vol. 5, no. 4, pp. 426–434, 1989.

- [117] H. Kazerooni, T.-I. Tsay, and K. Hollerbach, "A controller design framework for telerobotic systems," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 1, pp. 50–62, 1993.