CONTROL OF REDUNDANT ROBOT MANIPULATORS WITH TELEROBOTIC APPLICATIONS

A Dissertation Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Electronics and Communication Engineering

by
Kamil ÇETİN

November 2016
İZMİR

We approve the thesis of **Kamil ÇETİN**

Examining Committee Members:

_____

Prof. Dr. Aydoğan SAVRAN
Department of Electrical and Electronics Engineering
Ege University

_____

Prof. Dr. Musa ALCI
Department of Electrical and Electronics Engineering
Ege University

_____

Assoc. Prof. Dr. Enver TATLICIOĞLU
Department of Electrical and Electronics Engineering
İzmir Institute of Technology

_____

Assoc. Prof. Dr. Mustafa A. ALTINKAYA
Department of Electrical and Electronics Engineering
İzmir Institute of Technology

_____

Asst. Prof. Dr. M. İ. Can DEDE
Department of Mechanical Engineering
İzmir Institute of Technology

11 November 2016

_____

Assoc. Prof. Dr. Enver TATLICIOĞLU
Supervisor
Department of Electrical and Electronics Engineering
İzmir Institute of Technology

_____

Prof. Dr. M. Salih DİNLEYİCİ
Head of the Department of
Electrical and Electronics Engineering

_____

Prof. Dr. Bilge KARAÇALI
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

I would like to express special thanks to my supervisor Assoc. Prof. Dr. Enver Tatlıcıoğlu for his advice, motivation and support throughout my Ph.D. studies at IYTE. I would also like to thank my committee members Prof. Dr. Aydoğan Savran, Prof. Dr. Musa Alcı, Assoc. Prof. Dr. Mustafa A. Altınkaya and Asst. Prof. Dr. M. İ. Can Dede for their feedback and technical support throughout my Ph.D. thesis.

I would like to thank all my colleagues Merve Doğan, Meryem Deniz and Barış Bıdıklı for their friendship, motivation and support. I would also like to thank my colleagues Emre Uzunoğlu and Osman N. Şahin at Iztech Laboratory for their help on experimental studies. In addition, I would also like to thank Prof. Dr. Erkan Zergeroğlu and Asst. Prof. Dr. Alper Bayrak for their technical support and contributions to publications during my Ph.D. studies.

I gratefully acknowledge the The Scientific and Technological Research Council of Turkey which supported my Ph.D. thesis via grant number 113E147.

I am deeply grateful to my parents who gave me trust and love during my life. I am immensely grateful to my wife Dr. Bilge Kartal Çetin for her unconditional love, motivation and endless patience, which without I would not be able to complete my Ph.D. thesis. I dedicate this thesis to my wife and my little daughter Mavisu Çetin.

ABSTRACT

CONTROL OF REDUNDANT ROBOT MANIPULATORS WITH TELEROBOTIC APPLICATIONS

This thesis focuses on task-space control of kinematically redundant robot manipulators with telerobotic applications. The first aim is to design asymptotically stable sub-task controllers for kinematically redundant robot manipulators subject to parametric uncertainties in their dynamics. Initially, a novel combined analysis of the task-space tracking and sub-task controllers is performed for redundant robots having only one extra degree of freedom. Next, an extended task-space controller is designed by integrating manipulator Jacobian with the sub-task Jacobian. Both controllers ensure task-space tracking and sub-task objectives at the amount of redundant degree of freedom.

As the second aim, two robust control methods are proposed for task-space tracking of robot manipulators. First, a novel continuous robust controller is designed despite dynamic model and Jacobian uncertainties to ensure asymptotic task-space tracking while requiring measurements of joint positions and velocities. Then, a robust output feedback controller is proposed to ensure ultimately bounded task-space tracking requiring neither measurements of joint positions or velocities nor accurate knowledge of kinematic and dynamic models.

The third aim is to develop a passive decomposition method for task-space control of bilateral teleoperation systems. The proposed method ensures coordination of master and slave robots while achieving a desired overall motion for the bilateral teleoperation system. The proposed method is firstly considered for teleoperation systems consisting of kinematically similar master and slave robots, then extended to be applicable to kinematically redundant teleoperation systems.

Simulation and experimental studies are performed to present the viability of the proposed methods.

ÖZET

ARTIK EKLEMLİ ROBOT KOLLARININ KONTROLÜ VE TELEROBOTİK UYGULAMALARI

Bu tez çalışması kinematik olarak artık eklemli robot kollarının görev uzayında denetlenmesi ve telerobotik uygulamaları üzerine odaklanmaktadır. İlk amaç kinematik olarak artık eklemli robot kollarının dinamik modellerindeki parametrik belirsizliklerine karşı asimptotik kararlı ikincil görev denetleyicileri tasarlamaktır. Öncelikle sadece bir artık serbestlik derecesine sahip robot kollarının görev uzayında takip ve ikincil görev denetleyicilerinin yeni bir birleştirilmiş analizi gerçekleştirilmektedir. Daha sonra ise robot Jakobiyan'ına ikincil görev Jakobiyan'ı ekleyerek uzatılmış bir görev uzayı denetleyicisi tasarlanmaktadır. Her iki denetleyici de görev uzayı takibini ve artık eklem sayısına kadar ikincil görev hedeflerini sağlamaktadır.

Bu tezin ikinci amacında robot kollarının görev uzayında denetlenmesi için iki farklı gürbüz denetleyici yöntemi önerilir. Öncelikle eklem pozisyon ve hız bilgilerinin ölçülebildiği durumda dinamik modeldeki ve Jakobiyan'daki belirsizliklere karşı görev uzayı takibinde asimptotik kararlılık sağlayan yeni bir sürekli gürbüz denetleyici tasarlanmaktadır. Daha sonra ise ne eklem pozisyon ve hız bilgilerinin ölçülebildiği ne de dinamik ve kinematik modelin tam olarak bilindiği durumda görev uzayı takibinde sınırlı kararlılık sağlayan çıkış geri beslemeli bir gürbüz denetleyici önerilmektedir.

Bu tezin üçüncü amacı ise iki yönlü teleoperasyon sistemlerinin görev uzayında denetlenmesi için pasif ayrıştırma yöntemi geliştirmektir. Önerilen yöntem iki yönlü teleoperasyon sisteminin genel hareketini sağlarken ana ve bağımlı robotların koordinasyonu da sağlanmaktadır. Önerilen yöntem öncelikle kinematik olarak benzer ana ve bağımlı robotlara sahip teleoperasyon sistemleri için düşünülür ve daha sonra artık teleoperasyon sistemleri için geliştirilmektedir.

Önerilen tüm yöntemlerin uygulanabilirliğini sunmak için benzetim ve deneysel çalışmalar yapılmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES ix

LIST OF TABLES xi

LIST OF SYMBOLS xii

LIST OF ABBREVIATIONS xv

CHAPTER 1. INTRODUCTION 1
1.1. Joint–space Versus Task–space Control 1
1.2. Teleoperation Systems 3
1.3. Literature Review on Task–space Control of Robot Manipulators .. 3
1.4. Literature Review on Sub–task Control of Kinematically Redundant Robot Manipulators 6
1.4.1. Literature Review on Passive Decomposition Based Control of Bilateral Teleoperation Systems 6
1.5. Contributions, Comparisons and Advantages of the Proposed Methods 8
1.6. Organization of This Thesis 12

CHAPTER 2. MODELING OF ROBOT MANIPULATORS 14
2.1. Mathematical Models of Robot Manipulators 14
2.1.1. Kinematic Model 14
2.1.1.1. Kinematic Models of Kinematically Non–redundant Robot Manipulators 14
2.1.1.2. Kinematic Models of Kinematically Redundant Robot Manipulators 16
2.1.2. Dynamic Model 19
2.2. Experimental Testbeds 22
2.2.1. PHANToM Omni Haptic Device 22
2.2.2. 3 dof RRR Planar Robot Manipulator 25

CHAPTER 3. NULL–SPACE CONTROL OF KINEMATICALLY REDUNDANT ROBOT MANIPULATORS 28
3.1. Mathematical Models and Properties 28
3.2. Error System Formulation 28
3.3. Controller Design and Stability Analysis 29
3.3.1. Adaptive Controller Extension 32
3.4. Simulation Results 34
3.5. Conclusions 35

CHAPTER 4. PSEUDO–INVERSE FREE CONTROL OF KINEMATICALLY REDUNDANT ROBOT MANIPULATORS 41
4.1. Mathematical Models and Properties 41
4.2. Extended Task–space Formulation 41
4.3. Error System Formulation 42
4.4. Adaptive Controller Design and Stability Analysis 43
4.5. Experimental Studies 45
4.6. Conclusions 47

CHAPTER 5. TASK–SPACE TRACKING CONTROL OF ROBOT MANIPULATORS WITH UNCERTAIN KINEMATICS AND DYNAMICS 51
5.1. Mathematical Models and Properties 51
5.2. Error System Formulation 51
5.3. Control Design 53
5.4. Stability Analysis 54
5.5. Experimental Study 56
5.6. Conclusions 57

CHAPTER 6. ROBUST OUTPUT FEEDBACK APPROACH FOR TASK–SPACE TRACKING CONTROL OF ROBOT MANIPULATORS 61
6.1. Mathematical Models and Properties 61
6.2. Observer and Controller Design 62
6.3. Simulation Study 64
6.4. Experimental Study 64
6.5. Conclusions 65

CHAPTER 7. PASSIVE DECOMPOSITION: A TASK–SPACE CONTROL APPROACH 69
7.1. Mathematical Models of Master and Slave Robots 69
7.2. Problem Formulation 70
7.3. Control Design 71
7.4. Simulation Results 72
7.5. Conclusions 72

CHAPTER 8. PASSIVE DECOMPOSITION BASED TASK–SPACE CONTROL OF REDUNDANT TELEOPERATION SYSTEMS 75
8.1. Mathematical Models of Master and Slave Robots 75
8.2. Problem Formulation 75
8.3. Control Design 77
8.4. Experimental Results 78
8.5. Conclusions 80

CHAPTER 9. CONCLUSIONS 84
9.1. Future Works 86

REFERENCES 88

APPENDIX A. PROOFS OF BOUNDS 95

LIST OF FIGURES

Figure **Page**

Figure 1.1. Joint–space control method. 1
Figure 1.2. Task–space control method. 2
Figure 1.3. Flow diagram of a bilateral teleoperation system. 4
Figure 2.1. 2 dof RR planar robot manipulator. 16
Figure 2.2. PHANToM Omni haptic device 23
Figure 2.3. 3 dof planar robot manipulator. 26
Figure 3.1. Task–space tracking errors $\boldsymbol{e}(t)$ for EMKC. 36
Figure 3.2. Control input torques $\boldsymbol{\tau}(t)$ for EMKC. 37
Figure 3.3. Joint positions $\boldsymbol{\theta}(t)$ for EMKC. 37
Figure 3.4. Desired $\boldsymbol{x_d}(t)$ and actual $\boldsymbol{x}(t)$ task–space trajectories for EMKC. 38
Figure 3.5. Task–space tracking errors $\boldsymbol{e}(t)$ for AC. 38
Figure 3.6. Control input torques $\boldsymbol{\tau}(t)$ for AC. 39
Figure 3.7. Joint positions $\boldsymbol{\theta}(t)$ for AC. 39
Figure 3.8. Desired $\boldsymbol{x_d}(t)$ and actual $\boldsymbol{x}(t)$ task–space trajectories for AC. 40
Figure 3.9. Estimates of uncertain parameters $\hat{\boldsymbol{\phi}}(t)$ for AC. 40
Figure 4.1. Illustration of the laser/camera tracer sub–task. 46
Figure 4.2. Task–space position tracking error $\boldsymbol{e}(t)$. 47
Figure 4.3. Control input torques $\boldsymbol{\tau}(t)$. 48
Figure 4.4. Desired $\boldsymbol{x_d}(t)$ and actual $\boldsymbol{x}(t)$ task–space trajectories. 49
Figure 4.5. Sub–task function $\boldsymbol{y_s}(\boldsymbol{\theta})$. 49
Figure 4.6. Estimates of uncertain parameters $\hat{\boldsymbol{\phi}}(t)$. 50
Figure 5.1. Experiment of Robust FSFB control: Task–space tracking error $\boldsymbol{e}(t)$. ... 58
Figure 5.2. Experiment of Robust FSFB control: Desired and actual task–space trajectories. 59
Figure 5.3. Experiment of Robust FSFB control: Control input torques $\boldsymbol{\tau}(t)$. 59
Figure 5.4. Experiment of Robust FSFB control: Estimates of uncertain parameters $\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}$(t). 60
Figure 6.1. Simulation of Robust OFB: Desired and actual task–space trajectories. . 65
Figure 6.2. Simulation of Robust OFB: $\boldsymbol{e}(t)$ vs. $\hat{\boldsymbol{e}}(t)$. 66
Figure 6.3. Simulation of Robust OFB: Control input torques $\boldsymbol{\tau}(t)$. 67
Figure 6.4. Experiment of Robust OFB: Desired and actual task–space trajectories. 67

Figure 6.5. Experiment of Robust OFB: $\boldsymbol{e}(t)$ vs. $\hat{\boldsymbol{e}}(t)$. 68
Figure 6.6. Experiment of Robust OFB: Control input torques $\boldsymbol{\tau}(t)$. 68
Figure 7.1. Flow diagram of a passive decomposition system. 69
Figure 7.2. End–effector tracking of master and slave robots. 73
Figure 7.3. Positions of shape subsystem. 73
Figure 7.4. Control input torques of master and slave robots. 74
Figure 8.1. End–effector positions of master robot and virtual slave. 80
Figure 8.2. End–effector trajectories of master robot and virtual slave. 81
Figure 8.3. Positions of shape subsystem $\boldsymbol{x_E}(t)$. 82
Figure 8.4. Control input torques of master robot and virtual slave. 82
Figure 8.5. Joint positions of virtual slave. 83
Figure 8.6. Estimate of human force (top) and environmental force in (8.25) (bottom). 83

LIST OF TABLES

Table **Page**

Table 1.1. Comparison of the proposed task–space control methods. 12

LIST OF SYMBOLS

$\boldsymbol{\theta}(t) \in \mathbb{R}^n$ ..... Joint positions.
$\dot{\boldsymbol{\theta}}(t) \in \mathbb{R}^n$ ..... Joint velocities.
$\ddot{\boldsymbol{\theta}}(t) \in \mathbb{R}^n$ ..... Joint accelerations.
$\boldsymbol{x}(t) \in \mathbb{R}^m$ ..... End–effector positions.
$\dot{\boldsymbol{x}}(t) \in \mathbb{R}^m$ ..... End–effector velocities.
$\ddot{\boldsymbol{x}}(t) \in \mathbb{R}^m$ ..... End–effector accelerations.
$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ ..... Forward kinematics (where $n = m$).
$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ ..... Forward kinematics (where $n > m$).
$\bar{\boldsymbol{f}} : \mathbb{R}^n \to \mathbb{R}^n$ ..... Inverse kinematics.
$\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$ ..... Jacobian matrix (where $n = m$).
$\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{m \times n}$ ..... Jacobian matrix (where $n > m$).
$\boldsymbol{J}^+(\boldsymbol{\theta}) \in \mathbb{R}^{n \times m}$ ..... Pseudo–inverse of Jacobian (where $n > m$).
$\boldsymbol{I_n} \in \mathbb{R}^{n \times n}$ ..... $n \times n$ Identity matrix.
$\boldsymbol{0_{n \times n}} \in \mathbb{R}^{n \times n}$ ..... $n \times n$ Zero matrix.
$\boldsymbol{0_n} \in \mathbb{R}^n$ ..... Zero vector.
$\dot{\boldsymbol{\theta}}_{\boldsymbol{N}}(t) \in \mathbb{R}^n$ ..... Arbitrary joint velocity vector.
$\boldsymbol{W_j}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{n \times p}$ ..... Regression matrix.
$\boldsymbol{\phi_j} \in \mathbb{R}^p$ ..... Constant parameter vector.
$\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$ ..... Generalized mass matrix.
$\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{n \times n}$ ..... Centripetal–Coriolis effects.
$\boldsymbol{G}(\boldsymbol{\theta}) \in \mathbb{R}^n$ ..... Gravitational effects.
$\boldsymbol{F_d} \in \mathbb{R}^{n \times n}$ ..... Frictional effects.
$\boldsymbol{\tau_d}(t) \in \mathbb{R}^n$ ..... Bounded disturbance vector.
$\boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t) \in \mathbb{R}^n$ ..... Combination of $\boldsymbol{C}, \boldsymbol{G}, \boldsymbol{F_d}, \boldsymbol{\tau_d}$.
$\boldsymbol{\tau}(t) \in \mathbb{R}^n$ ..... Control input torque.
$m_1, m_2 \in \mathbb{R}$ ..... Known positive constants.
$\boldsymbol{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) \in \mathbb{R}^{n \times p}$ ..... Regression matrix.
$\boldsymbol{\phi} \in \mathbb{R}^p$ ..... Constant parameter vector.
$\zeta_{c1}, \zeta_g, \zeta_{fd}, \zeta_d \in \mathbb{R}$ ..... Known positive bounding constants.
$\boldsymbol{\tau_{f_1}}(t) \in \mathbb{R}^{n_1}$ ..... Human forces applied on the joints of the master robot.
$\boldsymbol{\tau_{f_2}}(t) \in \mathbb{R}^{n_2}$ ..... Environmental forces on the joints of the slave robot.
$\boldsymbol{e}(t)$ ..... Task–space tracking error.

$\boldsymbol{r}(t)$, $\boldsymbol{r_a}(t)$, $\boldsymbol{s}(t)$ ..... Auxiliary error–like terms.

$\boldsymbol{z}(t)$, $\bar{\boldsymbol{z}}(t)$ ..... Combined error terms.

$\hat{\boldsymbol{J}}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$ ..... Estimated Jacobian matrix.

$\widetilde{\boldsymbol{J}}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$ ..... Estimation error of Jacobian matrix.

$\boldsymbol{h}(t) \in \mathbb{R}^{n}$ ..... Null–space controller.

$y_s(\boldsymbol{\theta}) \in \mathbb{R}$ ..... Sub–task function.

$\boldsymbol{J_s}(\boldsymbol{\theta}) \in \mathbb{R}^{1 \times n}$ ..... Sub–task Jacobian.

$\boldsymbol{\alpha}, \boldsymbol{K}, \boldsymbol{K_r}, \boldsymbol{\beta} \in \mathbb{R}^{n \times n}$ ..... Control gain matrices.

k_s ..... Control gains.

V, V_1, V_2 ..... Non–negative Lyapunov functions.

$\lambda_{\min}(\cdot)$ ..... Minimum eigenvalue of a matrix.

$\lambda_{\max}(\cdot)$ ..... Maximum eigenvalue of a matrix.

$\hat{\boldsymbol{\phi}}(t) \in \mathbb{R}^{p}$ ..... Estimate of the uncertain parameter vector.

$\widetilde{\boldsymbol{\phi}}(t) \in \mathbb{R}^{p}$ ..... Parameter estimation error vector.

$\boldsymbol{\Gamma} \in \mathbb{R}^{p \times p}$ ..... Adaptation gain matrix.

$\boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{s}, t) \in \mathbb{R}^{n}$ ..... Auxiliary uncertain term.

$\boldsymbol{Q_d}(\boldsymbol{x_d}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \dddot{\boldsymbol{x}}_{\boldsymbol{d}}) \in \mathbb{R}^{n}$ ..... Desired form of uncertain term.

$\widetilde{\boldsymbol{Q}}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{s}, t) \in \mathbb{R}^{n}$ ..... An auxiliary term.

ρ ..... A non–negative non–decreasing bounding function.

$\boldsymbol{\Pi}(t) \in \mathbb{R}^{n}$ ..... An auxiliary term.

$\text{Sgn}(\cdot) \in \mathbb{R}^{n}$ ..... The vector signum function.

$\text{Sat}\,(\cdot) \in \mathbb{R}^{n}$ ..... The vector saturation function.

$\text{Tanh}\,(\cdot) \in \mathbb{R}^{n}$ ..... The vector hyperbolic tangent function.

$\boldsymbol{\mu} \in \mathbb{R}^{p}$ ..... An auxiliary term.

$P(t) \in \mathbb{R}$ ..... An auxiliary function.

$\eta(t)$, $\zeta_P \in \mathbb{R}$ ..... Auxiliary terms.

$\boldsymbol{S_M} \in \mathbb{R}^{n \times n}$ ..... A positive–definite and symmetric matrix.

$\boldsymbol{D} \in \mathbb{R}^{n \times n}$ ..... A diagonal matrix with entries ± 1

$\boldsymbol{U} \in \mathbb{R}^{n \times n}$ ..... A unity upper triangular matrix.

$\boldsymbol{\tau_{ff}}(t) \in \mathbb{R}^{n}$ ..... Feed–forward component of the control input.

$\hat{\boldsymbol{e}}(t) \in \mathbb{R}^{n}$ ..... Observer for the task–space tracking error.

$\hat{\boldsymbol{r}}(t) \in \mathbb{R}^{n}$ ..... Observer for the auxiliary error–like term.

$\epsilon, \varepsilon \in \mathbb{R}$ ..... Small positive constants.

$\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2} \in \mathbb{R}^{n \times n}$ ..... Observer gain matrices.

$\boldsymbol{S}$ ..... A decomposition matrix.

$\bar{\phi}(\boldsymbol{x_1}, \boldsymbol{x_2})$ . . . . . An auxiliary matrix of the decomposition matrix.
$\boldsymbol{x_1}$ . . . . . End–effector position of master robot.
$\boldsymbol{x_2}$ . . . . . End–effector position of slave robot.
$\boldsymbol{J_1}$ . . . . . Jacobian matrix of master robot.
$\boldsymbol{J_2}$ . . . . . Jacobian matrix of slave robot.
$\boldsymbol{\theta_1}$ . . . . . Joint positions of master robot.
$\boldsymbol{\theta_2}$ . . . . . Joint positions of slave robot.
$\boldsymbol{M_1}$ . . . . . Generalized mass matrix of master robot.
$\boldsymbol{M_2}$ . . . . . Generalized mass matrix of slave robot.
$\bar{\boldsymbol{M}}_1$ . . . . . Generalized mass matrix of master robot in task–space.
$\bar{\boldsymbol{M}}_2$ . . . . . Generalized mass matrix of slave robot in task–space.
$\boldsymbol{C_1}$ . . . . . Centripetal–Coriolis matrix of master robot.
$\boldsymbol{C_2}$ . . . . . Centripetal–Coriolis matrix of slave robot.
$\boldsymbol{\tau_1}$ . . . . . Control torque input of master robot.
$\boldsymbol{\tau_2}$ . . . . . Control torque input of slave robot.
$\boldsymbol{x_L}(t), \dot{\boldsymbol{x}}_{\boldsymbol{L}}(t), \ddot{\boldsymbol{x}}_{\boldsymbol{L}}(t)$. task–space position, velocity and acceleration of locked subsystem.
$\boldsymbol{x_E}(t), \dot{\boldsymbol{x}}_{\boldsymbol{E}}(t), \ddot{\boldsymbol{x}}_{\boldsymbol{E}}(t)$. task–space position, velocity and acceleration of shape subsystem.
$\bar{\boldsymbol{M}}_{\boldsymbol{L}}(t)$, $\bar{\boldsymbol{M}}_{\boldsymbol{E}}(t)$ . . . . . Inertia effects of locked and shape subsystems.
$\bar{\boldsymbol{C}}_{\boldsymbol{L}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{LE}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{EL}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{E}}(t)$ . . . . . Centripetal–Coriolis effects of locked and shape subsystems.
$\boldsymbol{\tau}_{\boldsymbol{f_L}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{f_E}}(t)$ . . . . . Force effects of locked and shape subsystems.
$\boldsymbol{\tau_L}(t)$, $\boldsymbol{\tau_E}(t)$ . . . . . Control torque inputs of locked and shape subsystems.
$\boldsymbol{K_v}$, $\boldsymbol{K_p}$, $\boldsymbol{K_{null}}$ . . . . . Control gain matrices.
$g(\cdot)$ . . . . . A scalar sub–task function.
$\nabla \boldsymbol{g}$ . . . . . The gradient of the scalar sub–task function.
$\ddot{\hat{\boldsymbol{\theta}}}_{\boldsymbol{1}}(t)$ . . . . . The estimated joint acceleration of master robot.
$\hat{\boldsymbol{\tau}}_{\boldsymbol{f.1}}(t)$ . . . . . Force observer for master robot.
$\boldsymbol{e_f}(t)$ . . . . . The observation error of joint velocity of master robot.
$\boldsymbol{K_f}$, $\boldsymbol{K_l}$ . . . . . Observer gain matrices.
$\boldsymbol{K_s}$, $\boldsymbol{K_d}$ . . . . . Spring and damper gain matrices.
$\boldsymbol{x}^*$, $\dot{\boldsymbol{x}}^*$ . . . . Position and velocity between the end–effector and the position of the virtual surface.

LIST OF ABBREVIATIONS

dof . . . . . Degree of freedom.
RRR . . . . . Revolute revolute revolute.
EMKC . . . . . Exact model knowledge control.
AC . . . . . Adaptive control.
FSFB . . . . . Full state feedback.
OFB . . . . . Output feedback.

CHAPTER 1

INTRODUCTION

According to the tracking control objective being aimed in joint level or end–effector level, the controllers are classified as joint–space control and task–space control.

1.1. Joint–space Versus Task–space Control

The control objective in joint–space is to develop a feedback controller such that joint positions $\boldsymbol{\theta}(t)$ of a robot manipulator tracks desired joint positions, denoted by $\boldsymbol{\theta_d}(t)$. As shown in Figure 1.1, firstly, joint–space tracking error is formulated by comparing the desired joint positions with joint positions. Then, the feedback controller determines the joint torques, denoted by $\boldsymbol{\tau}(t)$, necessary to move the joints of the robot manipulator along the desired joint trajectories Siciliano and Khatib (2008).

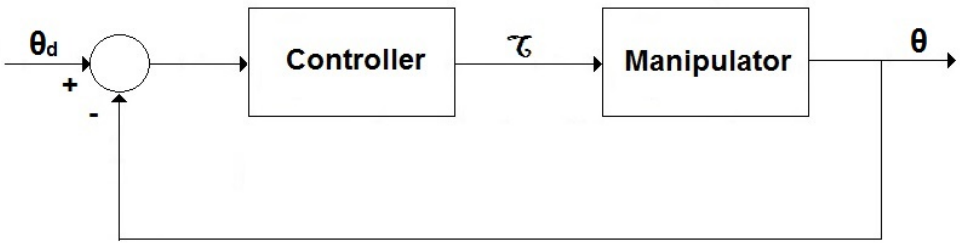

Figure 1.1. Joint–space control method.

If the desired motion of the robot manipulator is defined in task–space, the desired end–effector position, denoted by $\boldsymbol{x_d}(t)$, is converted to the desired joint position by utilizing the inverse kinematics, then after designing the joint–space controller, the joint positions are obtained from the robot manipulator. When the end–effector position cannot be measured, it can be calculated via using forward kinematics. This is called indirect task–space control method. However, this indirect method requires the calculation of inverse kinematics of the robot manipulator at the position level.

The control objective for many robotic applications is usually described in the task–space where the relationship between the robot and external objects is relevant. The

main goal of task–space control is to design a feedback controller that allows execution of an end–effector pose, denoted by $\boldsymbol{x}(t)$, that tracks the desired end–effector pose $\boldsymbol{x_d}(t)$.

Specifically, the direct task–space controller employs a feedback loop that directly minimizes task errors without requiring the solution of inverse kinematics at the position level, and aims to achieve the tracking of a given desired end–effector trajectory. The flow diagram of direct task–space controller is given in Figure 1.2. A good review on comparing task–space controllers from both theoretical and empirical perspectives can be found in Siciliano and Khatib (2008) and Nakanishi et al. (2008). While in the literature, the term task–space is alternatively named as operational–space, Cartesian space or end–effector space, in this thesis, the term task–space is preferred.

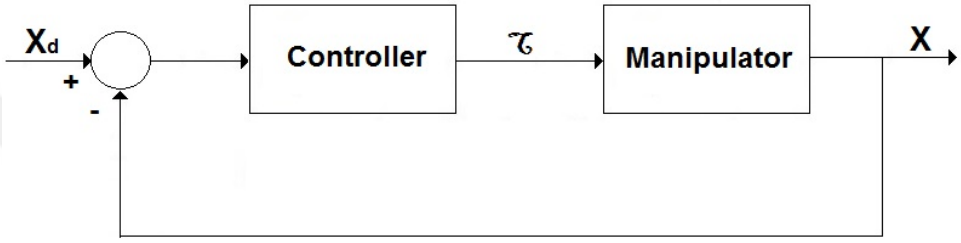

Figure 1.2. Task–space control method.

A robot manipulator is kinematically non–redundant when the dimension of the joint–space (i.e., n) is equal to the dimension of the task–space (i.e., m). When $n > m$, a robot manipulator is called kinematically redundant and there exists $n - m$ redundant degree of freedom (dof). In an alternative definition, a robot manipulator is called to be kinematically redundant when the number of dof is greater than the number of variables necessary to describe a given task. Robot manipulators with a much larger number of joints than the dimension of its task–space ($n \gg m$) are often called hyper redundant.

For kinematically redundant robot manipulators, the redundancy problem becomes challenging due to an infinite number of inverse kinematic solutions Hsu et al. (1989), Nakamura (1991), Nenchev (1989), Conkur and Buckingham (1997). While this may complicate the control design, after utilizing joint motion in the null–space of the Jacobian matrix, these redundant dof allow the robot manipulator to execute secondary sub–tasks (such as manipulability, joint limit avoidance, obstacle avoidance, limiting the impact force, minimizing potential energy).

Both joint–space and task–space controllers may be designed via utilizing linear controllers or nonlinear controllers Lewis et al. (1993), Lewis et al. (2003). In the literature, there are many nonlinear control methods that guarantee stability of the robot manip-

ulator and ensure joint–space or task–space tracking of a desired trajectory Dawson et al. (1995), Lewis et al. (2003), Dixon (2003), Khalil (2015). In order to exactly cancel the nonlinear effects, exact model knowledge controllers may be designed An et al. (1988). When there are structured/parametric uncertainties in the robot kinematic and/or dynamic models, adaptive controllers utilizing estimation methods can be developed Krstic et al. (1995), Ioannou and Sun (1996). In the presence of unstructured uncertainties associated with the robot dynamics, model free robust controllers can be designed Abdallah et al. (1991), Qu and Dawson (1995), Sage et al. (1999). These nonlinear control methods can be classified as full state feedback control and output feedback control depending on the availability of joint position and velocity measurements. Full state feedback controllers require measurements of both joint positions and joint velocities. On the other hand, output feedback controllers are designed under the restriction that only joint position measurements being available.

1.2. Teleoperation Systems

In a teleoperation system, a master robot handled by a human operator in a local station sends commands to a slave robot in a distance station and receives feedback from the remote environment. The teleoperation system allows the human operator to perform particular tasks in the remote environment where it is unreachable, expensive, dangerous or unfeasible to execute the tasks by a human operator. There are many important application areas of teleoperation systems such as space technologies, underwater explorations, telesurgery, handling of nuclear or hazardous materials, military operations or mine searches Sheridan (1992).

According to the choice of transmitted inputs, teleoperation systems can be classified as unilateral or bilateral. In unilateral teleoperation systems, there is only one way of information transmission between master and slave systems and no force feedback is available from the slave system. As shown in Figure 1.3, while the slave robot is performing the task commanded by the human operator in the remote environment, if the contact force information is reflected back to the operator (via visual, audio, kinesthetic or haptic feedback) through the master robot, this is named as bilateral teleoperation Hokayem and Spong (2006).

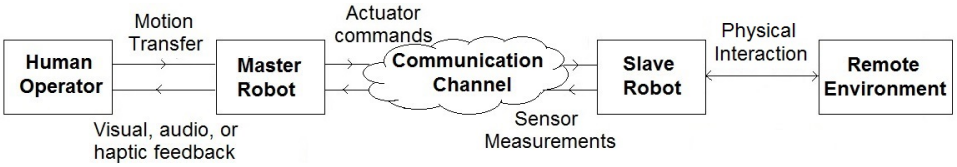

Figure 1.3. Flow diagram of a bilateral teleoperation system.

1.3. Literature Review on Task–space Control of Robot Manipulators

Some of the earlier research have relied on accurate knowledge of robot kinematic and dynamic models. For task–space tracking control of robot manipulators, Khatib (1983) proposed a control method based on the dynamic model of a manipulator in task–space. Assuming that the exact knowledge of the robot dynamics is known Hsu et al. (1989) presented a dynamical feedback linearizing controller that ensures task–space tracking and enables the self motion of the manipulator flow along the projection of a given arbitrary vector field. Zergeroglu et al. (2000) and Zergeroglu et al. (2004) designed an exact model knowledge controller that guarantees exponential task–space tracking. Xian et al. (2004) developed a quaternion based exact model knowledge controller that achieved asymptotic task–space tracking.

From a theoretical point of view, when robot kinematics or dynamics are subject to structured uncertainties, adaptive control techniques can be utilized. In view of this, several works addressed adaptive task–space tracking control to deal with kinematic uncertainties. Cheah et al. (1999) proposed a proportional integral derivative controller for robot manipulators having an uncertain Jacobian matrix. Cheah et al. (2000) proposed an approximate Jacobian control method for set point control of robots with kinematic uncertainties. Cheah (2003) proposed an adaptive law to estimate the uncertain kinematic model parameters for the approximate Jacobian method of Cheah et al. (2000). Without requiring the task–space velocity and the inverse of the approximate Jacobian matrix, Cheah et al. (2003) and Cheah et al. (2004) proposed an approximate Jacobian controller for robot manipulators having uncertainties in kinematics and in Jacobian. Cheah (2008) presented approximate transpose Jacobian and inverse Jacobian methods for set point control of non–redundant robots with parametric uncertainties in kinematics. Adaptive controller formulations to deal with dynamical uncertainties were also presented in sev-

eral works. Luo and Ahmad (1997) developed an adaptive control method for redundant robots by estimating uncertain parameters of dynamic model. Yazarel and Cheah (2002) proposed an adaptive task–space controller for set point control of robots with uncertainties in kinematics and in gravitational terms. Zergeroglu et al. (2000) and Zergeroglu et al. (2004) designed an adaptive controller that achieves asymptotic task–space tracking despite parametric uncertainties associated with the dynamic model. Tee and Yan (2011) presented an adaptive task–space controller for redundant robots by considering time–varying uncertainties and without knowledge of their bounds. Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a) developed a quaternion based adaptive full–state feedback controller for redundant robot manipulators with parametric uncertainties in their dynamic model. Tatlicioglu et al. (2008)(b) and Tatlicioglu et al. (2009) designed an adaptive feedback linearizing controller to compensate for the parametric uncertainties in dynamics.

While different research problems were solved in the aforementioned past works, nearly all of them required the parametric uncertainties associated with the robot dynamics to meet the linear in parameter property. To deal with unstructured uncertainties, robust control techniques can be utilized. In robust control literature Dawson et al. (1993), Qu and Dawson (1995), Sage et al. (1999), Ozbay et al. (2008), two methods are common, namely, variable structure type controllers and high gain controllers. Variable structure controllers usually require the use of switching functions, like the signum function, in their design, making most of them to be discontinuous. On the other hand, high gain controllers cannot ensure asymptotic tracking. A robust adaptive controller for kinematically redundant robot manipulators was presented in Colbaugh and Glass (1995). Using extended task–space formulation, a compliant motion controller was proposed by Peng and Adachi (1993). Zergeroglu et al. (2006) and Sahin et al. (2006) developed a robust controller that achieves uniformly ultimately bounded end–effector and sub–task tracking despite the parametric uncertainties associated with the dynamics and additive external disturbances. Braganza et al. (2006) and Kapadia et al. (2008) developed robust task–space tracking controllers for kinematically redundant robot manipulators in the presence of unstructured uncertainties while achieving different sub–task objectives. As a subclass of robust controllers, learning controllers also deal with unstructured uncertainties. For robot manipulators with periodic desired end–effector trajectory, Dogan et al. (2015) developed a task–space learning controller that ensures asymptotic task–space tracking by learning the uncertainties associated with the robot dynamics.

While full state feedback controllers require joint position and joint velocity measurements, output feedback controllers do not require joint velocity measurements. For

task–space tracking control of redundant robot manipulators, Zergeroglu et al. (2000) and Zergeroglu et al. (2004) designed an exact model knowledge output feedback controller that eliminates joint velocity measurements via a model based joint velocity observer. Xian et al. (2004) proposed a quaternion–based output feedback controller in conjunction with a joint velocity observer for task–space tracking control of non–redundant robot manipulators.

1.4. Literature Review on Sub–task Control of Kinematically Redundant Robot Manipulators

The research on kinematically redundant robot manipulators has been active for quite a long time. This is mostly due to the fact that the design and implementation of some of the relatively simple problems for conventional robotic manipulators, such as path planning or dynamic control, might become quite hard when kinematic redundancy has to be taken into account. For redundancy resolution in task–space control, in the literature Siciliano and Khatib (2008), there are two main methods as pseudo–inverse Jacobian and extended Jacobian methods. Khatib (1983) proposed an exact model knowledge control method for redundant robot manipulators by using pseudo–inverse of the Jacobian matrix. A configuration control approach in which the end–effector motion is augmented with sub–tasks was proposed in Baillieul (1985) and Seraji (1989).

Since there are an infinite number of joint configurations for any given end–effector pose of redundant robot manipulator Walker (1994), among all the possible configurations some can be preferred Nenchev (1989) while performing a desired task. One way of obtaining a preferred configuration at joint level is through the use of secondary functions. Examples to this are presented in Yoshikawa (1994) for maximizing the manipulability, in Seraji (1991) for minimizing joint velocities and joint accelerations, in Walker (1994) for reducing impact forces, in Kapadia et al. (2008) for obstacle avoidance in unstructured environment, in Braganza et al. (2006) for grasping, in Tatlicioglu et al. (2009) for joint limit avoidance, in Tatlicioglu et al. (2008)(a) for minimizing potential energy. The secondary tasks or sub–tasks are usually utilized as an add–on to the main task–space control objective Xian et al. (2004)(a), Nakanishi et al. (2008). Recently, there has been some work on the use of secondary task formulation for kinematically redundant robot manipulators that obtained practical sub–task tracking as in Tatlicioglu et al. (2005), Tatlicioglu et al. (2008)(a), Tatlicioglu et al. (2008)(b), and Tatlicioglu et al. (2009).

1.4.1. Literature Review on Passive Decomposition Based Control of Bilateral Teleoperation Systems

As detailed in the literature survey of Hokayem and Spong (2006), during the last six decades, an excessive amount of research has been carried out on bilateral teleoperation. According to the literature, many control methods have been developed that focus on stability, transparency and time delay in bilateral teleoperation systems. In addition, passivity of the teleoperator system or in other words, safety of both the human operator and the remote environment is also a critical issue in several teleoperation applications that requires contact (such as remote robotic surgery). In order to ensure safe interaction of human operator with master robot and slave robot with remote environment, controllers that target coordination of master and slave robots while ensuring passivity of the overall teleoperation system are designed. Several passivity based control methods have been developed for bilateral teleoperation Hashtrudi-Zaad and Salcudean (2001), Nuno et al. (2011), Li and Lee (2000), Lee and Li (2002a), Lee and Li (2002b), Lee and Li (2003), Lee and Li (2007). Among these past works, passive decomposition approach proposed by Li and Lee (2000), Lee and Li (2002a), Lee and Li (2002b), Lee and Li (2003), Lee and Li (2007) achieved both passivity and stability of the overall closed-loop teleoperation system. In general, passive decomposition approach achieves energetic passivity by decomposing the bilateral teleoperation system into two subsystems namely as shape and locked subsystems. The shape subsystem quantifies the coordination between master and slave robots, while the locked subsystem determines the overall motion of the closed-loop teleoperation system. Li and Lee (2000) proposed a passive feedforward approach for linear dynamically similar teleoperation systems. Later, Lee and Li (2002a), Lee and Li (2002b), and Lee and Li (2003) extended the proposed approach for dynamically dissimilar nonlinear teleoperated manipulators with power scaling. However, all of these passive decomposition methods were applied to kinematically similar teleoperated manipulators for joint–space control.

1.5. Contributions, Comparisons and Advantages of the Proposed Methods

The first aim of this thesis is to design a task–space controller fused with a novel sub–task controller that achieves asymptotic task–space tracking with asymptotic sub–task tracking, as opposed to the asymptotic end–effector tracking with practical sub–task tracking results presented in the literature. Zergeroglu et al. (2000) and Zergeroglu et al. (2004) designed an adaptive controller that achieved asymptotic task–space tracking without integrating the sub–task objective into the stability analysis. Later Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a) addressed this issue by designing a quaternion based adaptive controller that achieved asymptotic tracking in the task–space with systematic integration of the sub–task objective into the stability analysis. Different from the previously proposed controllers of Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a), the proposed methodology includes a new term that depends on the sub–task function and its partial derivative with respect to the joint positions. This new term allowed us to conduct a combined stability analysis for both task–space tracking and sub–task objective. A novel Lyapunov function including square of the scalar sub–task function is then introduced. Exponential stability of the exact model knowledge and asymptotic stability of the adaptive controller that compensates for parametric uncertainties in the robot dynamics are then ensured via Lyapunov type arguments. When compared to the previous works of Zergeroglu et al. (2000), Zergeroglu et al. (2004), Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a), the null–space component of both controllers achieves asymptotic sub–task tracking in addition to asymptotic task–space tracking.

Although the first aim of this thesis was accomplished, only one sub–task function can be achieved due to the proposed methodology being applicable to robots with one redundant dof (*i.e.,* $n - m = 1$). The second motivation arises from the need for applying the same method to redundant robots with more than one redundant dof (*i.e.,* $n - m > 1$). As another method, an extended task–space formulation is designed by integrating the manipulator Jacobian matrix with the sub–task Jacobian matrix. Additional sub–task functions, that depend on the joint positions and have equality constraints, are properly chosen up to the amount of the redundant dof. For the task–space tracking objective, an adaptive controller is designed in order to compensate for parametric uncertainties in the robot dynamics. Asymptotic stability of the designed controller is then ensured via Lyapunov type arguments. This method presents a major improvement to the literature in the sense that multiple sub–task objectives can be performed simultaneously along with

the main task–space objective. In addition, the performance of the proposed extended task–space formulation is experimentally validated using a planar 3 dof redundant robot manipulator. For the sub–task objective in the experiments, inspired from Buckingham and Graham (2005), a novel sub–task function is developed to ensure that a fixed laser or optic camera on the first joint accurately tracks the end–effector of the manipulator.

Proposed adaptive controllers compensate for only parametric uncertainties in robot dynamics. Next, dealing with unstructured uncertainties associated with kinematic and/or dynamic models is aimed. Two different control methods are proposed for robust task–space tracking control of robot manipulators. In the first method, in a novel departure from the existing results in the literature, task–space tracking control problem of robot manipulators under the additional restrictive constraints of dynamic model being subject to unstructured uncertainties and parametric kinematic uncertainties is tackled via a continuous robust controller scheme. Specifically, a continuous robust controller is designed to ensure asymptotic task–space tracking despite the presence of unstructured uncertainties associated with the dynamical terms while also dealing with parametric kinematic uncertainties. Integral of signum of an auxiliary error term is utilized in the design of the control input to ensure continuous actuation. The control design is based on Lyapunov based synthesis and analysis techniques, and asymptotic stability of the task–space tracking error is guaranteed. Experiments conducted on PHANToM Omni Haptic device are presented to demonstrate the feasibility and the performance of the robust controller.

While the above robust controller ensures asymptotic task–space tracking while dealing with unstructured uncertainties, it requires both joint position and velocity measurements. Next, a robust observer based output feedback controller is proposed to deal with unstructured uncertainties in both kinematic and dynamic models. Robust output feedback approach inspired by the works Seshagiri and Khalil (2005) and Chen et al. (2008) is adapted to task–space tracking control of robot manipulators. Under mild assumptions on both kinematics and dynamics of the robot manipulator, uniformly, ultimately bounded task–space tracking is ensured without requiring neither measurements of joint positions or velocities nor accurate knowledge of kinematic and dynamic models. Numerical simulations and experimental studies are performed to present the viability of the proposed controller.

A close review of the literature on passive decomposition based control of bilateral teleoperation systems highlighted the fact that they were restricted to joint–space control of teleoperation systems consisting of kinematically similar master and slave robots. To address this open research problem, firstly, a passive decomposition controller is designed

for task–space control of a $2n$ dof teleoperation system consisting of kinematically similar n dof master and slave robots. The error system is formed as the difference between the end–effector poses of master and slave robots while a desired trajectory is to be tracked. The proposed approach is based on availability of exact model knowledge of kinematic and dynamic models, and full-state feedback with force sensing from master and slave robots. A simulation study is performed to illustrate the performance of the proposed controller. In the simulation study, both human force on master robot and environmental force acting on slave robot are virtually modeled.

Since the previous study of the passive decomposition based task–space control considers only kinematically similar teleoperated manipulators, next, a non–redundant master robot and a kinematically redundant slave robot are considered to be the subsystems of the bilateral teleoperation system. So, the control problem is complicated by the slave robot having more dof than the master robot. The control problem is formed as end–effector tracking in accordance with the previous study on passive decomposition. Since the slave robot has redundant dof, these redundant dof can be utilized to meet some secondary objectives. Considering an $n_1 + n_2$ dof redundant teleoperation system consisting of an n_1 dof non-redundant master robot and an n_2 dof redundant slave robot (with $n_2 > n_1$), in task–space, the closed-loop teleoperation system is decomposed into two n_1 dof subsystems as shape and locked subsystems by utilizing a non-square decomposition matrix. A task–space controller is then designed. The controller ensures that the end–effector of the slave robot tracks the end–effector of the master robot while obtaining a desired overall motion for the closed-loop teleoperation system. The $n_2 - n_1$ redundant dof in the slave robot are made use of via the design of a null–space controller. Specifically, to solve the redundancy resolution, the pseudo–inverse Jacobian method in Siciliano (1990) is utilized. Experimental studies are conducted to illustrate the performance of the proposed control approach.

The main contributions of this thesis are:

- The **first** asymptotically stable task–space controller with asymptotically stable sub–task controller is designed. The analysis of two task–space controllers (one exact model knowledge and one adaptive) are presented for redundant robot manipulators which has only one redundant dof. The proposed combined stability analysis is **novel** when compared to the existing literature on control of redundant robot manipulators.

- For redundant robot manipulators having more than one redundant dof, an extended task–space formulation is designed by integrating the manipulator Jacobian matrix

with the sub–task Jacobian matrix. This method presents a major improvement to the relevant literature in the sense that multiple sub–task objectives can be performed simultaneously along with the main task–space tracking objective.

- A **new** robust task–space controller is designed that ensures asymptotic task–space tracking despite unstructured uncertainties in the robot dynamics and parametric uncertainties in the kinematic model on the velocity level. As opposed to most variable structure controller forms, the designed robust controller is continuous, and asymptotic tracking is ensured.

- A robust observer based output feedback controller is proposed without requiring measurements of joint positions and velocities and without the need of accurate knowledge of kinematics and dynamics of the robot manipulator. Only the end–effector position measurements are used thus a simple control structure is designed with minimum requirements. The proposed robust output feedback controller ensures uniformly, ultimately bounded task–space tracking while dealing with uncertainties in both kinematic and dynamic models. When compared to the previous robust task–space controllers in the literature, this is one of the few robust and output feedback task–space controllers that achieves practical task–space tracking.

- Passive decomposition approach is proposed for task–space control of bilateral teleoperation systems for the first time in the literature.

- The passive decomposition approach is extended for task–space control of bilateral teleoperation systems having kinematically redundant slave robot.

In Table 1.1, a comparison of the proposed task–space controllers in Chapters 3 - 6 is given. The comparison is based on the amount of dynamic model and Jacobian knowledge required by the controller and the availability of the state measurements needed to implement the controller. As can be seen from Table 1.1, the design of the controllers have a flow where lesser amount of model knowledge and sensory information are required.

The results in this thesis are presented in the following publications:

- K. Cetin, E. Tatlicioglu, and E. Zergeroglu, "On Null–Space Control of Kinematically Redundant Robot Manipulators", European Control Conference, pp. 678-683, Aalborg, Denmark, 2016.

- K. Cetin, E. Tatlicioglu, and E. Zergeroglu, "Pseudo–Inverse Free Control of Kinematically Redundant Robot Manipulators with Sub–task Control", Journal of Intelligent and Robotic Systems, 2016, under review.

Table 1.1. Comparison of the proposed task–space control methods.

Task–space Control Methods	**Required Model Knowledge**						**Required Measurements**	
	Dynamic Model			Jacobian				
	Full	Partial	None	Full	Partial	None	Position	Velocity
EMK FSFB	✓			✓			✓	✓
Adaptive FSFB		✓		✓			✓	✓
Robust FSFB			✓		✓		✓	✓
Robust OFB			✓			✓	✓	

- K. Cetin, E. Tatlicioglu, and E. Zergeroglu, "Continuous Robust Task–Space Tracking Control of Robotic Manipulators with Uncertain Dynamics", IEEE Multi - Conference on Systems and Control, pp. 312-317, Sydney, Australia, 2015.
- K. Cetin, E. Tatlicioglu, and E. Zergeroglu, "Task–Space Tracking Control of Robotic Manipulators with Uncertain Dynamic and Kinematic Terms: A Continuous Robust Approach", Robotica, 2016, under review.
- K. Cetin, E. Tatlicioglu, and M. Deniz, "Task–space Tracking Control of Robot Manipulators with Uncertain Dynamics and Kinematics: Robust Output Feedback Approach", Journal of Intelligent and Robotic Systems, 2016, under review.
- E. Tatlicioglu and K. Cetin, "Passive Decomposition: A Task–space Control Approach", TrC IFToMM Symposium on Theory of Machines and Mechanisms, pp. 615-618, Izmir, Turkey, 2015.
- K. Cetin and E. Tatlicioglu, "Passive Decomposition Approach for Redundant Teleoperation Systems in Task–Space Control", Journal of Intelligent and Robotic Systems, 2016, under review.

1.6. Organization of This Thesis

The rest of the thesis is organized as follows. In Chapter 2, an overview of mathematical models of robot manipulators is explained. Then, the kinematic and dynamic models of the robot manipulators utilized in the numerical studies are presented. In Chapter 3, two task–space tracking controllers (one exact model knowledge and one adaptive) with a novel sub–task controller are developed for redundant robot manipulators which have only one redundant dof. In Chapter 4, the sub–task control design in Chapter 3 is

advanced to redundant robot manipulators having more than one redundant dof where an adaptive task–space controller is designed in order to compensate for parametric uncertainties in the robot dynamics. In Chapter 5, a novel continuous robust controller formulation is developed for robot manipulators having unstructured uncertainties in the robot dynamics and parametric uncertainties in the Jacobian. In Chapter 6, a robust observer based output feedback approach is proposed for task–space tracking control of robot manipulators in the presence of unstructured uncertainties in both kinematic and dynamic models. In Chapter 7, passive decomposition method is developed for task–space control of bilateral teleoperation consisting kinematically similar non–redundant master and slave robots. In Chapter 8, the results in Chapter 7 are advanced to task–space control of kinematically redundant teleoperation systems consisting of a non–redundant master and a redundant slave robots. In Chapter 9, the proposed approaches in this thesis are summarized and compared to the literature. Finally, open problems and research directions related to the studies in this thesis are presented as possible future works.

CHAPTER 2

MODELING OF ROBOT MANIPULATORS

This chapter sets the stage for this thesis by providing an overview of mathematical models of robot manipulators. Then the specifications of the robot manipulators utilized in the numerical studies are presented by giving their kinematic and dynamic models.

2.1. Mathematical Models of Robot Manipulators

In this section, kinematic models of non–redundant and redundant robot manipulators are presented. Then, dynamic models of the robot manipulators are provided along with model properties.

2.1.1. Kinematic Model

Robot kinematics concern the relationship between the dimensions and connectivity of kinematic chains and the position, velocity and acceleration of each of the joints of the robot manipulator. The formulation of the kinematics relationship between joint positions and end–effector pose of the robot manipulator allows the study of the forward/direct kinematics and the inverse kinematics. Forward kinematics compute the pose of the end–effector using the specified positions of the joints; on the other hand, inverse kinematics at the position level compute the joint positions from the specified pose of the end–effector.

2.1.1.1. Kinematic Models of Kinematically Non–redundant Robot Manipulators

The end–effector pose of a non–redundant robot manipulator, denoted by $\boldsymbol{x}(t) \in \mathbb{R}^n$, is obtained as a function of the joint position vector $\boldsymbol{\theta}(t) \in \mathbb{R}^n$ as

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\theta}) \tag{2.1}$$

where $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the transformation representing the forward kinematics of the non–redundant robot manipulator. Differentiating (2.1) with respect to time yields

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{2.2}$$

where $\dot{\boldsymbol{x}}(t) \in \mathbb{R}^n$ is the task–space velocity vector and the Jacobian matrix of the non–redundant robot manipulator $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ is defined as

$$\boldsymbol{J}(\boldsymbol{\theta}) \triangleq \frac{\partial \boldsymbol{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{2.3}$$

The inverse kinematics of the non–redundant robot manipulator can be obtained as Dawson et al. (1995)

$$\boldsymbol{\theta} = \bar{\boldsymbol{f}}(\boldsymbol{x}) \tag{2.4}$$

where $\bar{\boldsymbol{f}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the inverse kinematics. The time derivative of (2.4) can be written in the form

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{-1}(\boldsymbol{x})\dot{\boldsymbol{x}} \tag{2.5}$$

where $\boldsymbol{J}^{-1}(\boldsymbol{x}) \triangleq \frac{\partial \bar{\boldsymbol{f}}}{\partial \boldsymbol{x}} \in \mathbb{R}^{n\times n}$ is the inverse Jacobian matrix. It is considered that by using (2.1) and (2.4), the task–space position $\boldsymbol{x}$ and the joint position $\boldsymbol{\theta}$ can be written in terms of each other then both $\boldsymbol{J}^{-1}(\boldsymbol{x})$ and $\boldsymbol{J}^{-1}(\boldsymbol{\theta})$ will be used interchangeably in the rest of this thesis.

In Figure 2.1, a 2 dof planar RR (revolute joint) robot manipulator is given as an example of a non–redundant robot manipulator. Using forward kinematics, its end–effector position in X and Y axes can be obtained as

$$\boldsymbol{x} = \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) \end{bmatrix} \tag{2.6}$$

where l_1 and l_2 are the first and second link lengths of the robot manipulator, respectively. Based on the above forward kinematics, the manipulator Jacobian is obtained as

$$\boldsymbol{J} = \begin{bmatrix} -l_1\sin(\theta_1) - l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \end{bmatrix}. \tag{2.7}$$

For the 2 dof robot manipulator in Figure 2.1, given the position of the end–effector, the joint positions can be calculated as follows

$$\theta_2 = \arctan(\sin\theta_2, \cos\theta_2) \tag{2.8}$$

$$\theta_1 = \arctan(\sin\theta_1, \cos\theta_1) \tag{2.9}$$

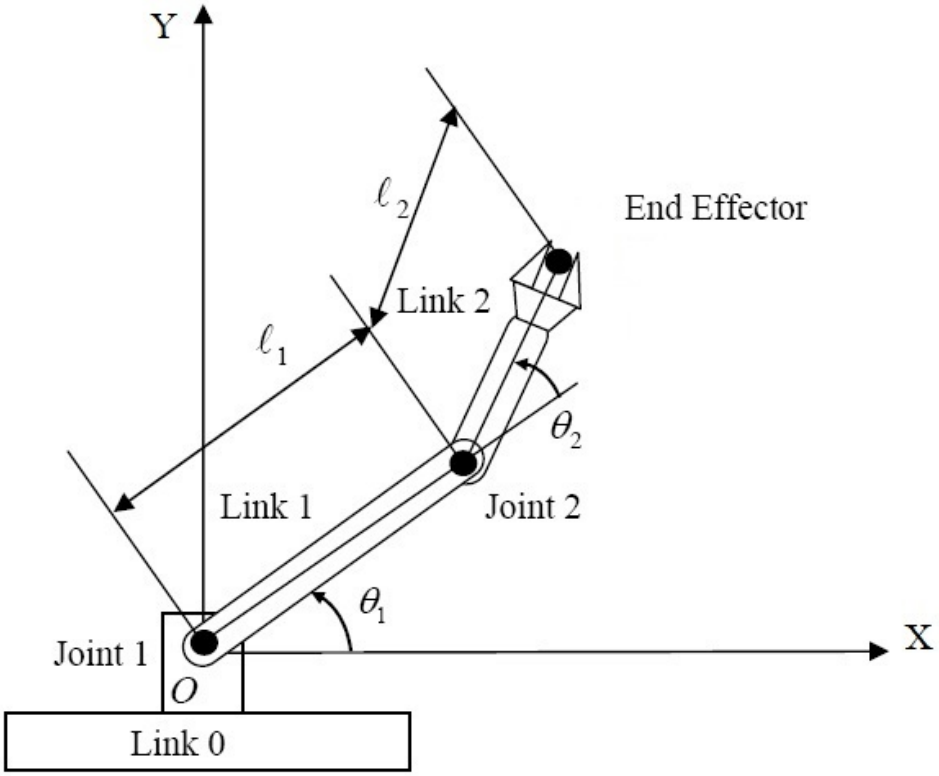

Figure 2.1. 2 dof RR planar robot manipulator.

where

$$\cos(\theta_2) = \frac{X^2 + Y^2 - l_1^2 - l_2^2}{2l_1 l_2} \tag{2.10}$$

$$\sin(\theta_2) = \pm\sqrt{(1 - \cos^2\theta_2)} \tag{2.11}$$

$$\cos(\theta_1) = \frac{Y - (l_1 + l_2\cos\theta_2)\sin\theta_1}{l_2\sin\theta_2} \tag{2.12}$$

$$\sin(\theta_1) = \frac{Y(l_1 + l_2\cos\theta_2) - X(l_2\sin\theta_2)}{(l_1 + l_2\cos\theta_2)^2 + (l_2\sin\theta_2)^2}. \tag{2.13}$$

2.1.1.2. Kinematic Models of Kinematically Redundant Robot Manipulators

The end-effector position of an n dof redundant robot manipulator, denoted by $\boldsymbol{x}(t) \in \mathbb{R}^m$, is obtained as follows

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\theta}) \tag{2.14}$$

where $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ denotes the forward kinematics where $n > m$. The first–order differential kinematic relationship between end-effector velocities and joint velocities can be obtained as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{2.15}$$

where Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{m \times n}$ is defined as

$$\boldsymbol{J}(\boldsymbol{\theta}) \triangleq \frac{\partial \boldsymbol{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{2.16}$$

While the first–order differential kinematics relates task–space velocities to joint–space velocities, the second–order differential kinematics provides a relationship between accelerations as follows

$$\ddot{\boldsymbol{x}} = \dot{\boldsymbol{J}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{J}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}. \tag{2.17}$$

The Jacobian, which is always a function of the configuration of the robot manipulator, describes the linear mapping from the joint velocity space to the task velocity space. The range space of $\boldsymbol{J}(\boldsymbol{\theta})$ is the sub–space $\boldsymbol{R}(\boldsymbol{J}) \in \mathbb{R}^m$ of end–effector velocities that can be generated by the joint velocities for the given robot manipulator configuration. The null space of $\boldsymbol{J}(\boldsymbol{\theta})$ is the sub–space $\boldsymbol{N}(\boldsymbol{J}) \in \mathbb{R}^n$ of joint velocities that do not produce any end–effector velocity for the given robot manipulator configuration. If the Jacobian has full rank, then $dim\boldsymbol{R}(\boldsymbol{J}) = m$ and $dim\boldsymbol{N}(\boldsymbol{J}) = n - m$ are obtained, and the range space of $\boldsymbol{J}(\boldsymbol{\theta})$ spans the entire $\mathbb{R}^m$. That is, any joint velocity in the null space of the manipulator Jacobian will not affect the task velocity. Since this motion of the joints is not observed in the task-space, it is referred to as self-motion. These extra dof allow the robot manipulator to perform more dexterous manipulation and/or provide the robot manipulator with increased flexibility for execution of sophisticated sub–tasks.

For a kinematically redundant robot manipulator, since Jacobian is not a square matrix ($n > m$), its pseudo–inverse, denoted by $\boldsymbol{J}^+(\boldsymbol{\theta}) \in \mathbb{R}^{n \times m}$, can be defined as

$$\boldsymbol{J}^+ \triangleq \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1} \tag{2.18}$$

which satisfies

$$\boldsymbol{J}\boldsymbol{J}^+ = \boldsymbol{I_m} \tag{2.19}$$

where $\boldsymbol{I_m} \in \mathbb{R}^{m \times m}$ denotes the $m \times m$ identity matrix. The pseudo–inverse of the Jacobian defined by (2.18) satisfies the Moore-Penrose Conditions given below Nakamura (1991)

$$\boldsymbol{J}\boldsymbol{J}^+\boldsymbol{J} = \boldsymbol{J}, \;\; \boldsymbol{J}^+\boldsymbol{J}\boldsymbol{J}^+ = \boldsymbol{J}^+, \;\; (\boldsymbol{J}^+\boldsymbol{J})^T = \boldsymbol{J}^+\boldsymbol{J}, \;\; (\boldsymbol{J}\boldsymbol{J}^+)^T = \boldsymbol{J}\boldsymbol{J}^+. \tag{2.20}$$

In addition to the above properties, the matrix $(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})$ represents the orthogonal projection matrix into the null space of the Jacobian and satisfies the following useful properties

$$(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J}) = \boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J} \quad (2.21)$$

$$(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})^T = \boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J} \quad (2.22)$$

$$\boldsymbol{J}(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J}) = \boldsymbol{0_{m\times n}} \quad (2.23)$$

$$(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\boldsymbol{J}^+ = \boldsymbol{0_{n\times m}} \quad (2.24)$$

where $\boldsymbol{I_n} \in \mathbb{R}^{n\times n}$ denotes the $n \times n$ identity matrix, $\boldsymbol{0_{m\times n}} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{0_{n\times m}} \in \mathbb{R}^{n\times m}$ are zero matrices.

Redundancy resolution is one of the most important problems for kinematically redundant robot manipulators when finding joint position $\boldsymbol{\theta}(t)$ for given end–effector position $\boldsymbol{x}(t)$. The redundancy resolution problem (or inverse kinematics problem) can be solved by inverting either the direct kinematics in (2.14) or the first–order differential kinematics in (2.15) or the second–order differential kinematics in (2.17) Siciliano and Khatib (2008). The conventional solution to the first–order differential kinematics in (2.15) can be expressed by using pseudo–inverse of the Jacobian in (2.18) as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^+\dot{\boldsymbol{x}} + (\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\dot{\boldsymbol{\theta}}_{\boldsymbol{N}} \quad (2.25)$$

where $\dot{\boldsymbol{\theta}}_{\boldsymbol{N}}(t) \in \mathbb{R}^n$ is an arbitrary joint velocity vector. The second term at the right–hand side of (2.25) is a null–space velocity. After setting $\dot{\boldsymbol{\theta}}_{\boldsymbol{N}} = \boldsymbol{0_{n\times 1}}$, the minimum norm velocity solution of (2.15) is obtained as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^+\dot{\boldsymbol{x}}. \quad (2.26)$$

As for the second–order differential kinematics in (2.17), the conventional pseudo–inverse type solution can be formulated as

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}^+(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}}) + (\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\ddot{\boldsymbol{\theta}}_{\boldsymbol{N}} \quad (2.27)$$

where $\ddot{\boldsymbol{\theta}}_{\boldsymbol{N}}(t) \in \mathbb{R}^n$ is an arbitrary joint acceleration vector. Choosing $\ddot{\boldsymbol{\theta}}_{\boldsymbol{N}} = \boldsymbol{0_{n\times 1}}$, the minimum norm acceleration solution of (2.17) is obtained as

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}^+(\ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}}). \quad (2.28)$$

In the subsequent chapters, during the control development, the common assumption that the minimum singular value of the Jacobian matrix, denoted by σ_m, being greater than a known small positive constant $\zeta_j > 0$, such that $\max\{\|\boldsymbol{J}^{-1}(\theta)\|\}$ or

$\max\{\|\boldsymbol{J}^{+}(\boldsymbol{\theta})\|\}$ is known *a priori*, will be utilized and all kinematic singularities are always avoided. The kinematic terms $\boldsymbol{J}(\boldsymbol{\theta})$ and $\boldsymbol{J}^{-1}(\boldsymbol{\theta})$ or $\boldsymbol{J}^{+}(\boldsymbol{\theta})$ are functions of $\boldsymbol{\theta}(t)$ as arguments of trigonometric functions, and hence, remain bounded for all possible $\boldsymbol{\theta}(t)$. It is considered that $\boldsymbol{J}(\boldsymbol{\theta})$, $\boldsymbol{J}^{-1}(\boldsymbol{\theta})$ or $\boldsymbol{J}^{+}(\boldsymbol{\theta})$ are second order differentiable, and $\dot{\boldsymbol{J}}$, $\ddot{\boldsymbol{J}}$, $\dot{\boldsymbol{J}}^{-1}$, $\dot{\boldsymbol{J}}^{+}$, $\ddot{\boldsymbol{J}}^{-1}$, $\ddot{\boldsymbol{J}}^{+} \in \mathcal{L}_{\infty}$ provided their arguments are bounded.

Property 1 *The velocity kinematics in* (2.2) *is linearly parameterizable in the sense that*

$$\boldsymbol{J}\dot{\boldsymbol{\theta}} = \boldsymbol{W}_{\boldsymbol{j}}\boldsymbol{\phi}_{\boldsymbol{j}} \tag{2.29}$$

where $\boldsymbol{W}_{\boldsymbol{j}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{n\times p}$ *denotes a known regression matrix, and* $\boldsymbol{\phi}_{\boldsymbol{j}} \in \mathbb{R}^{p}$ *denotes an unknown constant parameter vector.*

Property 2 *The lower and upper bounds for each unknown parameter* $\boldsymbol{\phi}_{\boldsymbol{j}}$ *can be written as follows*

$$\underline{\phi}_{ji} \leq \phi_{ji} \leq \overline{\phi}_{ji} \tag{2.30}$$

where $\phi_{ji} \in \mathbb{R}$ *denotes the* i^{th} *component of* $\boldsymbol{\phi}_{\boldsymbol{j}} \in \mathbb{R}^{p}$ *and* $\underline{\phi}_{ji}$, $\overline{\phi}_{ji} \in \mathbb{R}$ *denote the* i^{th} *components of lower and upper bounds* $\underline{\boldsymbol{\phi}}_{\boldsymbol{j}}$, $\overline{\boldsymbol{\phi}}_{\boldsymbol{j}} \in \mathbb{R}^{p}$*, respectively, which are defined as follows*

$$\underline{\boldsymbol{\phi}}_{\boldsymbol{j}} \triangleq \left[\underline{\phi}_{j1}, \underline{\phi}_{j2}, \ldots \underline{\phi}_{jp}\right]^{T} \tag{2.31}$$

$$\overline{\boldsymbol{\phi}}_{\boldsymbol{j}} \triangleq \left[\overline{\phi}_{j1}, \overline{\phi}_{j2}, \ldots \overline{\phi}_{jp}\right]^{T}. \tag{2.32}$$

2.1.2. Dynamic Model

Dynamic modeling of a robot manipulator concerns the derivation of the equations of motion of the robot manipulator as a function of forces and moments acting on it. Specifically, a mapping is found between the forces and moments exerted on the robot manipulator, and joint positions, velocities and accelerations. The availability of the dynamic model is very useful for mechanical design of the robot manipulator and choice of actuators, testing and design of control strategies, and computer simulation of the motion of the robot manipulator. Computer simulation of the motion of the robot manipulator allows testing control strategies and motion planning techniques without using the actual system. Computation of the forces and torques required for the execution of typical motions provides useful information for the design of joints, drivers and actuators.

In the literature Spong and Vidyasagar (1989), mainly three formulations are used to derive the dynamic model of robot manipulators: Lagrange, Hamilton and Newton-Euler methods. Lagrange's method uses derivations of kinetic and potential energies of a robot manipulator. Hamilton's method uses integrations of the changes in kinetic and potential energies of a robot manipulator. Newton-Euler method uses the computation of equation of motion through a recursive formulation.

Using Lagrange's method, the dynamic model of an n dof revolute joint robot manipulator can be described as Spong and Vidyasagar (1989)

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t) = \boldsymbol{\tau}(t) \tag{2.33}$$

where $\boldsymbol{\theta}(t)$, $\dot{\boldsymbol{\theta}}(t)$, $\ddot{\boldsymbol{\theta}}(t) \in \mathbb{R}^n$ are the joint position, velocity and acceleration vectors, respectively, $\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ represents the generalized mass matrix, $\boldsymbol{\tau}(t) \in \mathbb{R}^n$ is the control input torque, and the rest of the dynamical terms are combined in $\boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t) \in \mathbb{R}^n$ and can be defined as

$$\boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t) = \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{F_d}\dot{\boldsymbol{\theta}} + \boldsymbol{\tau_d} \tag{2.34}$$

where $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{n\times n}$ represents centripetal–Coriolis effects, $\boldsymbol{G}(\boldsymbol{\theta}) \in \mathbb{R}^n$ represents gravitational effects, $\boldsymbol{F_d} \in \mathbb{R}^{n\times n}$ is the constant, positive–definite, diagonal, dynamic frictional effects, $\boldsymbol{\tau_d}(t) \in \mathbb{R}^n$ is a bounded disturbance vector that represents other unmodeled dynamics (e.g., static friction). The dynamic modeling terms in (2.33) and (2.34) exhibit the following properties, assumptions and standard remarks which are employed during control development and stability analysis in the subsequent chapters.

Property 3 *The generalized mass matrix $\boldsymbol{M}(\boldsymbol{\theta})$ is symmetric and positive–definite, and satisfies the following inequalities Khatib (1983), Nicosia and Tomei (1990), Lewis et al. (2003)*

$$m_1 \|\boldsymbol{\xi}\|^2 \leq \boldsymbol{\xi}^T \boldsymbol{M}(\boldsymbol{\theta})\boldsymbol{\xi} \leq m_2 \|\boldsymbol{\xi}\|^2 \ \forall \boldsymbol{\xi} \in \mathbb{R}^n \tag{2.35}$$

where m_1, $m_2 \in \mathbb{R}$ are known positive constants, and $\|\cdot\|$ denotes the standard Euclidean norm. The induced infinity norm, denoted by $\|\cdot\|_{i\infty}$, of the inverse of the generalized mass matrix is assumed to be bounded as

$$\left\|\boldsymbol{M}^{-1}(\boldsymbol{\theta})\right\|_{i\infty} \leq \zeta_M \tag{2.36}$$

where $\zeta_M \in \mathbb{R}$ is a known positive constant.

Property 4 *The generalized mass and centripetal–Coriolis matrices satisfy the following skew symmetric relationship Khatib (1983), Nicosia and Tomei (1990), Lewis et al. (2003)*

$$\boldsymbol{\xi}^T[\dot{\boldsymbol{M}}(\boldsymbol{\theta}) - 2\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})]\boldsymbol{\xi} = 0 \ \forall \boldsymbol{\xi} \in \mathbb{R}^n. \tag{2.37}$$

Property 5 *The dynamical terms can be linearly parametrized as Khatib (1983), Nicosia and Tomei (1990), Lewis et al. (2003)*

$$\boldsymbol{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})\boldsymbol{\phi} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{F_d}\dot{\boldsymbol{\theta}} \tag{2.38}$$

where $\boldsymbol{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) \in \mathbb{R}^{n \times p}$ *being the regression matrix, and* $\boldsymbol{\phi} \in \mathbb{R}^p$ *is the constant parameter vector.*

Property 6 *The dynamical terms can be upper bounded as Khatib (1983), Nicosia and Tomei (1990), Lewis et al. (2003)*

$$\left\| \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \right\|_{i\infty} \leq \zeta_{c1} \left\| \dot{\boldsymbol{\theta}} \right\| \tag{2.39}$$

$$\|\boldsymbol{G}(\boldsymbol{\theta})\| \leq \zeta_g \tag{2.40}$$

$$\|\boldsymbol{F_d}\|_{i\infty} \leq \zeta_{fd} \tag{2.41}$$

$$\|\boldsymbol{\tau_d}\| \leq \zeta_d \tag{2.42}$$

where ζ_{c1}, ζ_g, ζ_{fd}, ζ_d *denote known positive bounding constants.*

Property 7 *The centripetal–Coriolis matrix satisfies the following switching relationship Khatib (1983), Nicosia and Tomei (1990), Lewis et al. (2003)*

$$\boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\xi})\boldsymbol{\nu} = \boldsymbol{C}(\boldsymbol{\theta}, \boldsymbol{\nu})\boldsymbol{\xi} \qquad \forall \boldsymbol{\xi}, \boldsymbol{\nu} \in \mathbb{R}^n. \tag{2.43}$$

The entries of centripetal–Coriolis matrix are trigonometric functions of $\boldsymbol{\theta}(t)$ (valid for robots with revolute joints) and they are bounded when $\dot{\boldsymbol{\theta}}(t)$ is bounded.

The entries of $\boldsymbol{G}(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}(t)$ via trigonometric functions only, and as result of this they remain bounded for all possible $\boldsymbol{\theta(t)}$.

The left hand–side of (2.33) is assumed to be at least second order differentiable, with the modeling terms being bounded provided their arguments are bounded. That is $\boldsymbol{M}(\boldsymbol{\theta})$, $\boldsymbol{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, t) \in \mathcal{C}^2$, and $\boldsymbol{M}$, $\dot{\boldsymbol{M}}$, $\ddot{\boldsymbol{M}}$, $\boldsymbol{N}$, $\dot{\boldsymbol{N}}$, $\ddot{\boldsymbol{N}} \in \mathcal{L}_\infty$ when their arguments are bounded.

For teleoperation systems considered in this thesis, the dynamic models of master and slave systems can be described as

$$\boldsymbol{M_i}(\boldsymbol{\theta_i})\ddot{\boldsymbol{\theta}}_{\boldsymbol{i}} + \boldsymbol{C_i}(\boldsymbol{\theta_i}, \dot{\boldsymbol{\theta}}_{\boldsymbol{i}})\dot{\boldsymbol{\theta}}_{\boldsymbol{i}} = \boldsymbol{\tau_i} + \boldsymbol{\tau_{f_i}} \tag{2.44}$$

where the subscript $i = 1$ denotes n_1 dof master robot and the subscript $i = 2$ denotes n_2 dof slave robot. In (2.44), $\boldsymbol{\theta_i}(t), \dot{\boldsymbol{\theta}}_{\boldsymbol{i}}(t), \ddot{\boldsymbol{\theta}}_{\boldsymbol{i}}(t) \in \mathbb{R}^{n_i}$ denote joint positions, velocities, and accelerations, respectively, $\boldsymbol{M_i}(\boldsymbol{\theta_i}) \in \mathbb{R}^{n_i \times n_i}$ denotes generalized mass matrices, $\boldsymbol{C_i}(\boldsymbol{\theta_i}, \dot{\boldsymbol{\theta}}_{\boldsymbol{i}}) \in \mathbb{R}^{n_i \times n_i}$ denotes centripetal-Coriolis matrices, $\boldsymbol{\tau_i}(t) \in \mathbb{R}^{n_i}$ denotes control

input torques, and $\boldsymbol{\tau_{f_1}}(t) \in \mathbb{R}^{n_1}$ denotes human forces acting on the joints of the master robot and $\boldsymbol{\tau_{f_2}}(t) \in \mathbb{R}^{n_2}$ denotes environmental forces acting on the joints of the slave robot. The human and environmental forces $\boldsymbol{\tau_{f_i}}(t)$ are in joint–space and can also be represented as $\boldsymbol{J_i^T \bar{\tau}_{f_i}}$ with $\boldsymbol{J_i(\theta_i)} \in \mathbb{R}^{n_1 \times n_i}$ being Jacobian matrices of master and slave robots while $\boldsymbol{\bar{\tau}_{f_i}}(t) \in \mathbb{R}^{n_1}$ represent human and environmental forces applied to the end–effectors.

From (2.44), an n_1+n_2 dof combined dynamic model in joint–space can be written as follows

$$\begin{bmatrix} \boldsymbol{M_1} & \boldsymbol{0_{n_1 \times n_2}} \\ \boldsymbol{0_{n_2 \times n_1}} & \boldsymbol{M_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\ddot{\theta}_1} \\ \boldsymbol{\ddot{\theta}_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C_1} & \boldsymbol{0_{n_1 \times n_2}} \\ \boldsymbol{0_{n_2 \times n_1}} & \boldsymbol{C_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\dot{\theta}_1} \\ \boldsymbol{\dot{\theta}_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau_1} \\ \boldsymbol{\tau_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau_{f_1}} \\ \boldsymbol{\tau_{f_2}} \end{bmatrix}. \tag{2.45}$$

2.2. Experimental Testbeds

To illustrate the performances of the proposed controllers, experimental studies have been conducted on the following robot manipulators and numerical simulations are performed on their models.

2.2.1. PHANToM Omni Haptic Device

The highlighted technical specifications of the PHANToM Omni haptic device, as shown in Figure 2.2, are a weight of 3lbs 15oz, a nominal position resolution of 0.055mm, back-drive friction of 0.26N, maximum exertable force nominal of 3.3N, continuous exertable force of 0.88N. The device is 6 dof and can give position on three translational and three rotational directions. The device has two links, three active joints and three passive wrist joints connected to the end of the second link. Since the device is only actuated in the first three joints, it generates haptic force feedback in the translational directions on X, Y, and Z. The end-effector of the device has positional sensing on X, Y, and Z measured by digital encoders. For the communication interface, local area network port were used between the device and a computer, and OpenHaptics ToolKit allows real-time applications on the computer. Experimental studies were conducted with ODE3 (Bogacki-Shampine) solver with a fixed step size of 0.01 seconds in MATLAB Simulink. In the experiments, Phantom Toolbox were utilized in MATLAB Simulink Quarc Library to transmit torques

and forces in task–space or joint–space to the Phantom device, and to receive the encoder values, end-effector position, and joint positions of the device.

The following kinematic and dynamic models obtained from Nygaard (2008), Silva et al. (2009), Sansanayuth et al. (2012) are utilized.

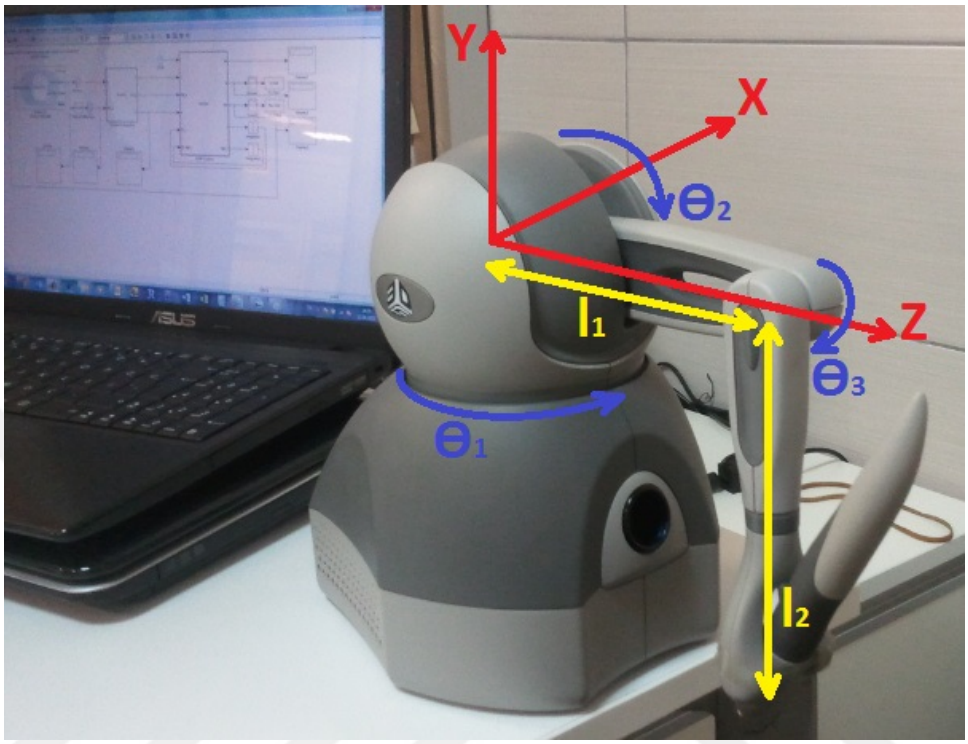

Figure 2.2. PHANToM Omni haptic device

The end-effector position of the PHANToM Omni haptic device can be expressed as

$$\boldsymbol{x}(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = \begin{bmatrix} -s_1(l_1c_2 + l_2s_3) \\ -l_2c_3 + l_1s_2 + l_y \\ c_1(l_1c_2 + l_2s_3) - l_z \end{bmatrix} \tag{2.46}$$

where $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)\ \forall i, j \in \{1, 2, 3\}$ and $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^T$ is the joint position vector, and $l_1 = 0.133$m, $l_2 = 0.133$m represent the first and the second link lengths, respectively, $l_y = 0.023$m, $l_z = 0.168$m are the task–space transformation offsets between the origin of the end–effector and the first joint. The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ can be derived from the forward kinematics and is expressed as

$$\boldsymbol{J}(\boldsymbol{\theta}) = \begin{bmatrix} -c_1(l_1c_2 + l_2s_3) & l_1s_1s_2 & -l_2s_1c_3 \\ 0 & l_1c_2 & l_2s_3 \\ -s_1(l_1c_2 + l_2s_3) & -l_1c_1s_2 & l_2c_1c_3 \end{bmatrix}. \tag{2.47}$$

The dynamic model of the PHANToM Omni haptic device can be written as follows

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{\tau} \tag{2.48}$$

where $\boldsymbol{\theta}(t)$, $\dot{\boldsymbol{\theta}}(t)$, $\ddot{\boldsymbol{\theta}}(t) \in \mathbb{R}^3$ are the joint position, velocity and acceleration vectors, respectively, $\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times3}$ represents the generalized mass matrix, $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{3\times3}$ represents centripetal-Coriolis matrix, $\boldsymbol{G}(\boldsymbol{\theta}) \in \mathbb{R}^3$ represents the gravitational effects, and $\boldsymbol{\tau}(t) \in \mathbb{R}^3$ is control input torque. The generalized mass matrix, centripetal-Coriolis matrix and gravitational effects vector can be represented as

$$\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} k_1 + k_2 c_{2.2} + k_3 c_{2.3} + k_4 c_2 s_3 & k_5 s_2 & 0 \\ k_5 s_2 & k_6 & -0.5 k_4 s_{2-3} \\ 0 & -0.5 k_4 s_{2-3} & k_7 \end{bmatrix} \tag{2.49}$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{2.50}$$

$$\boldsymbol{G}(\boldsymbol{\theta}) = \begin{bmatrix} 0 \\ k_8 c_2 + k_{10} \cos(\theta_2 - 0.5\pi) \\ k_9 s_3 \end{bmatrix} \tag{2.51}$$

where the entries of the centripetal–Coriolis matrix in (2.50) are given as follows

$$\begin{aligned} C_{11} &= -k_2\dot{\theta}_2 s_{2.2} - k_3\dot{\theta}_3 s_{2.3} - 0.5k_4\dot{\theta}_2 s_2 s_3 + 0.5k_4\dot{\theta}_3 c_2 c_3 \\ C_{12} &= -k_2\dot{\theta}_1 s_{2.2} + k_5\dot{\theta}_2 c_2 - 0.5k_4\dot{\theta}_1 s_2 s_3 \\ C_{13} &= -k_3\dot{\theta}_1 s_{2.3} + 0.5k_4\dot{\theta}_1 c_2 c_3 \\ C_{21} &= k_2\dot{\theta}_1 s_{2.2} + 0.5k_4\dot{\theta}_1 c_2 c_3 \\ C_{22} &= 0 \\ C_{23} &= 0.5k_4\dot{\theta}_3 c_{2-3} \\ C_{31} &= k_3\dot{\theta}_1 s_{2.3} + 0.5k_4\dot{\theta}_1 c_2 c_3 \\ C_{32} &= -0.5k_4\dot{\theta}_2 c_{2-3} \\ C_{33} &= 0 \end{aligned} \tag{2.52}$$

where $c_{2.i} = \cos(2\theta_i)$, $s_{2.i} = \sin(2\theta_i)$, $c_{i-j} = \cos(\theta_i - \theta_j)$ and $s_{i-j} = \sin(\theta_i - \theta_j)$

$\forall i, j \in \{1, 2, 3\}$. The real values of the model parameters k_1 to k_{10} are given as

$$\begin{aligned} k_1 &= 1.798 \times 10^{-3}\text{Kg.m}^2 & k_6 &= 2.526 \times 10^{-3}\text{Kg.m}^2 \\ k_2 &= 0.864 \times 10^{-3}\text{Kg.m}^2 & k_7 &= 1.652 \times 10^{-3}\text{Kg.m}^2 \\ k_3 &= 0.486 \times 10^{-3}\text{Kg.m}^2 & k_8 &= 164.158 \times 10^{-3}\text{Kg.m}^2/\text{s}^2 \\ k_4 &= 2.766 \times 10^{-3}\text{Kg.m}^2 & k_9 &= 94.05 \times 10^{-3}\text{Kg.m}^2/\text{s}^2 \\ k_5 &= 0.308 \times 10^{-3}\text{Kg.m}^2 & k_{10} &= 117.294 \times 10^{-3}\text{Kg.m}^2/\text{s}^2. \end{aligned} \tag{2.53}$$

2.2.2. 3 dof RRR Planar Robot Manipulator

The 3 dof revolute revolute revolute (RRR) robot manipulator, as shown in Figure 2.3, has articulated structure with 3 links and 3 actuators, and works on plane. Direct drive actuators of E137576 Maxon Motors with the technical features of nominal voltage of 24 VDC, torque constant of 36.4×10^{-3} Nm/A, speed constant of 263rpm/V, nominal speed of 5530 rpm, nominal torque of 78.2×10^{-3} Nm were used. The motors are driven by Maxon Escon 36/2 DC 4-Q Servo-controller with a maximum power of 72 Watts. For absolute angular measurement, AS5045 Magnetic Rotary Encoders were used with a resolution of 4096 positions per revolution based on contactless magnetic sensor technology. Experimental studies were conducted with ODE3 (Bogacki-Shampine) solver with a fixed step size of 0.001 seconds and utilized Real Time Windows Target in MATLAB Simulink. The control inputs are transmitted to the motor drivers with analog signals and encoder signals are received as quadrature counter inputs. The data transmission between the computer and the drivers is carried out with Humusoft MF624 data acquisition board.

It should be noted that when the orientation of the end–effector of the robot manipulator is not considered, it can be utilized and/or modeled as a kinematically redundant robot manipulator.

The end–effector position of the 3 dof robot manipulator can be obtained to have the following form

$$\boldsymbol{x}(t) = \begin{bmatrix} X(t) \\ Y(t) \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{bmatrix} \tag{2.54}$$

where the link lengths are $l_1 = l_2 = l_3 = 0.127$m, and s_i, c_i, s_{ij}, c_{ij}, s_{ijk}, c_{ijk} represent $\sin(\theta_i)$, $\cos(\theta_i)$, $\sin(\theta_i + \theta_j)$, $\cos(\theta_i + \theta_j)$, $\sin(\theta_i + \theta_j + \theta_k)$, $\cos(\theta_i + \theta_j + \theta_k)$ $(i, j, k$

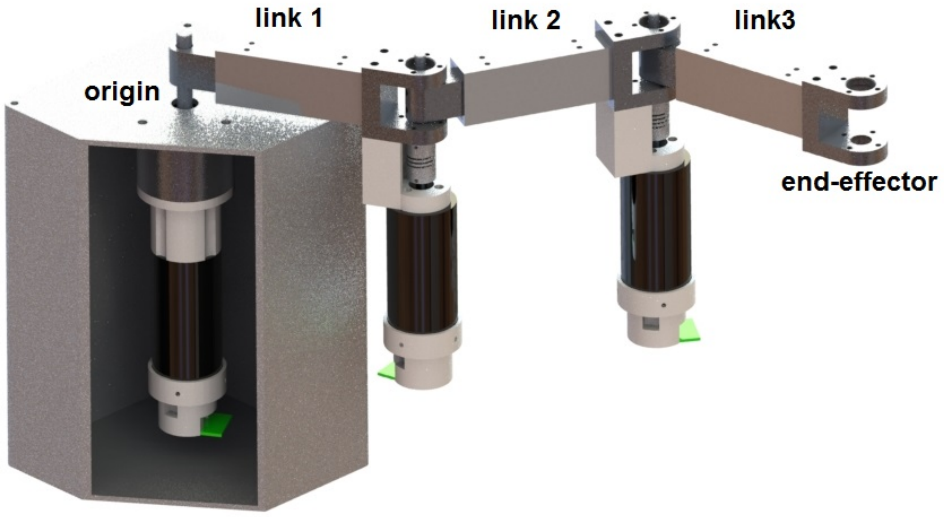

Figure 2.3. 3 dof planar robot manipulator.

$\in \{1, 2, 3\}$), respectively. The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{2\times 3}$ is obtained as

$$\boldsymbol{J}(\boldsymbol{\theta}) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix}. \tag{2.55}$$

The dynamic model of the 3 dof robot manipulator can be written as follows

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} = \boldsymbol{\tau} \tag{2.56}$$

where $\boldsymbol{\theta}(t)$, $\dot{\boldsymbol{\theta}}(t)$, $\ddot{\boldsymbol{\theta}}(t) \in \mathbb{R}^3$ are the joint position, velocity and acceleration vectors, respectively, $\boldsymbol{M}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times 3}$ represents the generalized mass matrix, $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{3\times 3}$ represents centripetal-Coriolis matrix, and $\boldsymbol{\tau}(t) \in \mathbb{R}^3$ is the control input torque. The generalized mass matrix and centripetal-Coriolis matrix can be represented as

$$\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{12} & M_{22} & M_{23} \\ M_{13} & M_{23} & M_{33} \end{bmatrix} \tag{2.57}$$

$$\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{2.58}$$

where the entries of the generalized mass matrix in (2.57) and the entries of the centripetal–

Coriolis matrix in (2.58) are given as follows

$$\begin{aligned} M_{11} &= p_1 c_2 + p_2(c_3 + c_{23}) + p_3 & M_{22} &= p_2 c_3 + p_4 \\ M_{12} &= p_6 c_2 + p_2 c_3 + p_7 c_{23} + p_4 & M_{23} &= p_7 c_3 + p_5 \\ M_{13} &= p_7(c_3 + c_{23}) & M_{33} &= p_5 \end{aligned} \tag{2.59}$$

$$\begin{aligned} C_{11} &= -(p_1 s_2 + p_2 s_{23})\dot{\theta}_2 - p_2(s_3 + s_{23})\dot{\theta}_3 \\ C_{12} &= -(p_6 s_2 + p_7 s_{23})\dot{\theta}_2 - p_2(s_3 + s_{23})\dot{\theta}_3 \\ C_{13} &= -p_7(s_3 + s_{23})\dot{\theta}_3 \\ C_{21} &= (p_6 s_2 + p_7 s_{23})\dot{\theta}_1 \\ C_{22} &= -p_2 s_3 \dot{\theta}_3 \\ C_{23} &= -p_2 s_3 \dot{\theta}_1 - p_7 s_3 \dot{\theta}_3 \\ C_{31} &= p_7(s_3 + s_{23})\dot{\theta}_1 \\ C_{32} &= p_2 s_3 \dot{\theta}_1 + p_7 s_3 \dot{\theta}_2 \\ C_{33} &= 0 \end{aligned} \tag{2.60}$$

where p_i ($i \in \{1, ..., 7\}$) denote the model parameters (*i.e.*, mass of the links, center of mass of each link, link lengths) and are given as,

$$\begin{aligned} p_1 &= 21.3 \times 10^{-3}\text{Kg.m}^2 \\ p_2 &= 2.9 \times 10^{-3}\text{Kg.m}^2 \\ p_3 &= 43.3 \times 10^{-3}\text{Kg.m}^2 \\ p_4 &= 17.7 \times 10^{-3}\text{Kg.m}^2 \\ p_5 &= 1.7 \times 10^{-3}\text{Kg.m}^2 \\ p_6 &= 10.6 \times 10^{-3}\text{Kg.m}^2 \\ p_7 &= 1.5 \times 10^{-3}\text{Kg.m}^2. \end{aligned} \tag{2.61}$$

It should be noted that the gravitational effects were not considered since the manipulator is moving horizontally on plane. In addition, the frictional effects were also neglected since it was observed that they do not have a significant effect.

CHAPTER 3

NULL–SPACE CONTROL OF KINEMATICALLY REDUNDANT ROBOT MANIPULATORS

This chapter focuses on the null-space control problem of kinematically redundant robot manipulators. Specifically, for kinematically redundant robot manipulators having one extra dof, a novel null–space controller is designed to ensure the use of the extra dof for a possible secondary control objective while still ensuring the task–space tracking objective via the designs of two task–space controllers, one exact model knowledge and one adaptive. Combined stability analysis of the task–space tracking objective and sub–task objective are performed. Exponential stability of the exact model knowledge and asymptotic stability of the adaptive controller are then ensured via Lyapunov type arguments. Numerical studies for the proposed controllers are presented to illustrate the liability of the proposed method.

3.1. Mathematical Models and Properties

In this chapter, the kinematic model of an n dof revolute joint robot manipulator given in (2.14) is utilized. The dynamic model given in (2.33) is utilized when $\boldsymbol{\tau_d}(t)$ in (2.34) is ignored. The dynamic terms satisfy Properties 3, 4 and 5.

3.2. Error System Formulation

The task–space tracking error, denoted by $\boldsymbol{e}(t) \in \mathbb{R}^m$, is defined as

$$\boldsymbol{e} \triangleq \boldsymbol{x_d} - \boldsymbol{x} \tag{3.1}$$

where $\boldsymbol{x_d}(t) \in \mathbb{R}^m$ denotes the desired task–space trajectory.

After taking the time derivative of (3.1) and substituting velocity kinematics in (2.15), following expression can be obtained

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_{\boldsymbol{d}} - \boldsymbol{J}\dot{\boldsymbol{\theta}}. \tag{3.2}$$

To further facilitate the subsequent analysis, an auxiliary error–like term, denoted by $\boldsymbol{r}(t) \in \mathbb{R}^n$, is defined as follows

$$\boldsymbol{r} \triangleq \boldsymbol{J}^{+}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha e}) + (\boldsymbol{I_n} - \boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{h} - \dot{\boldsymbol{\theta}} \tag{3.3}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{m\times m}$ is a constant, positive–definite, diagonal control gain matrix, and $\boldsymbol{h}(t) \in \mathbb{R}^n$ is a null–space controller that will be designed later. Premultiplying (3.3) by $\boldsymbol{J}$ and substituting (2.15) and (3.2), following expression can be obtained

$$\dot{\boldsymbol{e}} = -\boldsymbol{\alpha e} + \boldsymbol{Jr} \tag{3.4}$$

where (2.19), (2.21), (2.22), (2.23), (2.24) and (3.2) were utilized. For simplicity, another auxiliary term, denoted by $\boldsymbol{r_a}(t) \in \mathbb{R}^n$, is defined as follows

$$\boldsymbol{r_a} \triangleq \boldsymbol{J}^{+}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha e}) + (\boldsymbol{I_n} - \boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{h} \tag{3.5}$$

which can be substituted into (3.3) to obtain

$$\boldsymbol{r} = \boldsymbol{r_a} - \dot{\boldsymbol{\theta}}. \tag{3.6}$$

Taking the time derivative of (3.6), pre–multiplying by generalized mass matrix $\boldsymbol{M}(\boldsymbol{\theta})$, and then substituting (2.33), following open–loop error dynamics are obtained

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{Y}\boldsymbol{\phi} - \boldsymbol{Cr} - \boldsymbol{\tau} \tag{3.7}$$

where

$$\boldsymbol{Y}\boldsymbol{\phi} = \boldsymbol{M}\dot{\boldsymbol{r}}_{\boldsymbol{a}} + \boldsymbol{Cr_a} + \boldsymbol{G} + \boldsymbol{F_d}\dot{\boldsymbol{\theta}} \tag{3.8}$$

with $\boldsymbol{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{r_a}, \dot{\boldsymbol{r}}_{\boldsymbol{a}}) \in \mathbb{R}^{n\times p}$ being the regression matrix depending only on known and available quantities and $\boldsymbol{\phi} \in \mathbb{R}^p$ is a vector containing dynamic model parameters.

3.3. Controller Design and Stability Analysis

In this section, control design and the accompanying stability analysis will be presented when exact knowledge of the robot dynamics is available (*i.e.*, (3.8) is known and available for control design).

From the error system development in Section 3.2 and the subsequent stability analysis, the control input torque $\boldsymbol{\tau}(t)$ is designed as follows

$$\boldsymbol{\tau} = \boldsymbol{Y}\boldsymbol{\phi} + \boldsymbol{K_r}\boldsymbol{r} + \boldsymbol{J}^{T}\boldsymbol{e} - y_s\boldsymbol{J}_{\boldsymbol{s}}^{T} \tag{3.9}$$

where $y_s(\boldsymbol{\theta}) \in \mathbb{R}$ is the yet to be designed sub–task function, $\boldsymbol{J_s}(\boldsymbol{\theta}) \in \mathbb{R}^{1\times n}$ is also yet to be designed, and $\boldsymbol{K_r} \in \mathbb{R}^{n\times n}$ is a constant, positive–definite, diagonal control gain matrix.

The exact model knowledge version of the adaptive controller in Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a) expressed by using the notation in this chapter is of the following form

$$\boldsymbol{\tau} = \boldsymbol{Y}\boldsymbol{\phi} + \boldsymbol{K_r}\boldsymbol{r} + \boldsymbol{J}^T\boldsymbol{e}. \tag{3.10}$$

As opposed to the controller of (3.10), the proposed controller in (3.9) includes the new term $y_s\boldsymbol{J_s}^T$ which will subsequently allow us to carry out a combined stability analysis for both task–space tracking and sub–task objectives. This is an important **novelty** when compared with Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a).

After substituting (3.9) into (3.7), the closed–loop error system for $\boldsymbol{r}(t)$ can be obtained as follows

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{C}\boldsymbol{r} - \boldsymbol{K_r}\boldsymbol{r} - \boldsymbol{J}^T\boldsymbol{e} + y_s\boldsymbol{J_s}^T. \tag{3.11}$$

The sub–task function y_s will be designed to depend only on $\boldsymbol{\theta}$. Special care must be taken when designing the sub–task function y_s. Specifically, in the subsequent stability analysis convergence of y_s to zero will be proven. As a result, when designing the sub–task function, this should be taken into account and functions that could be driven to zero must be considered.

The time derivative of y_s can be obtained as

$$\dot{y}_s = \boldsymbol{J_s}\dot{\boldsymbol{\theta}} \tag{3.12}$$

where $\boldsymbol{J_s}(\boldsymbol{\theta})$ is an auxiliary Jacobian–like vector defined as

$$\boldsymbol{J_s} \triangleq \frac{\partial y_s}{\partial \boldsymbol{\theta}}. \tag{3.13}$$

After utilizing (3.3), from (3.12), following expression is obtained

$$\dot{y}_s = \boldsymbol{J_s}\boldsymbol{J}^+(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha}\boldsymbol{e}) + \boldsymbol{J_s}(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\boldsymbol{h} - \boldsymbol{J_s}\boldsymbol{r}. \tag{3.14}$$

Based on the structure of (3.14) and the subsequent stability analysis, the null–space controller $\boldsymbol{h}$ introduced in (3.3) is designed as follows

$$\boldsymbol{h} = -\frac{(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\boldsymbol{J_s}^T}{\left\|\boldsymbol{J_s}(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\right\|^2}[k_s y_s + \boldsymbol{J_s}\boldsymbol{J}^+(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha}\boldsymbol{e})] \tag{3.15}$$

where $k_s \in \mathbb{R}$ is a positive constant. Notice that, $\boldsymbol{J}\boldsymbol{h} = \boldsymbol{0}_{\boldsymbol{m}\times\boldsymbol{1}}$. Provided that the following sufficient condition holds

$$\left\|\boldsymbol{J_s}(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\right\| > 0 \tag{3.16}$$

there is no singularity issue in $\boldsymbol{h}(t)$.

Mathematically speaking, the term $\|\boldsymbol{J_s}(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})\|$ will be equal to zero when $\boldsymbol{J_s}^T$ is in the null–space of $(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})$. And when this is the case, then, in (3.14), the null–space controller is multiplied with zero. As a result, it is reasonable to assume that (3.16) is satisfied. In addition, it should be noted that the condition in (3.16) was previously introduced in Tatlicioglu et al. (2005) and then utilized in Tatlicioglu et al. (2008)(a), Tatlicioglu et al. (2008)(b), Tatlicioglu et al. (2009).

The null–space controller in Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a) expressed by using the notation in this chapter is of the following form

$$\boldsymbol{h} = -k_s[\boldsymbol{J_s}(\boldsymbol{I_n} - \boldsymbol{J}^+\boldsymbol{J})]^T y_s. \tag{3.17}$$

In contrast to the null–space controller of (3.17), the proposed null–space controller in (3.15) is quite different. The **novel** null–space controller will be useful for the combined stability analysis.

Finally, substituting (3.15) into (3.14), following closed–loop dynamics is obtained for $y_s(t)$

$$\dot{y}_s = -k_s y_s - \boldsymbol{J_s}\boldsymbol{r}. \tag{3.18}$$

Now the combined stability analysis can be proceeded by introducing the following theorem.

Theorem 3.3.1 *The controller in* (3.9) *with the null–space controller in* (3.15) *ensures exponential task–space tracking and exponential sub–task tracking.*

Proof The proof starts by defining a non–negative scalar function (*i.e.*, a Lyapunov function), denoted by $V_1\ (\boldsymbol{e}, \boldsymbol{r}, y_s) \in \mathbb{R}$, as

$$V_1 \triangleq \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} + \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\boldsymbol{r} + \frac{1}{2}y_s^2. \tag{3.19}$$

By using (2.35), it can easily be shown that the following bounds hold for (3.19)

$$\lambda_1 \|\boldsymbol{z}\|^2 \leq V_1 \leq \lambda_2 \|\boldsymbol{z}\|^2 \tag{3.20}$$

where $\lambda_1 \triangleq \frac{1}{2}\min\{1, m_1\}$, $\lambda_2 \triangleq \frac{1}{2}\max\{1, m_2\}$, and $\boldsymbol{z}(t) \in \mathbb{R}^{(m+n+1)\times 1}$ is the combined error vector defined as

$$\boldsymbol{z}(t) \triangleq \begin{bmatrix} \boldsymbol{e}^T & \boldsymbol{r}^T & y_s \end{bmatrix}^T. \tag{3.21}$$

After taking the time derivative of (3.19), following expression is obtained

$$\dot{V}_1 = \boldsymbol{e}^T\dot{\boldsymbol{e}} + \boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + y_s\dot{y}_s \tag{3.22}$$

Substituting (3.4), (3.11) and (3.18) into (3.22) yields

$$\dot{V}_1 = \boldsymbol{e}^T(-\boldsymbol{\alpha e} + \boldsymbol{Jr}) + \boldsymbol{r}^T(-\boldsymbol{Cr} - \boldsymbol{K_r r} - \boldsymbol{J}^T\boldsymbol{e} + \frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + y_s\boldsymbol{J_s}^T) + y_s(-k_s y_s - \boldsymbol{J_s r}) \tag{3.23}$$

and after canceling common terms, following expression is obtained

$$\dot{V}_1 = -\boldsymbol{e}^T\boldsymbol{\alpha e} - \boldsymbol{r}^T\boldsymbol{K_r r} - k_s y_s^2 \tag{3.24}$$

where (2.37) was also utilized. Notice that, (3.24) can be upper bounded as

$$\dot{V}_1 \leq -\lambda_3 \left\|\boldsymbol{z}\right\|^2 \tag{3.25}$$

where $\lambda_3 \triangleq \min\{\lambda_{\min}(\boldsymbol{\alpha}), \lambda_{\min}(\boldsymbol{K_r}), k_s\}$ with $\lambda_{\min}(\cdot)$ denoting the minimum eigenvalue of a matrix.

From (3.19), (3.20) and (3.24), (3.25), it is easy to see that $V_1(\boldsymbol{e}, \boldsymbol{r}, y_s)$ is exponentially stable Khalil (2002). Therefore $\boldsymbol{z}(t)$ and thus $\boldsymbol{e}(t)$, $\boldsymbol{r}(t)$, $y_s(t)$ are exponentially stable. So, $V_1(\boldsymbol{e}, \boldsymbol{r}, y_s) \in \mathcal{L}_\infty$, and thus $\boldsymbol{z}(t)$, $\boldsymbol{e}(t)$, $\boldsymbol{r}(t)$, $y_s(t) \in \mathcal{L}_\infty$. Based on the boundedness of the desired task–space trajectory, from (3.1), it is clear that $\boldsymbol{x}(t) \in \mathcal{L}_\infty$. The boundedness of $\boldsymbol{e}(t)$ and $\boldsymbol{r}(t)$ can be utilized along with (3.4) to conclude that $\dot{\boldsymbol{x}}(t) \in \mathcal{L}_\infty$. Above boundedness statements can be utilized along with (3.3) to prove that $\dot{\boldsymbol{\theta}}(t) \in \mathcal{L}_\infty$ and along with (3.7) to prove that $\dot{\boldsymbol{r}}(t) \in \mathcal{L}_\infty$. The robot manipulator dynamics in (2.33) can be utilized to demonstrate $\boldsymbol{\tau}(t) \in \mathcal{L}_\infty$. Standard signal chasing arguments can then be used to prove that all signals remain bounded under the closed–loop operation.

In Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a), the Lyapunov functions of task–space tracking (*i.e.*, V_t) and sub–task objectives (*i.e.*, V_s) with their time derivatives expressed by using the notation in this chapter are of the following forms

$$\begin{aligned} V_t &= \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} + \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\boldsymbol{r} \text{ with } \dot{V}_t = -\boldsymbol{e}^T\boldsymbol{\alpha e} - \boldsymbol{r}^T\boldsymbol{K_r r} \\ V_s &= \frac{1}{2}y_s^2 \text{ with } \dot{V}_s \leq -\gamma y_s^2 + \epsilon \end{aligned} \tag{3.26}$$

where ϵ, γ are positive scalar constants. While the exact model knowledge version of the task–space controller of Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a) provided exponential stability, their null–space controller achieved ultimately bounded sub–task tracking in the following sense

$$|y_s(t)| \leq \sqrt{y_s^2(t_0)\exp(-2\gamma t) + \frac{\epsilon}{\gamma}}.$$

On the other hand, the proposed combined stability analysis achieves exponential task–space tracking and exponential null–space control which is a major improvement over Tatlicioglu et al. (2005) and Tatlicioglu et al. (2008)(a).

3.3.1. Adaptive Controller Extension

In this section, the exact model knowledge controller of (3.9) is modified to compensate for parametric uncertainties in the dynamic model (*i.e.*, ϕ in (3.8) is considered as uncertain and thus cannot be utilized in the control design).

The adaptive controller is designed as

$$\boldsymbol{\tau} = \boldsymbol{Y}\hat{\boldsymbol{\phi}} + \boldsymbol{K_r}\boldsymbol{r} + \boldsymbol{J}^T\boldsymbol{e} - y_s\boldsymbol{J_s}^T \tag{3.27}$$

where $\hat{\boldsymbol{\phi}}(t) \in \mathbb{R}^p$ is the estimate of the uncertain parameter vector $\boldsymbol{\phi}$, and is updated according to

$$\dot{\hat{\boldsymbol{\phi}}} = \boldsymbol{\Gamma}\boldsymbol{Y}^T\boldsymbol{r} \tag{3.28}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times p}$ is a constant, positive–definite, diagonal, adaptation gain matrix, and $\boldsymbol{Y}$ is the regression matrix introduced in (3.8).

The only difference between the exact model knowledge controller and the adaptive controller is the time–varying update rule $\hat{\boldsymbol{\phi}}(t)$ introduced to compensate for the lack of accurate knowledge of dynamic model parameters.

The parameter estimation error vector $\widetilde{\boldsymbol{\phi}}(t) \in \mathbb{R}^p$ is defined as

$$\widetilde{\boldsymbol{\phi}} \triangleq \boldsymbol{\phi} - \hat{\boldsymbol{\phi}}. \tag{3.29}$$

Substituting (3.27) and (3.29) into (3.7), the closed–loop error system for $\boldsymbol{r}(t)$ can be obtained as follows

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{C}\boldsymbol{r} - \boldsymbol{K_r}\boldsymbol{r} - \boldsymbol{J}^T\boldsymbol{e} + y_s\boldsymbol{J_s}^T + \boldsymbol{Y}\widetilde{\boldsymbol{\phi}}. \tag{3.30}$$

Since the dynamics of task–space tracking error and sub–task function y_s do not depend on the dynamic model parameters they remain unchanged. Following theorem can now be stated to analyze the stability of the adaptive controller.

Theorem 3.3.2 *The adaptive controller in* (3.27) *along with the null–space controller in* (3.15) *and the parameter update law in* (3.28) *ensures asymptotic task–space tracking and asymptotic sub–task tracking in the sense that*

$$\|\boldsymbol{e}(t)\| \to 0\,,\ |y_s(t)| \to 0 \text{ as } t \to \infty. \tag{3.31}$$

Proof The proof starts by defining a non–negative scalar function, denoted by $V_2\ (\boldsymbol{e}, \boldsymbol{r}, y_s, \widetilde{\boldsymbol{\phi}}) \in \mathbb{R}$, as

$$V_2 \triangleq V_1 + \frac{1}{2}\widetilde{\boldsymbol{\phi}}^T\boldsymbol{\Gamma}^{-1}\widetilde{\boldsymbol{\phi}} \tag{3.32}$$

where $V_1(\boldsymbol{e}, \boldsymbol{r}, y_s)$ was defined in (3.19). Similar to (3.20), following bounds can be obtained for V_2

$$\lambda_4 \left\| \boldsymbol{s} \right\|^2 \leq V_2 \leq \lambda_5 \left\| \boldsymbol{s} \right\|^2 \tag{3.33}$$

where $\lambda_4 \triangleq \frac{1}{2} \min\{1, m_1, \lambda_{\max}(\boldsymbol{\Gamma})\}$, $\lambda_5 \triangleq \frac{1}{2} \max\{1, m_2, \lambda_{\min}(\boldsymbol{\Gamma})\}$ with $\lambda_{\max}(\cdot)$ denoting the maximum eigenvalue of a matrix, and $\boldsymbol{s}(t) \in \mathbb{R}^{(m+n+p+1)\times 1}$ is the combined error vector defined as

$$\boldsymbol{s}(t) \triangleq \begin{bmatrix} \boldsymbol{e}^T & \boldsymbol{r}^T & y_s & \widetilde{\boldsymbol{\phi}}^T \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{z}^T & \widetilde{\boldsymbol{\phi}}^T \end{bmatrix}^T. \tag{3.34}$$

After taking the time derivative of (3.32), following expression is obtained

$$\dot{V}_2 = \boldsymbol{e}^T \dot{\boldsymbol{e}} + \boldsymbol{r}^T \boldsymbol{M} \dot{\boldsymbol{r}} + \frac{1}{2} \boldsymbol{r}^T \dot{\boldsymbol{M}} \boldsymbol{r} + y_s \dot{y}_s + \widetilde{\boldsymbol{\phi}}^T \boldsymbol{\Gamma}^{-1} \dot{\widetilde{\boldsymbol{\phi}}}. \tag{3.35}$$

Substituting (3.4), (3.18), (3.30), time derivative of (3.29) along with (3.28), and then canceling common terms, following expression is obtained

$$\dot{V}_2 = -\boldsymbol{e}^T \boldsymbol{\alpha} \boldsymbol{e} - \boldsymbol{r}^T \boldsymbol{K}_{\boldsymbol{r}} \boldsymbol{r} - k_s y_s^2 \leq -\lambda_3 \left\| \boldsymbol{z} \right\|^2 \tag{3.36}$$

where (2.37) was utilized and λ_3 was introduced in (3.25).

From (3.32), (3.33) and (3.36), $V_2(\boldsymbol{e}, \boldsymbol{r}, y_s, \widetilde{\boldsymbol{\phi}}) \in \mathcal{L}_\infty$, therefore, $\boldsymbol{s}(t)$, and thus $\boldsymbol{e}(t)$, $\boldsymbol{r}(t)$, $y_s(t)$, $\widetilde{\boldsymbol{\phi}}(t) \in \mathcal{L}_\infty$. Similar to the proof of Theorem 3.3.1, all the signals can be shown to remain bounded under the closed–loop operation.

After integrating (3.36) in time from 0 to $+\infty$, the following expression can be obtained as

$$\int_0^{+\infty} \dot{V}_2(t) dt \leq -\lambda_3 \int_0^{+\infty} \left\| \boldsymbol{z}(t) \right\|^2 dt \tag{3.37}$$

and after evaluating the integral on the left hand–side, recalling $V_2(t) \geq 0$ and then rearranging yields

$$\int_0^{+\infty} \left\| \boldsymbol{z}(t) \right\|^2 dt \leq \frac{V_2(0)}{\lambda_3} \tag{3.38}$$

from which it can be seen that $\boldsymbol{z}(t)$ is square integrable. Barbalat's Lemma in Krstic et al. (1995) can then be utilized to obtain asymptotic task–space tracking and asymptotic null–space control as given in (3.31).

3.4. Simulation Results

To illustrate the performance of the task–space controller and the null–space controller, two sets of simulations were performed for the exact model knowledge and adaptive controllers on the model of a 3 dof planar robot manipulator given in Section 2.2.2. It

is highlighted that in both simulations, the model parameters were utilized to simulate the robot dynamics and they were not utilized in the adaptive controller formulation in (3.27).

For both controllers, the first set of simulations were performed without null–space control. In the second set of simulations, following sub–task function was selected

$$y_s = \theta_1 - 0.5 \tag{3.39}$$

with which the position of the first joint is forced to go to 0.5rad.

The manipulator was initialized to be at rest at the joint position $\boldsymbol{\theta}(0) = [0, 1.5, 1.5]^T$ rad. The desired task–space trajectory was selected as

$$\boldsymbol{x_d} = \begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \begin{bmatrix} 0.02\sin(0.1t) \\ 0.15 - 0.02\cos(0.1t) \end{bmatrix} \text{(m)}. \tag{3.40}$$

For both set of simulations, the controller gains were chosen as $\boldsymbol{\alpha} = 2\boldsymbol{I_2}$, $\boldsymbol{K_r} = 3\boldsymbol{I_3}$, and for the adaptive controller, the adaptation gain was chosen as $\boldsymbol{\Gamma} = 2\boldsymbol{I_7}$ and $k_s = 2$ for the second set of simulations. In the simulations of adaptive controller, the initial values of the parameter update vector were chosen as $\hat{\boldsymbol{\phi}}(0) = 10^{-3} \times [20, 2, 40, 20, 1, 10, 1]^T$. Exact model knowledge and adaptive controllers were shortly named as EMKC and AC, respectively, in the simulation results.

The simulation results for the exact model knowledge controller are presented in Figures 3.1-3.4. In Figures 3.1, 3.2, 3.3 and 3.4, the task–space tracking error, the control input torques, the joint positions, and desired and actual task–space trajectories are shown, respectively. From Figures 3.1 and 3.4, it is clear that task–space tracking objective was met for both simulation sets. From Figure 3.3, it is clear that when there is null–space control, the position of the first joint went to 0.5 rad. thus satisfying the sub–task function in (3.39).

The simulation results for the adaptive controller are presented in Figures 3.5-3.9. In Figures 3.5, 3.6, 3.7, 3.8 and 3.9, the task–space tracking error, the control input torques, the joint positions, desired and actual task–space trajectories, and the estimates of uncertain parameters are shown, respectively. From Figures 3.5 and 3.8, it is clear that task–space tracking objective was met for both simulation sets. From Figure 3.7, it is clear that when there is null–space control, the position of the first joint went to 0.5 rad. thus satisfying the sub–task function in (3.39).

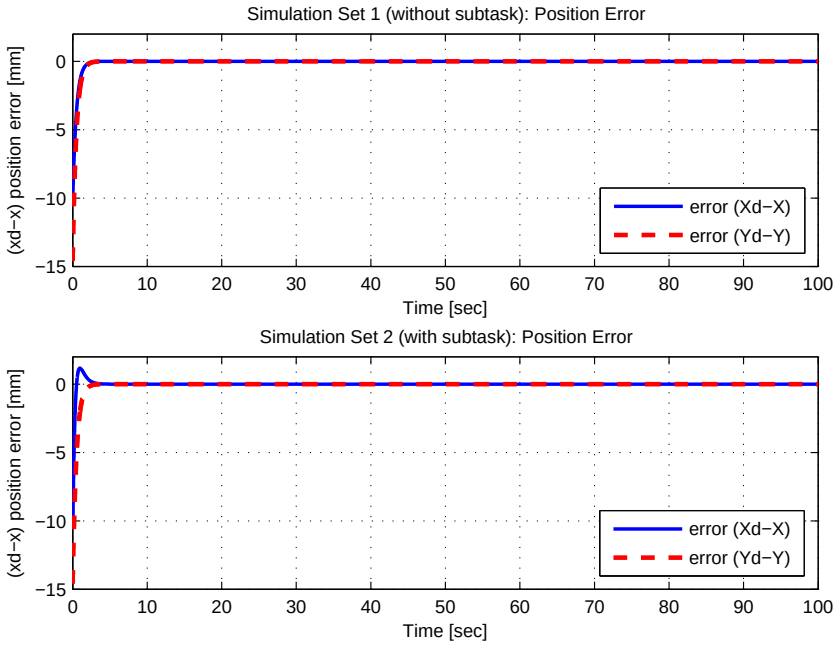

Figure 3.1. Task–space tracking errors $\boldsymbol{e}(t)$ for EMKC.

3.5. Conclusions

This chapter mainly focused on the design of an asymptotically stable null–space controller and presented the design and the corresponding stability analysis of two task–space controllers (one exact model knowledge and one adaptive) for redundant robot manipulators that have one redundant dof to achieve a sub–task objective. The sub–task function has been designed as an error–like function of joint positions. A combined stability analysis that ensured stability and convergence of both end–effector and sub–task function have been presented. The proposed combined analysis is **novel** when compared to the existing literature on control of redundant robot manipulators. Numerical simulations were conducted for both task–space controllers with and without null–space controllers that demonstrated the performance of the proposed null–space controller.

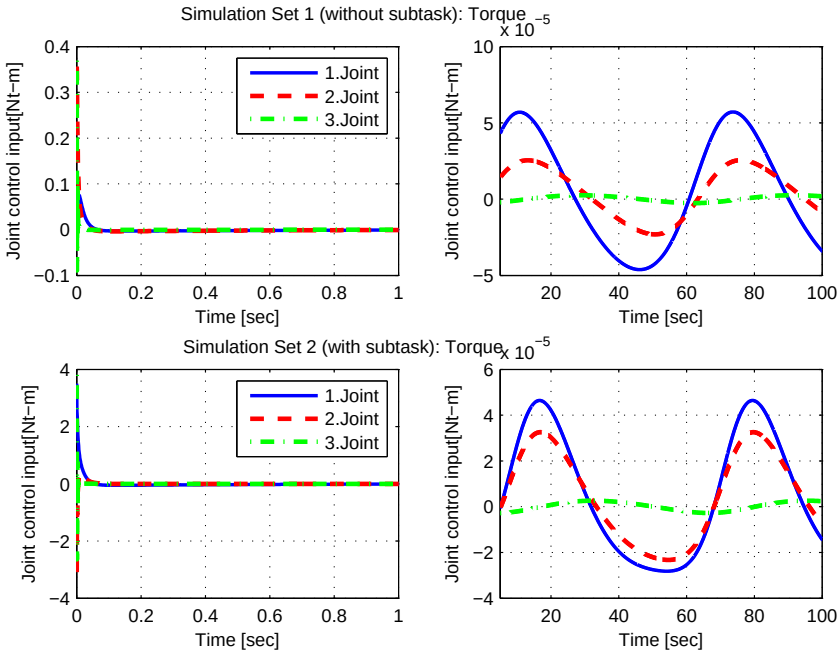

Figure 3.2. Control input torques $\boldsymbol{\tau}(t)$ for EMKC.

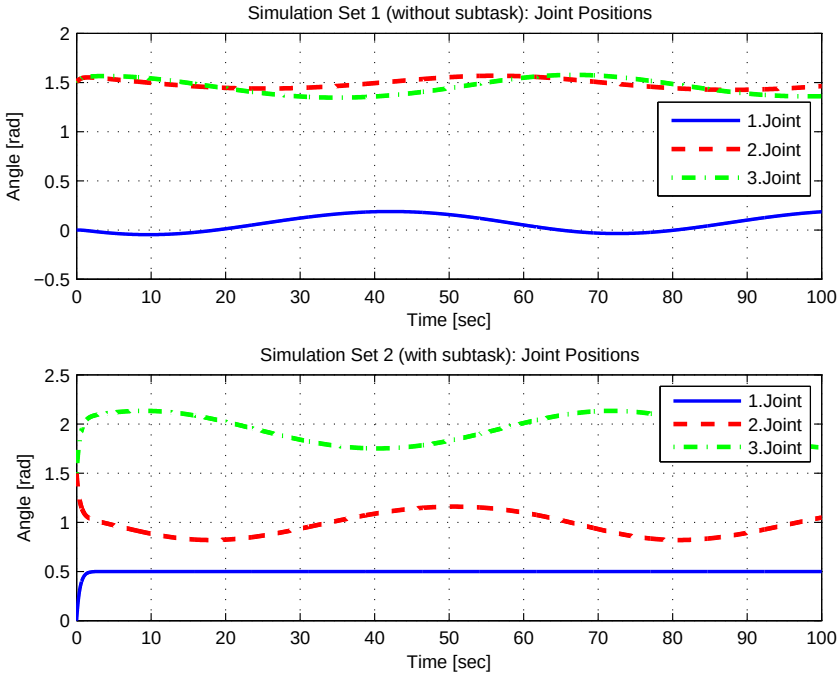

Figure 3.3. Joint positions $\boldsymbol{\theta}(t)$ for EMKC.

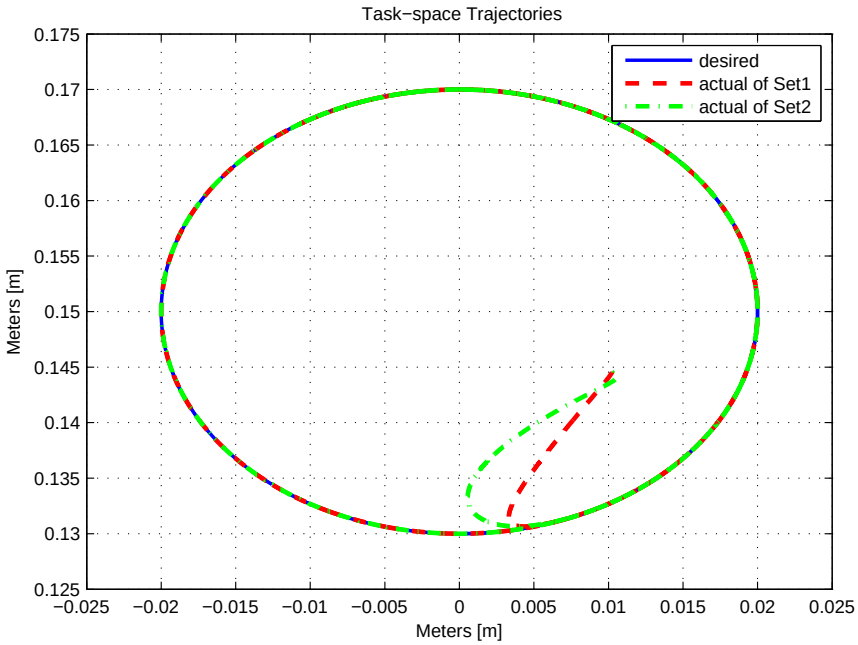

Figure 3.4. Desired $\boldsymbol{x_d}(t)$ and actual $\boldsymbol{x}(t)$ task–space trajectories for EMKC.

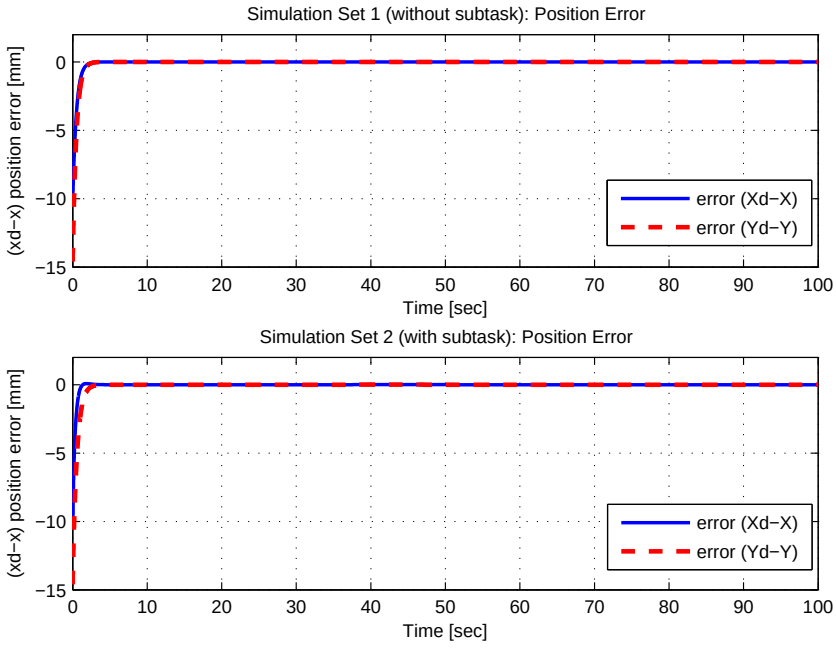

Figure 3.5. Task–space tracking errors $\boldsymbol{e}(t)$ for AC.

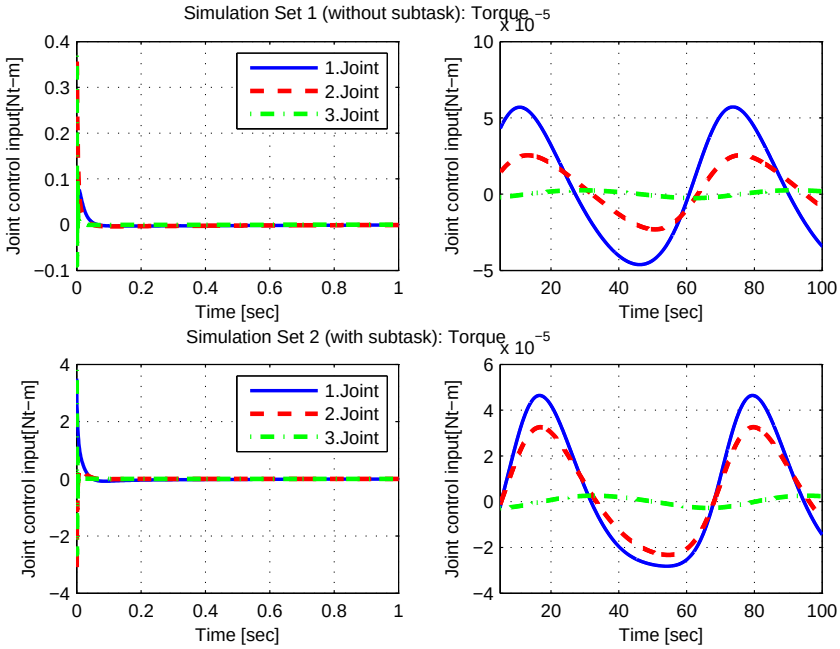

Figure 3.6. Control input torques $\boldsymbol{\tau}(t)$ for AC.

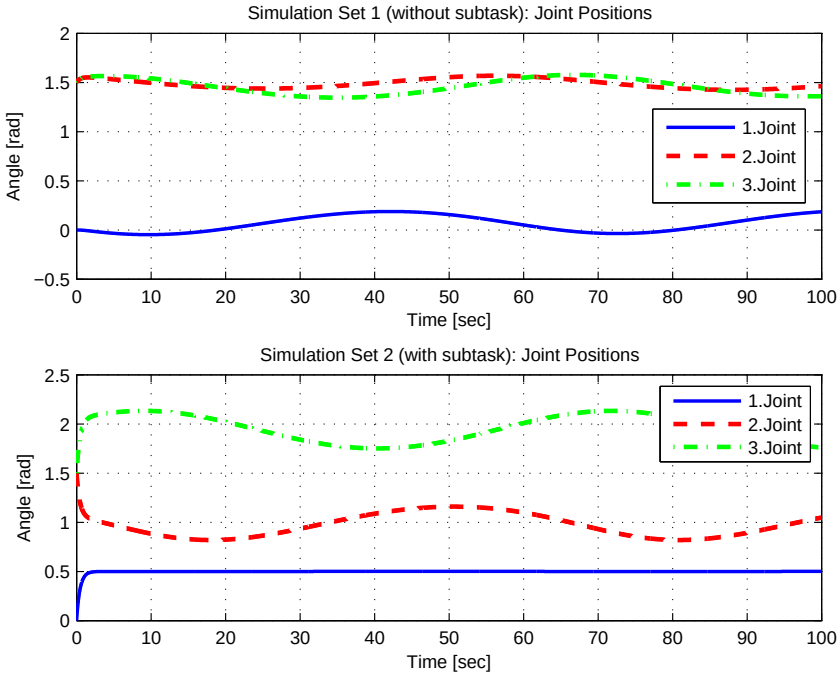

Figure 3.7. Joint positions $\boldsymbol{\theta}(t)$ for AC.

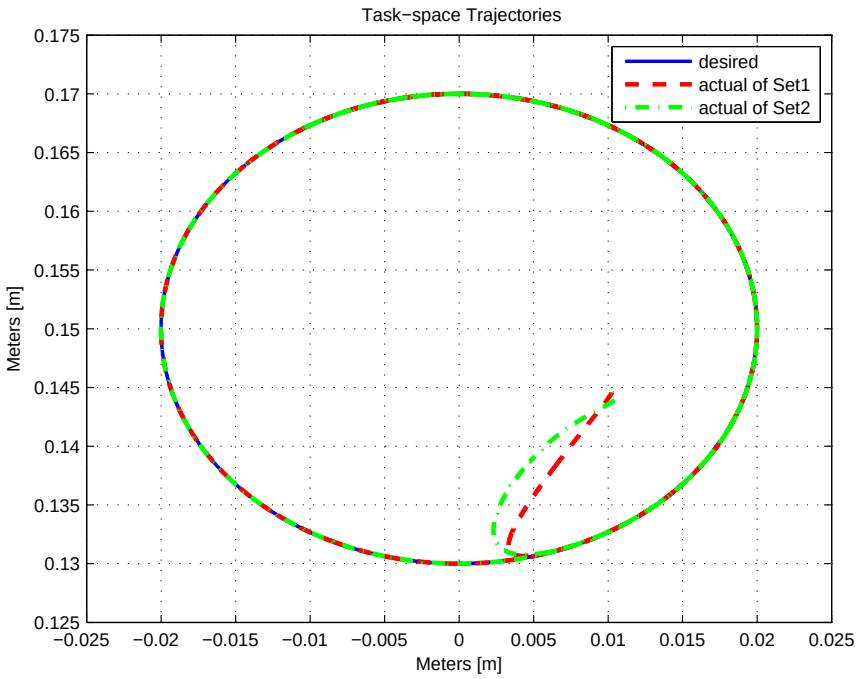

Figure 3.8. Desired $\boldsymbol{x_d}(t)$ and actual $\boldsymbol{x}(t)$ task–space trajectories for AC.

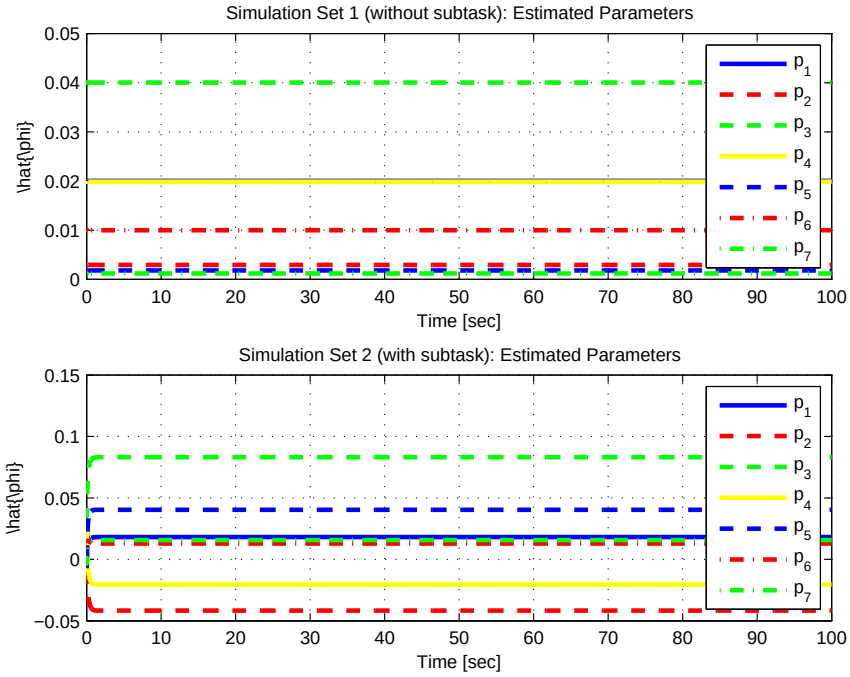

Figure 3.9. Estimates of uncertain parameters $\hat{\boldsymbol{\phi}}(t)$ for AC.

CHAPTER 4

PSEUDO–INVERSE FREE CONTROL OF KINEMATICALLY REDUNDANT ROBOT MANIPULATORS

The proposed methodology in Chapter 3 was applicable to kinematically redundant robot manipulators with one redundant dof (*i.e.,* $n-m=1$). The aim of this chapter is to design an extended task–space controller for redundant robot manipulators having more than one redundant dof (*i.e.,* $n-m>1$). The extended task–space controller is designed by integrating the manipulator Jacobian matrix with the sub–task Jacobian matrix. Sub–task functions, that depend on the joint positions and have equality constraints, are properly chosen up to the amount of the redundant dof. An adaptive controller is designed to deal with parametric uncertainties in robot dynamics and sub–task objectives in the amount of number of redundant dof are achieved while still ensuring the task–space control objective. The stability of task–space and sub–task objectives are obtained via Lyapunov based arguments. An experimental study is presented to illustrate the performance of the proposed method by considering a **novel** sub–task objective.

4.1. Mathematical Models and Properties

In this chapter, the kinematic model of an n dof revolute joint robot manipulator given in (2.14) is utilized. The dynamic model given in (2.33) is utilized when $\boldsymbol{\tau_d}(t)$ in (2.34) is ignored. The dynamic terms satisfy Properties 3, 4 and 5.

4.2. Extended Task–space Formulation

The sub–task function, denoted by $\boldsymbol{y_s}(\boldsymbol{\theta}) \in \mathbb{R}^{(n-m)}$, will be specifically designed to depend only on $\boldsymbol{\theta}(t)$ as

$$\boldsymbol{y_s} = \boldsymbol{f_s}(\boldsymbol{\theta}) \tag{4.1}$$

where $\boldsymbol{f_s} : \mathbb{R}^n \rightarrow \mathbb{R}^{(n-m)}$. The time derivative of $\boldsymbol{y_s}$ can then be obtained as

$$\dot{\boldsymbol{y}}_{\boldsymbol{s}} = \boldsymbol{J_s}\dot{\boldsymbol{\theta}} \tag{4.2}$$

where $\boldsymbol{J_s}(\boldsymbol{\theta}) \in \mathbb{R}^{(n-m)\times n}$ is the sub–task Jacobian matrix defined as

$$\boldsymbol{J_s} \triangleq \frac{\partial \boldsymbol{y_s}}{\partial \boldsymbol{\theta}}. \tag{4.3}$$

In order to avoid utilizing pseudo–inverse of the Jacobian matrix, the m dimensional forward kinematics and the $n - m$ dimensional sub–task objectives are combined into $\bar{\boldsymbol{x}}(t) \in \mathbb{R}^n$ defined as

$$\bar{\boldsymbol{x}}(t) \triangleq \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y_s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{\theta}) \\ \boldsymbol{f_s}(\boldsymbol{\theta}) \end{bmatrix}. \tag{4.4}$$

Taking the time derivative of (4.4) yields

$$\dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{J}}\dot{\boldsymbol{\theta}} \tag{4.5}$$

where $\bar{\boldsymbol{J}}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ is the extended Jacobian matrix defined as

$$\bar{\boldsymbol{J}} \triangleq \begin{bmatrix} \boldsymbol{J} \\ \boldsymbol{J_s} \end{bmatrix}. \tag{4.6}$$

The extended Jacobian matrix $\bar{\boldsymbol{J}}$ has full rank if and only if the Jacobian matrix $\boldsymbol{J}$ has full rank and the sub–task function $\boldsymbol{y_s}(t)$ is carefully defined in terms of the joint positions $\boldsymbol{\theta}$. It is highlighted that, inverse of the extended Jacobian matrix $\bar{\boldsymbol{J}}^{-1}$ exists $\forall \boldsymbol{\theta}$.

The design objectives are to make the end–effector position $\boldsymbol{x}$ go to a desired task–space position $\boldsymbol{x_d}(t) \in \mathbb{R}^m$, and to make sub–task function $\boldsymbol{y_s}$ go to some desired sub–task $\boldsymbol{y_d}(t) \in \mathbb{R}^{(n-m)}$. Therefore, an extended desired position, denoted by $\bar{\boldsymbol{x}}_{\boldsymbol{d}}(t) \in \mathbb{R}^n$, is defined as

$$\bar{\boldsymbol{x}}_{\boldsymbol{d}} \triangleq \begin{bmatrix} \boldsymbol{x_d} \\ \boldsymbol{y_d} \end{bmatrix}. \tag{4.7}$$

4.3. Error System Formulation

The tracking error, denoted by $\boldsymbol{e}(t) \in \mathbb{R}^n$, is defined as follows

$$\boldsymbol{e} \triangleq \bar{\boldsymbol{x}}_{\boldsymbol{d}} - \bar{\boldsymbol{x}}. \tag{4.8}$$

After taking the time derivative of (4.8) and substituting (4.5), the following expression is obtained

$$\dot{\boldsymbol{e}} = \dot{\bar{\boldsymbol{x}}}_{\boldsymbol{d}} - \bar{\boldsymbol{J}}\dot{\boldsymbol{\theta}}. \tag{4.9}$$

To further facilitate the subsequent analysis, an auxiliary error–like term, denoted by $\boldsymbol{r}(t) \in \mathbb{R}^n$, is defined as

$$\boldsymbol{r} \triangleq \bar{\boldsymbol{J}}^{-1}(\dot{\bar{\boldsymbol{x}}}_{\boldsymbol{d}} + \boldsymbol{\alpha}\boldsymbol{e}) - \dot{\boldsymbol{\theta}} \tag{4.10}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{n\times n}$ is a constant, positive–definite, diagonal, control gain matrix. Pre–multiplying (4.10) by $\bar{\boldsymbol{J}}$ and substituting (4.5) and (4.9), we obtain

$$\dot{\boldsymbol{e}} = -\boldsymbol{\alpha}\boldsymbol{e} + \bar{\boldsymbol{J}}\boldsymbol{r}. \tag{4.11}$$

Taking the time derivative of (4.10), pre–multiplying by generalized mass matrix $\boldsymbol{M}(\boldsymbol{\theta})$, and then substituting (2.33), the following open–loop error dynamics is obtained

$$\boldsymbol{M}\dot{\boldsymbol{r}} = \boldsymbol{Y}\boldsymbol{\phi} - \boldsymbol{C}\boldsymbol{r} - \boldsymbol{\tau} \tag{4.12}$$

where

$$\boldsymbol{Y}\boldsymbol{\phi} = \boldsymbol{M}\frac{d}{dt}\{\bar{\boldsymbol{J}}^{-1}(\dot{\bar{\boldsymbol{x}}}_{\boldsymbol{d}} + \boldsymbol{\alpha}\boldsymbol{e})\} + \boldsymbol{C}\bar{\boldsymbol{J}}^{-1}(\dot{\bar{\boldsymbol{x}}}_{\boldsymbol{d}} + \boldsymbol{\alpha}\boldsymbol{e}) + \boldsymbol{G} + \boldsymbol{F}_{\boldsymbol{d}}\dot{\boldsymbol{\theta}} \tag{4.13}$$

with $\boldsymbol{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \bar{\boldsymbol{x}}_{\boldsymbol{d}}, \dot{\bar{\boldsymbol{x}}}_{\boldsymbol{d}}, \ddot{\bar{\boldsymbol{x}}}_{\boldsymbol{d}}, \boldsymbol{e}, \dot{\boldsymbol{e}}, t) \in \mathbb{R}^{n\times p}$ denoting an available regression matrix, and $\boldsymbol{\phi} \in \mathbb{R}^p$ representing a constant parameter vector.

4.4. Adaptive Controller Design and Stability Analysis

In this section, the control design and the accompanying stability analysis will be presented. An adaptive controller is designed when parametric uncertainties exist in the dynamic model (*i.e.*, the parameter vector $\boldsymbol{\phi}$ in (4.13) is uncertain).

From the error system development in Section 4.3 and the subsequent stability analysis, the control input torque $\boldsymbol{\tau}(t)$ is designed as

$$\boldsymbol{\tau} = \boldsymbol{Y}\hat{\boldsymbol{\phi}} + \boldsymbol{K}_{\boldsymbol{r}}\boldsymbol{r} + \bar{\boldsymbol{J}}^T\boldsymbol{e} \tag{4.14}$$

where $\boldsymbol{K}_{\boldsymbol{r}} \in \mathbb{R}^{n\times n}$ is a constant, positive–definite, diagonal, control gain matrix, and $\hat{\boldsymbol{\phi}}(t) \in \mathbb{R}^p$ is the estimate of the uncertain parameter vector $\boldsymbol{\phi}$, and is updated according to

$$\dot{\hat{\boldsymbol{\phi}}} = \boldsymbol{\Gamma}\boldsymbol{Y}^T\boldsymbol{r} \tag{4.15}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times p}$ is a constant, positive–definite, diagonal, adaptation gain matrix. As can be seen from (4.14), pseudo–inverse of the Jacobian matrix is not utilized in the controller. The parameter estimation error vector $\widetilde{\boldsymbol{\phi}}(t) \in \mathbb{R}^p$ is defined as

$$\widetilde{\boldsymbol{\phi}} \triangleq \boldsymbol{\phi} - \hat{\boldsymbol{\phi}}. \tag{4.16}$$

Substituting (4.14) and (4.16) into (4.12), the closed–loop error system for $\boldsymbol{r}(t)$ can be obtained as

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{C}\boldsymbol{r} - \boldsymbol{K}_r\boldsymbol{r} - \bar{\boldsymbol{J}}^T\boldsymbol{e} + \boldsymbol{Y}\widetilde{\boldsymbol{\phi}}. \tag{4.17}$$

Now the stability analysis can be proceeded by introducing the following theorem.

Theorem 4.4.1 *The adaptive controller in* (4.14) *along with the parameter update law in* (4.15) *ensures asymptotic task–space tracking and asymptotic sub–task control in the sense that*

$$\|\boldsymbol{e}(t)\| \to 0 \text{ as } t \to \infty. \tag{4.18}$$

Proof The proof is started by defining a non–negative scalar function (*i.e.*, a Lyapunov function), denoted by $V(\boldsymbol{e}, \boldsymbol{r}, \widetilde{\boldsymbol{\phi}}) \in \mathbb{R}$, as

$$V \triangleq \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} + \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\boldsymbol{r} + \frac{1}{2}\widetilde{\boldsymbol{\phi}}^T\boldsymbol{\Gamma}^{-1}\widetilde{\boldsymbol{\phi}}. \tag{4.19}$$

By using (2.35), it can easily be shown that the following bounds hold for (4.19)

$$\lambda_1 \|\boldsymbol{z}\|^2 \leq V \leq \lambda_2 \|\boldsymbol{z}\|^2 \tag{4.20}$$

where $\lambda_1 \triangleq \frac{1}{2}\min\{1, m_1, \lambda_{\max}(\boldsymbol{\Gamma})\}$, $\lambda_2 \triangleq \frac{1}{2}\max\{1, m_2, \lambda_{\min}(\boldsymbol{\Gamma})\}$ and $\boldsymbol{z}(t) \in \mathbb{R}^{(2n+p)\times 1}$ is defined as

$$\boldsymbol{z}(t) \triangleq \begin{bmatrix} \boldsymbol{e}^T & \boldsymbol{r}^T & \widetilde{\boldsymbol{\phi}}^T \end{bmatrix}^T. \tag{4.21}$$

After taking the time derivative of (4.19), we obtain

$$\dot{V} = \boldsymbol{e}^T\dot{\boldsymbol{e}} + \boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + \widetilde{\boldsymbol{\phi}}^T\boldsymbol{\Gamma}^{-1}\dot{\widetilde{\boldsymbol{\phi}}}. \tag{4.22}$$

Substituting (4.9), (4.17), time derivative of (4.16) along with (4.15) into (4.22), and then canceling common terms, the following expression is obtained

$$\dot{V} = -\boldsymbol{e}^T\boldsymbol{\alpha}\boldsymbol{e} - \boldsymbol{r}^T\boldsymbol{K}_r\boldsymbol{r} \tag{4.23}$$

where (2.37) was also utilized. Notice that, the right–hand side of (4.23) can be upper bounded as

$$\dot{V} \leq -\lambda_3(\|\boldsymbol{e}\|^2 + \|\boldsymbol{r}\|^2) \tag{4.24}$$

where $\lambda_3 \triangleq \min\{\lambda_{\min}(\boldsymbol{\alpha}), \lambda_{\min}(\boldsymbol{K_r})\}$.

From (4.19), (4.20), (4.23) and (4.24), it is easy to see that $V(\boldsymbol{e}, \boldsymbol{r}, \widetilde{\boldsymbol{\phi}}) \in \mathcal{L}_\infty$, therefore, $\boldsymbol{e}(t)$, $\boldsymbol{r}(t)$ and $\widetilde{\boldsymbol{\phi}}(t) \in \mathcal{L}_\infty$. From the boundedness of $\widetilde{\boldsymbol{\phi}}(t)$ along with (4.16), it is clear that $\hat{\boldsymbol{\phi}}(t) \in \mathcal{L}_\infty$. Based on the boundedness of $\bar{\boldsymbol{x}}_{\boldsymbol{d}}(t)$, from (4.8), it is clear that $\bar{\boldsymbol{x}}(t) \in \mathcal{L}_\infty$. The boundedness of $\boldsymbol{e}(t)$ and $\boldsymbol{r}(t)$ can be utilized along with (4.11) to conclude that $\dot{\bar{\boldsymbol{x}}}(t) \in \mathcal{L}_\infty$. Above boundedness statements can be utilized with (4.10) to prove that $\dot{\boldsymbol{\theta}}(t) \in \mathcal{L}_\infty$. The boundedness of $\dot{\boldsymbol{\theta}}(t)$ can be utilized to prove that $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathcal{L}_\infty$. The above boundedness statements can be utilized with (4.12) to prove that $\dot{\boldsymbol{r}}(t) \in \mathcal{L}_\infty$. The robot manipulator dynamics in (2.33) can be utilized to demonstrate $\boldsymbol{\tau}(t) \in \mathcal{L}_\infty$. Standard signal chasing arguments can then be used to prove that all signals remain bounded under the closed–loop operation.

After integrating (4.24) in time from 0 to $+\infty$, the following expression is obtained

$$\int_0^{+\infty} \dot{V}(t)dt \leq -\lambda_3 \int_0^{+\infty} (\|\boldsymbol{e}(t)\|^2 + \|\boldsymbol{r}(t)\|^2)dt \tag{4.25}$$

and after evaluating the integral on the left hand–side, recalling $V(t) \geq 0$ and then rearranging yield

$$\int_0^{+\infty} (\|\boldsymbol{e}(t)\|^2 + \|\boldsymbol{r}(t)\|^2)dt \leq \frac{V(0)}{\lambda_3} \tag{4.26}$$

from which it can be seen that $\boldsymbol{e}(t)$ and $\boldsymbol{r}(t)$ are square integrable. Barbalat's Lemma in Krstic et al. (1995) can then be utilized to obtain asymptotic task–space tracking and asymptotic sub–task control as given in (4.18).

4.5. Experimental Studies

In order to demonstrate the performance of the proposed controller, an experimental study is conducted on the 3 dof robot manipulator given in Section 2.2.2.

The manipulator was initialized to be at rest at the joint position $\boldsymbol{\theta}(0) = [0, \pi/2, \pi/3]^T$ rad. The desired task–space trajectory was selected as

$$\boldsymbol{x_d} = \begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \begin{bmatrix} 0.017 + 0.02\sin(0.1t)(1 - \exp(-0.1t)) \\ 0.1905 - 0.02\cos(0.1t)(1 - \exp(-0.1t)) \end{bmatrix} \text{(m)}. \tag{4.27}$$

This experiment introduces a **novel** sub–task function (*i.e.,* laser/camera tracer) which aims to allow a perpendicularly fixed laser beam or optic camera on the middle of the first joint which traces line of the sight of the end–effector of the manipulator. This sub–task was motivated by the inspiring work of Buckingham and Graham (2005) where

two hyper redundant robot manipulators worked co–operatively to replace a section of a critical pipe in a nuclear reactor room where one of them was only utilized to hold a vision system at its end–effector. Via this sub–task, a similar task can be performed with only one hyper redundant robot. As represented in Figure 4.1, the sub–task function can be trigonometrically written as follows

$$y_s = \frac{l_1}{2} + l_2 \cos(\theta_2) + l_3 \cos(\theta_2 + \theta_3). \tag{4.28}$$

According to the aim of the sub–task function, the desired sub–task is to force y_s to go to zero, therefore $y_d = 0$.

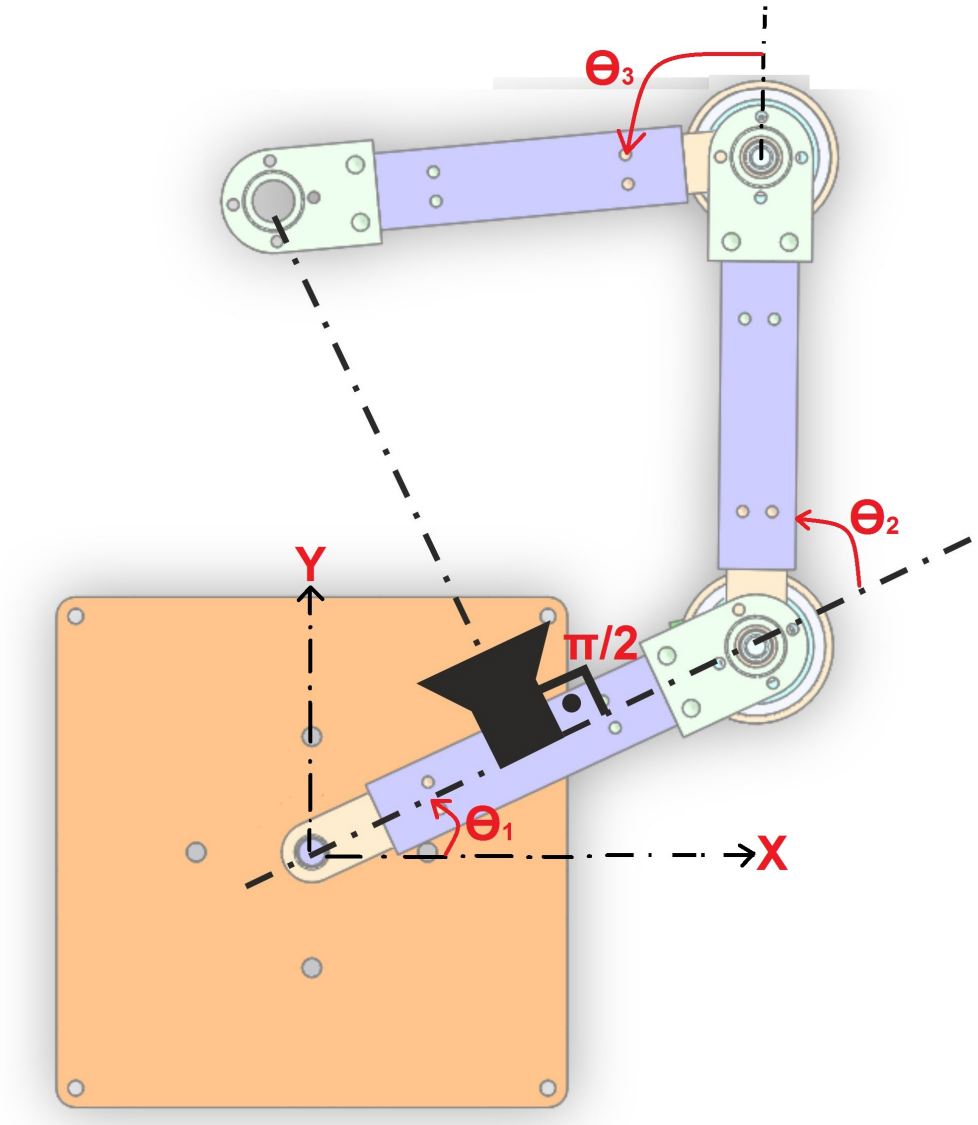

Figure 4.1. Illustration of the laser/camera tracer sub–task.

The control and adaptation gains were chosen as $\boldsymbol{\alpha} = 90 \times \text{diag}\ \{1.2, 1.1, 1\}$,

$\boldsymbol{K_r} = 45 \times \text{diag}\ \{1.2, 1.1, 1\}$ and $\boldsymbol{\Gamma} = 10^{-4} \times \text{diag}\ \{1, 0.1, 0.1, 1, 1, 1, 1\}$. In the experiments, the initial values of the parameter update vector were chosen as $\hat{\boldsymbol{\phi}}(0) = 10^{-3} \times [20, 2, 40, 20, 1, 10, 1]^T$.

Figures 4.2, 4.3, 4.4, 4.5 and 4.6 show the task–space tracking error, the control input torques, desired and actual task–space trajectories, the sub–task objective function, and the estimates of uncertain parameters, respectively. From Figures 4.2 and 4.4, it is clear that the task–space tracking objective was met and from Figure 4.5, it is clear that the sub–task function went to the desired value thus satisfying the sub–task objective in (4.28).

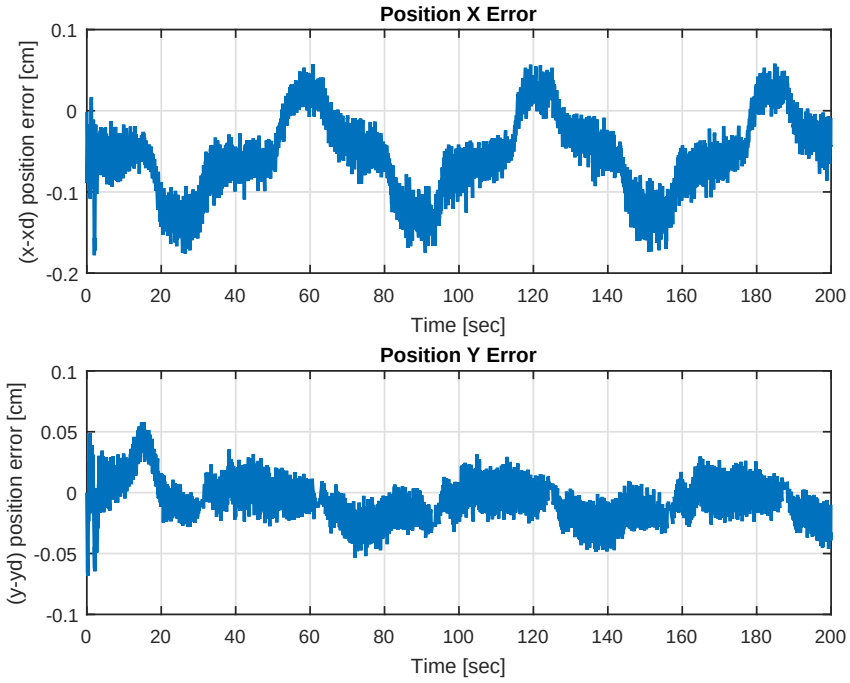

Figure 4.2. Task–space position tracking error $\boldsymbol{e}(t)$.

4.6. Conclusions

This chapter mainly focused on an extended task–space control design for kinematically redundant robot manipulators. Specifically, the main aim is to form the extended Jacobian by integrating the manipulator Jacobian matrix with the sub–task Jacobian matrix. Another motivation of this chapter is to propose a method that is applicable to hyper redundant robot manipulators without requiring a separate stability analysis for multiple

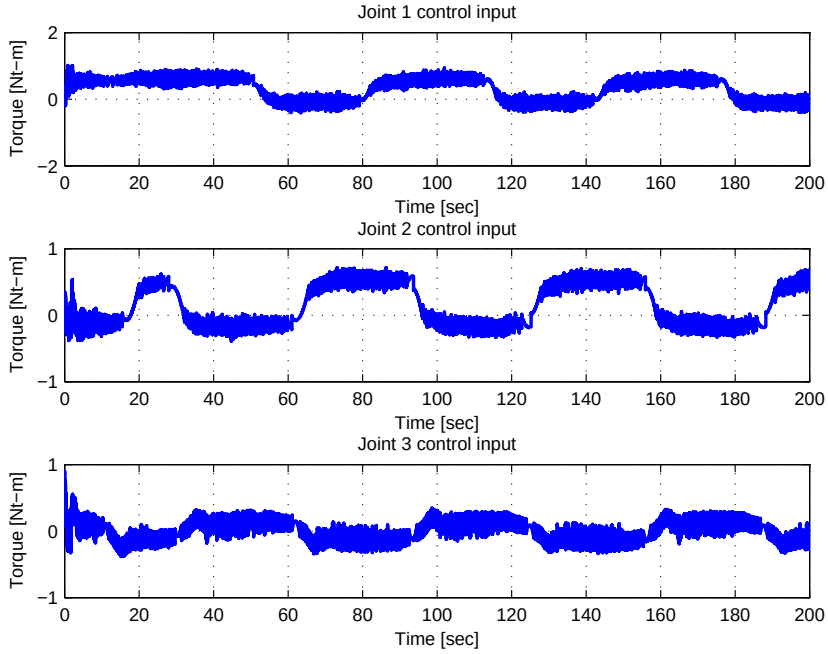

Figure 4.3. Control input torques $\boldsymbol{\tau}(t)$.

sub–task objectives. To deal with parametric uncertainties in robot dynamics, an adaptive controller was proposed. Lyapunov based stability analysis ensured stability and convergence of both task–space tracking and sub–task objectives. When compared to the existing literature on control of redundant robot manipulators, the proposed extended task–space controller is **novel** with two important specifications; the first one is that ensuring asymptotic stability of both task–space tracking and sub–task objectives, and the second one is that being applicable for multiple sub–tasks on hyper redundant robot manipulators. Experiments on a 3 dof planar redundant robot manipulator were conducted to demonstrate the performance of the proposed adaptive controller with a **novel** sub–task. In the sub–task objective, the redundant robot manipulator is considered as being equipped with a camera or a laser tracer on one of the link that traces the end–effector of the manipulator while performing the task–space tracking objective.

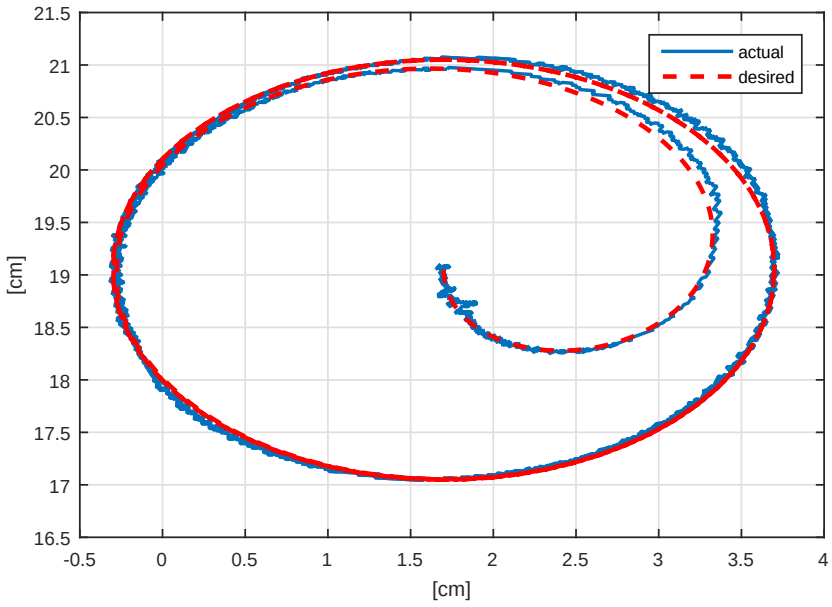

Figure 4.4. Desired $\boldsymbol{x_d}(t)$ and actual $\boldsymbol{x}(t)$ task–space trajectories.

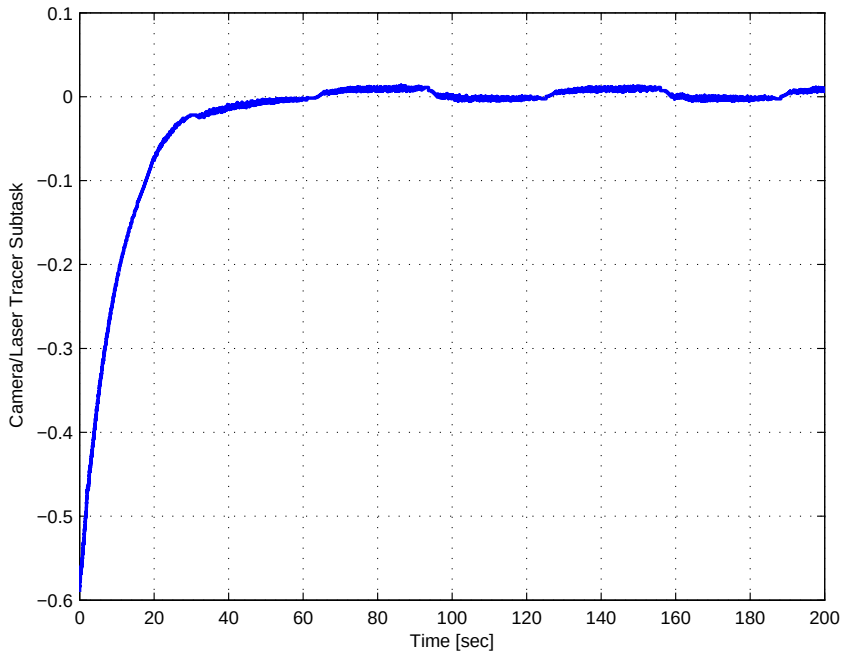

Figure 4.5. Sub–task function $\boldsymbol{y_s}(\boldsymbol{\theta})$.

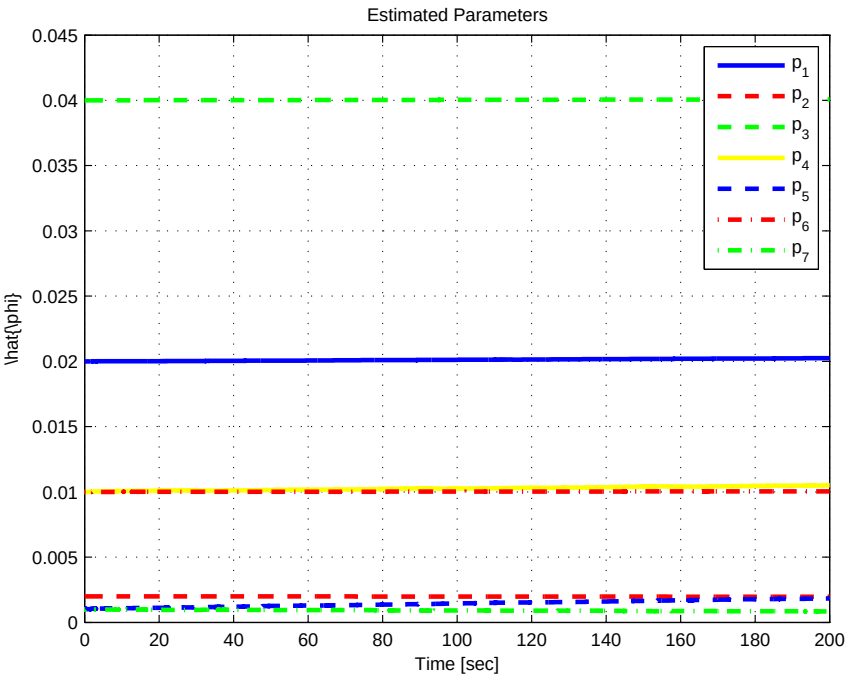

Figure 4.6. Estimates of uncertain parameters $\hat{\phi}(t)$.

CHAPTER 5

TASK–SPACE TRACKING CONTROL OF ROBOT MANIPULATORS WITH UNCERTAIN KINEMATICS AND DYNAMICS

This chapter considers the problem of task–space tracking control of robot manipulators with uncertain kinematics and dynamics. Specifically, a continuous robust task–space controller that achieves asymptotic tracking, despite the presence of unstructured uncertainties associated with the dynamical terms and parametric uncertainties with the velocity kinematics, is proposed. Experimental results are presented to illustrate the feasibility and performance of the proposed method.

During the subsequent design and stability analysis of the robust task–space controller, the dimension of the end–effector position and the number of joints will be considered same (*i.e.,* $m = n$). So the presentation is made for non–redundant robot manipulators. However, via utilizing the extended Jacobian method in Chapter 4, the proposed robust controller can easily be modified to be applicable to kinematically redundant robot manipulators as well.

5.1. Mathematical Models and Properties

In this chapter, the kinematic model of an n dof revolute joint robot manipulator given in (2.1) is utilized. The kinematic terms satisfy Properties 1 and 2. The dynamic model given in (2.33) is utilized. The dynamic terms satisfy Property 3.

5.2. Error System Formulation

The control objective is to ensure that the end–effector of the robot manipulator tracks a desired task–space trajectory. To quantify this objective, the task–space tracking error vector $\boldsymbol{e}(t) \in \mathbb{R}^n$ is defined as

$$\boldsymbol{e} \triangleq \boldsymbol{x_d} - \boldsymbol{x} \tag{5.1}$$

where $\boldsymbol{x_d}(t) \in \mathbb{R}^n$ denotes the desired task–space trajectory. After taking the time derivative of (5.1) and substituting (2.15), the following expression can be obtained

$$\dot{\boldsymbol{e}} = -\boldsymbol{\alpha e} + \hat{\boldsymbol{J}}\boldsymbol{r} - \widetilde{\boldsymbol{J}}\dot{\boldsymbol{\theta}} \tag{5.2}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{n\times n}$ denotes a constant, positive–definite, diagonal, control gain matrix and $\boldsymbol{r}(t) \in \mathbb{R}^n$ is an auxiliary error–like vector defined to have the following form

$$\boldsymbol{r} \triangleq \hat{\boldsymbol{J}}^{-1}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha e}) - \dot{\boldsymbol{\theta}} \tag{5.3}$$

where $\hat{\boldsymbol{J}}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ is the estimated Jacobian matrix defined as

$$\hat{\boldsymbol{J}} \triangleq \boldsymbol{J}|_{\boldsymbol{\phi_j}=\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}} \tag{5.4}$$

where $\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}(t) \in \mathbb{R}^p$ is the estimated parameter vector and $\widetilde{\boldsymbol{J}}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ is the difference between the Jacobian matrix and the estimated Jacobian matrix defined as

$$\widetilde{\boldsymbol{J}} \triangleq \boldsymbol{J} - \hat{\boldsymbol{J}}. \tag{5.5}$$

From (5.4) and (5.5), it is easy to see that $\hat{\boldsymbol{J}}\dot{\boldsymbol{\theta}} = \boldsymbol{W_j}\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}$ and $\widetilde{\boldsymbol{J}}\dot{\boldsymbol{\theta}} = \boldsymbol{W_j}\widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}}$ with $\widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}} \triangleq \boldsymbol{\phi_j} - \hat{\boldsymbol{\phi}}_{\boldsymbol{j}} \in \mathbb{R}^p$ being the parameter estimation error where $\boldsymbol{W_j}$ was introduced in (2.29).

To further facilitate the subsequent control development, another auxiliary error vector, denoted by $\boldsymbol{s}(t) \in \mathbb{R}^n$, is defined as

$$\boldsymbol{s} \triangleq \dot{\boldsymbol{r}} + \boldsymbol{\Gamma r} \tag{5.6}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$ denotes a constant, positive–definite, diagonal, gain matrix. Taking the time derivative of (5.6), substituting the second time derivative of (5.3) and then pre-multiplying by $\boldsymbol{M}(\boldsymbol{\theta})$ yields

$$\boldsymbol{M}\dot{\boldsymbol{s}} = \boldsymbol{M}\frac{d^2}{dt^2}\{\hat{\boldsymbol{J}}^{-1}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha e})\} - \dot{\boldsymbol{\tau}} + \dot{\boldsymbol{M}}\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{N}} + \boldsymbol{M\Gamma}\dot{\boldsymbol{r}} \tag{5.7}$$

where the time derivative of (2.33) was also utilized. After straightforward mathematical manipulations, the following expression is obtained

$$\boldsymbol{M}\dot{\boldsymbol{s}} = \boldsymbol{Q} - \frac{1}{2}\dot{\boldsymbol{M}}\boldsymbol{s} - \boldsymbol{r} - \dot{\boldsymbol{\tau}} \tag{5.8}$$

where $\boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{s}, t) \in \mathbb{R}^n$ is an auxiliary uncertain term defined as

$$\boldsymbol{Q} \triangleq \boldsymbol{M}\frac{d^2}{dt^2}\{\hat{\boldsymbol{J}}^{-1}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha e})\} + \dot{\boldsymbol{M}}\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{N}} + \boldsymbol{M\Gamma}\dot{\boldsymbol{r}} + \frac{1}{2}\dot{\boldsymbol{M}}\boldsymbol{s} + \boldsymbol{r}. \tag{5.9}$$

To facilitate the subsequent analysis, the desired form of $\boldsymbol{Q}$, denoted by $\boldsymbol{Q_d}$ $(\boldsymbol{x_d}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \dddot{\boldsymbol{x}}_{\boldsymbol{d}}) \in \mathbb{R}^n$, is defined as follows

$$\boldsymbol{Q_d} \triangleq \boldsymbol{Q}|_{\boldsymbol{x}=\boldsymbol{x_d}, \dot{\boldsymbol{x}}=\dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}=\ddot{\boldsymbol{x}}_{\boldsymbol{d}}}. \tag{5.10}$$

An auxiliary term, denoted by $\widetilde{\boldsymbol{Q}}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{s}, t) \in \mathbb{R}^n$, is defined as follows

$$\widetilde{\boldsymbol{Q}} \triangleq \boldsymbol{Q} - \boldsymbol{Q_d}. \tag{5.11}$$

As detailed in Appendix A.1, the difference between $\boldsymbol{Q}$ and $\boldsymbol{Q_d}$ can be upper bounded by functions of the error terms in the sense that

$$\left\| \widetilde{\boldsymbol{Q}} \right\| \leq \rho(\|\boldsymbol{z}\|) \, \|\boldsymbol{z}\| \tag{5.12}$$

where ρ is a non–negative non–decreasing bounding function of its argument, and $\boldsymbol{z}(t) \in \mathbb{R}^{3n}$ is the combined error vector defined as

$$\boldsymbol{z} \triangleq [\boldsymbol{e}^T, \boldsymbol{r}^T, \boldsymbol{s}^T]^T. \tag{5.13}$$

It should be noted that $\boldsymbol{Q}$, $\boldsymbol{Q_d}$ and $\widetilde{\boldsymbol{Q}}$ will be used only for the subsequent stability analysis, thus they are not required to be known.

5.3. Control Design

From the error system development in Section 5.2 and the subsequent stability analysis, the control input torque $\boldsymbol{\tau}(t)$ is designed as

$$\boldsymbol{\tau} = (\boldsymbol{K} + \boldsymbol{I_n})[\boldsymbol{r}(t) - \boldsymbol{r}(0) + \boldsymbol{\Gamma} \int_0^t \boldsymbol{r}(\sigma) d\sigma] + \boldsymbol{\Pi}(t) \tag{5.14}$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is a constant, positive–definite, diagonal control gain matrix, and $\boldsymbol{\Pi}(t) \in \mathbb{R}^n$ is an auxiliary term updated according to

$$\dot{\boldsymbol{\Pi}}(t) = \boldsymbol{\beta} \text{Sgn}(\boldsymbol{r}(t)) \text{ with } \boldsymbol{\Pi}(0) = \boldsymbol{0_{n \times 1}} \tag{5.15}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{n \times n}$ is a constant, positive–definite, diagonal control gain matrix, and $\text{Sgn}(\cdot) \in \mathbb{R}^n$ is the vector signum function. It is noted that the term $\boldsymbol{r}(0)$ is introduced in (5.14) to satisfy $\boldsymbol{\tau}(0) = \boldsymbol{0_{n \times 1}}$. The controller in (5.14) and (5.15) requires $\boldsymbol{r}(t)$ only which has the form $\boldsymbol{r} = \hat{\boldsymbol{J}}(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}}_{\boldsymbol{j}})^{-1}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{\alpha}(\boldsymbol{x_d} - \boldsymbol{x})) - \dot{\boldsymbol{\theta}}$ and can be calculated via measurements of $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$, and $\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}$ which is updated according to

$$\dot{\hat{\boldsymbol{\phi}}}_{\boldsymbol{j}} = \text{proj}\,\{\boldsymbol{\mu}\} \tag{5.16}$$

where the auxiliary term $\boldsymbol{\mu} \in \mathbb{R}^p$ is defined as

$$\boldsymbol{\mu} \triangleq \boldsymbol{\Gamma_j} \boldsymbol{W_j^T} \boldsymbol{e} \tag{5.17}$$

where $\boldsymbol{\Gamma_j} \in \mathbb{R}^{p\times p}$ is a constant, positive–definite, diagonal matrix and the projection of the i^{th} entry of $\boldsymbol{\mu}$ (*i.e.*, μ_i) is designed as follows

$$\text{proj}\left\{\mu_i\right\} = \begin{cases} \mu_i & \text{if} \quad \hat{\phi}_{ji} > \underline{\phi}_{ji} \\ \mu_i & \text{if} \quad \hat{\phi}_{ji} = \underline{\phi}_{ji} \quad \text{and} \quad \mu_i > 0 \\ 0 & \text{if} \quad \hat{\phi}_{ji} = \underline{\phi}_{ji} \quad \text{and} \quad \mu_i < 0 \\ 0 & \text{if} \quad \hat{\phi}_{ji} = \overline{\phi}_{ji} \quad \text{and} \quad \mu_i > 0 \\ \mu_i & \text{if} \quad \hat{\phi}_{ji} = \overline{\phi}_{ji} \quad \text{and} \quad \mu_i \leq 0 \\ \mu_i & \text{if} \quad \hat{\phi}_{ji} < \overline{\phi}_{ji} \end{cases} \tag{5.18}$$

where $\hat{\phi}_{ji}(t)$ denotes the i^{th} component of $\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}(t)$. The projection algorithm ensures that lower and upper bounds of the estimated parameter vector satisfy $\underline{\boldsymbol{\phi}}_{\boldsymbol{j}} \leq \hat{\boldsymbol{\phi}}_{\boldsymbol{j}}(t) \leq \overline{\boldsymbol{\phi}}_{\boldsymbol{j}}$ provided that $\underline{\phi}_{ji} \leq \hat{\phi}_{ji}(0) \leq \overline{\phi}_{ji}\ \forall i = 1, 2, ..., p$ is satisfied Krstic et al. (1995), Braganza et al. (2008).

After substituting (5.11) and the time derivative of (5.14) into (5.8), the closed–loop error system for $\boldsymbol{s}(t)$ can be written as follows

$$\boldsymbol{M}\dot{\boldsymbol{s}} = \boldsymbol{Q_d} + \widetilde{\boldsymbol{Q}} - \frac{1}{2}\dot{\boldsymbol{M}}\boldsymbol{s} - \boldsymbol{r} - (\boldsymbol{K} + \boldsymbol{I_n})\boldsymbol{s} - \boldsymbol{\beta}\text{Sgn}(\boldsymbol{r}) \tag{5.19}$$

where (5.6) and (5.15) were also utilized. At this stage, the stability analysis can be proceeded.

5.4. Stability Analysis

The stability analysis can be proceeded by introducing the following theorem.

Theorem 5.4.1 *The controller in* (5.14) *and* (5.15) *and the adaptive update law in* (5.16) *ensures asymptotic task–space tracking in the sense that*

$$\|\boldsymbol{e}(t)\| \to 0 \text{ as } t \to \infty \tag{5.20}$$

provided that the controller gains are selected to satisfy

$$\beta_i \geq |Q_{di}(t)| + \frac{1}{\Gamma_i}\left|\dot{Q}_{di}(t)\right| \ \forall t \tag{5.21}$$

$$\lambda_{\min}(\boldsymbol{\alpha}) > \frac{\xi_{\hat{\boldsymbol{J}}}}{2} \tag{5.22}$$

$$\lambda_{\min}(\boldsymbol{\Gamma}) > \frac{\xi_{\hat{\boldsymbol{J}}}}{2} \tag{5.23}$$

and the entries of $\boldsymbol{K}$ *are chosen sufficiently large compared to the initial conditions of the system. In* (5.21), (5.22), (5.23), β_i, $\Gamma_i \in \mathbb{R}$ *denotes the i–th diagonal entry of* $\boldsymbol{\beta}$

and $\boldsymbol{\Gamma}$*, respectively, and* $Q_{di}(t)$ *and* $\dot{Q}_{di}(t)$ *denote the i–th entries of* $\boldsymbol{Q_d}(t)$ *and* $\dot{\boldsymbol{Q}}_{\boldsymbol{d}}(t)$*, respectively, and* $\xi_{\hat{\boldsymbol{J}}} \in \mathbb{R}$ *is a positive bounding constant satisfying* $\xi_{\hat{\boldsymbol{J}}} \geq \left\| \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \right\|_{i\infty} \forall \boldsymbol{\theta}$*.*

Proof The proof is started by defining a non–negative scalar function (*i.e.*, a Lyapunov function), denoted by $V(\bar{\boldsymbol{z}}, t) \in \mathbb{R}$, as

$$V \triangleq \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} + \frac{1}{2}\boldsymbol{r}^T\boldsymbol{r} + \frac{1}{2}\boldsymbol{s}^T\boldsymbol{M}\boldsymbol{s} + P + \frac{1}{2}\widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}}^T\boldsymbol{\Gamma}_{\boldsymbol{j}}^{-1}\widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}} \tag{5.24}$$

where $P(t) \in \mathbb{R}$ is an auxiliary function defined as Xian et al. (2004)(b)

$$P \triangleq \zeta_P - \int_0^t \eta(\sigma)d\sigma \tag{5.25}$$

where $\eta(t)$, $\zeta_P \in \mathbb{R}$ are defined as follows

$$\eta(t) \triangleq \boldsymbol{s}^T(t)[\boldsymbol{Q_d}(t) - \boldsymbol{\beta}\text{Sgn}(\boldsymbol{r}(t))] \tag{5.26}$$

$$\zeta_P \triangleq \sum_{i=1}^{n} \beta_i \left| r_i(0) \right| - \boldsymbol{r}^T(0)\boldsymbol{Q_d}(0) \tag{5.27}$$

and $\bar{\boldsymbol{z}}(t) \in \mathbb{R}^{(3n+p+1)}$ is defined as

$$\bar{\boldsymbol{z}}(t) \triangleq \left[\begin{array}{cccc} \boldsymbol{e}^T & \boldsymbol{r}^T & \boldsymbol{s}^T & \sqrt{P} \quad \widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}}^T \end{array} \right]^T. \tag{5.28}$$

Based on the proof in Appendix A.2, it is clear that $P(t) \geq 0$ and thus $V(\bar{\boldsymbol{z}}, t)$ is a Lyapunov function.

Note that, (5.24) can be lower and upper bounded as follows

$$\lambda_1 \left\| \bar{\boldsymbol{z}} \right\|^2 \leq V(\bar{\boldsymbol{z}}) \leq \lambda_2 \left\| \bar{\boldsymbol{z}} \right\|^2 \tag{5.29}$$

where λ_1, $\lambda_2 \in \mathbb{R}$ are positive bounding constants defined as

$$\lambda_1 \triangleq \frac{1}{2}\min\{1, m_1, \lambda_{\max}(\boldsymbol{\Gamma_j})\} \text{ , } \lambda_2 \triangleq \max\{1, \frac{1}{2}m_2, \lambda_{\min}(\boldsymbol{\Gamma_j})\}. \tag{5.30}$$

Taking the time derivative of (5.24) yields

$$\dot{V} = \boldsymbol{e}^T\dot{\boldsymbol{e}} + \boldsymbol{r}^T\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{s}^T\dot{\boldsymbol{M}}\boldsymbol{s} + \boldsymbol{s}^T\boldsymbol{M}\dot{\boldsymbol{s}} + \dot{P} + \widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}}^T\boldsymbol{\Gamma}_{\boldsymbol{j}}^{-1}\dot{\widetilde{\boldsymbol{\phi}}}_{\boldsymbol{j}}. \tag{5.31}$$

Substituting (5.2), (5.6), (5.16), (5.19), and the time derivative of (5.25) into (5.31), and canceling common terms, the following expression can be obtained

$$\dot{V} = -\boldsymbol{e}^T\boldsymbol{\alpha}\boldsymbol{e} + \boldsymbol{e}^T\hat{\boldsymbol{J}}\boldsymbol{r} - \boldsymbol{r}^T\boldsymbol{\Gamma}\boldsymbol{r} + \boldsymbol{s}^T\widetilde{\boldsymbol{Q}} - \boldsymbol{s}^T(\boldsymbol{K} + \boldsymbol{I_n})\boldsymbol{s}. \tag{5.32}$$

After utilizing (5.12), following upper bound for the right–hand side of (5.32) is obtained

$$\begin{aligned} \dot{V} \leq & -\lambda_{\min}(\boldsymbol{\alpha}) \left\| \boldsymbol{e} \right\|^2 + \frac{\xi_{\hat{\boldsymbol{J}}}}{2} \left\| \boldsymbol{e} \right\|^2 + \frac{\xi_{\hat{\boldsymbol{J}}}}{2} \left\| \boldsymbol{r} \right\|^2 - \lambda_{\min}(\boldsymbol{\Gamma}) \left\| \boldsymbol{r} \right\|^2 - \left\| \boldsymbol{s} \right\|^2 + \rho \left\| \boldsymbol{s} \right\| \left\| \boldsymbol{z} \right\| \\ & -\lambda_{\min}(\boldsymbol{K}) \left\| \boldsymbol{s} \right\|^2 \end{aligned} \tag{5.33}$$

where $\xi_{\hat{\boldsymbol{J}}} > \left\| \hat{\boldsymbol{J}} \right\|_{i\infty}$ was also utilized. Note that the last two terms of (5.33) can be upper bounded as

$$\rho \left\| \boldsymbol{s} \right\| \left\| \boldsymbol{z} \right\| - \lambda_{\min}(\boldsymbol{K}) \left\| \boldsymbol{s} \right\|^2 \leq \frac{\rho^2}{4\lambda_{\min}(\boldsymbol{K})} \left\| \boldsymbol{z} \right\|^2 \tag{5.34}$$

and in view of this inequality, the right–hand side of (5.33) can further be upper bounded as

$$\dot{V} \leq -[\min\{(\lambda_{\min}(\boldsymbol{\alpha}) - \frac{\xi_{\hat{\boldsymbol{J}}}}{2}), (\lambda_{\min}(\boldsymbol{\Gamma}) - \frac{\xi_{\hat{\boldsymbol{J}}}}{2}), 1\} - \frac{\rho^2}{4\lambda_{\min}(\boldsymbol{K})}] \left\| \boldsymbol{z} \right\|^2 . \tag{5.35}$$

Provided that (5.22) and (5.23) are satisfied, and the entries of $\boldsymbol{K}$ are chosen sufficiently large when compared to the initial conditions of the system, following expression can be obtained

$$\dot{V} \leq -\lambda \left\| \boldsymbol{z} \right\|^2 \tag{5.36}$$

for some $0 < \lambda < 1$.

From (5.24) and (5.36), it can be ensured that $V(\bar{\boldsymbol{z}}, t) \in \mathcal{L}_\infty$. Therefore, $\bar{\boldsymbol{z}}(t) \in \mathcal{L}_\infty$, and thus, based on its definition in (5.28), $\boldsymbol{e}(t)$, $\boldsymbol{r}(t)$, $\boldsymbol{s}(t)$, $\widetilde{\boldsymbol{\phi}}_{\boldsymbol{j}}(t) \in \mathcal{L}_\infty$. Based on the boundedness of the desired task–space trajectory, from (5.1), it is clear that $\boldsymbol{x}(t) \in \mathcal{L}_\infty$. The boundedness of $\boldsymbol{e}(t)$ and $\boldsymbol{r}(t)$ can be utilized along with (5.2) to conclude that $\dot{\boldsymbol{x}}(t) \in \mathcal{L}_\infty$. Above boundedness statements can be utilized with (5.3) to prove that $\dot{\boldsymbol{\theta}}(t) \in \mathcal{L}_\infty$. The boundedness of $\dot{\boldsymbol{\theta}}(t)$ can be utilized to prove that $\boldsymbol{W}_{\boldsymbol{j}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathcal{L}_\infty$. Above boundedness statements can be utilized with (5.16) to prove that $\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}(t) \in \mathcal{L}_\infty$. From (5.6), it is clear that $\dot{\boldsymbol{r}}(t) \in \mathcal{L}_\infty$, which can be utilized along with the time derivatives of (5.2) and (5.3) to show that $\ddot{\boldsymbol{\theta}}(t)$, $\ddot{\boldsymbol{e}}(t) \in \mathcal{L}_\infty$, respectively. The above boundedness statements can be utilized along with (5.19) to prove that $\dot{\boldsymbol{s}}(t) \in \mathcal{L}_\infty$. The robot manipulator dynamic model in (2.33) can be utilized to demonstrate $\boldsymbol{\tau}(t) \in \mathcal{L}_\infty$. Standard signal chasing arguments can then be used to prove that all signals remain bounded under the closed–loop operation.

Integrating the inequality in (5.36) in time from 0 to $+\infty$ results

$$\int_0^{+\infty} \left\| \boldsymbol{z}(t) \right\|^2 dt \leq \frac{V(0)}{\lambda} \tag{5.37}$$

from which, it is easy to see that $\boldsymbol{z}(t)$ is square integrable. Finally, since $\boldsymbol{z}(t) \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ and $\dot{\boldsymbol{z}}(t) \in \mathcal{L}_\infty$, utilizing Barbalat's Lemma in Krstic et al. (1995) yields $\left\| \boldsymbol{z}(t) \right\| \rightarrow 0$ as $t \rightarrow \infty$ from which the asymptotic tracking result given in (5.20) follows.

5.5. Experimental Study

To illustrate the performance of the proposed controller, experimental studies are conducted on PHANToM Omni haptic device.

In view of (2.29), from (2.46), the regression matrix $\boldsymbol{W_j} \in \mathbb{R}^{3\times 2}$ is obtained as

$$\boldsymbol{W_j} = \begin{bmatrix} -c_1c_2\dot{\theta}_1 + s_1s_2\dot{\theta}_2 & -c_1s_3\dot{\theta}_1 - s_1c_3\dot{\theta}_3 \\ c_2\dot{\theta}_2 & s_3\dot{\theta}_3 \\ -s_1c_2\dot{\theta}_1 - c_1s_2\dot{\theta}_2 & -s_1s_3\dot{\theta}_1 + c_1c_3\dot{\theta}_3 \end{bmatrix} \tag{5.38}$$

with $\boldsymbol{\phi_j} = [l_1, l_2]^T \in \mathbb{R}^2$.

During the experimental studies the desired task–space trajectory is chosen as

$$\boldsymbol{x_d}(t) = \begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix} = \begin{bmatrix} 0.05(1 - \exp(-0.05t)) \\ -0.01 - 0.02\cos(0.05t) \\ -0.01 - 0.02\sin(0.05t) \end{bmatrix} \text{(m)}. \tag{5.39}$$

The control gains were chosen as $\boldsymbol{\alpha} = \text{diag}\{40, 30, 20\}$, $\boldsymbol{\beta} = 0.1\boldsymbol{I_3}$, $\boldsymbol{\Gamma} = 0.1\boldsymbol{I_3}$, $\boldsymbol{K} = \text{diag}\{0.12, 0.03, 0.02\}$, and $\boldsymbol{\Gamma_j} = 2\boldsymbol{I_2}$. The manipulator was initialized to be at rest at the following joint positions $\boldsymbol{\theta}(0) = [0, 0.58, 0.36]^T$rad. In the experiment, the entries of the constant parameter vector $\boldsymbol{\phi_j} \in \mathbb{R}^2$ are assumed uncertain and the initial values were chosen as $\hat{\boldsymbol{\phi}}_{\boldsymbol{j}}(0) = [0.1, 0.1]^T$. Full state feedback control was shortly named as FSFB control in the experiment results.

The results of the experimental study are presented in Figures 5.1–5.4. Figure 5.1 presents the task–space tracking error. In Figure 5.2, desired and actual task–space trajectories are presented on X, Y and Z axes. In Figure 5.3, control input torques are presented. Figure 5.4 presents the estimates of uncertain parameters. From Figures 5.1 and 5.2, it is clear that the tracking control objective was met in the experimental study.

5.6. Conclusions

This chapter presents a **novel** task–space controller formulation for robot manipulators. The proposed robust task–space controller ensured asymptotic end–effector tracking despite unstructured uncertainties in the dynamics and structured uncertainties in the velocity kinematics. As opposed to most robust or variable structure controllers presented in the literature, the proposed robust controller was continuous, and asymptotic tracking was ensured via this continuous actuation. The stability of the proposed controller was

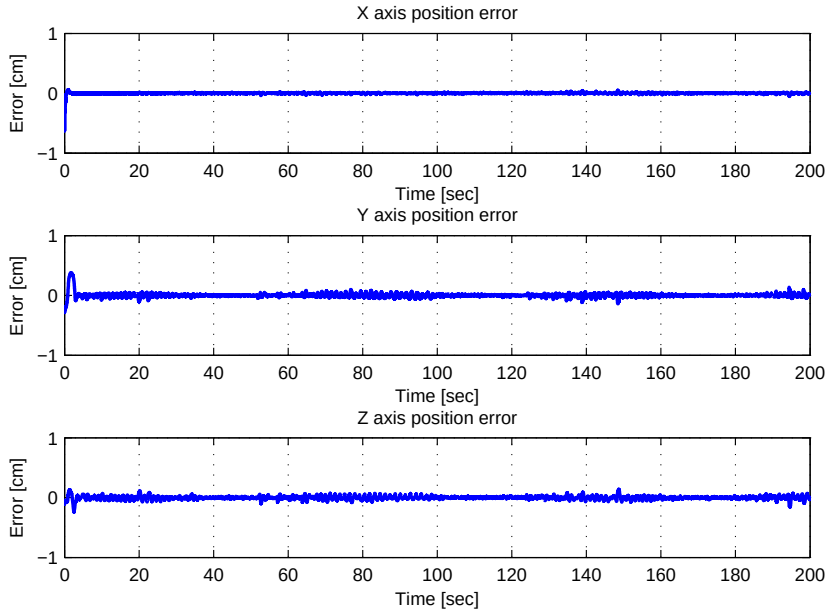

Figure 5.1. Experiment of Robust FSFB control: Task–space tracking error $\boldsymbol{e}(t)$.

ensured via rigorous theoretical analysis based on Lyapunov techniques. Experimental studies performed on PHANToM Omni Haptic device confirmed the performance of the proposed robust controller.

When compared to the previous adaptive task–space controllers in Zergeroglu et al. (2000), Zergeroglu et al. (2004), Xian et al. (2004)(a), Tatlicioglu et al. (2008)(a), the proposed controller can compensate for a broader class of uncertainties. And when compared to the robust controller of Ozbay et al. (2008), asymptotic task–space tracking is achieved as opposed to ultimate boundedness of the end–effector tracking error.

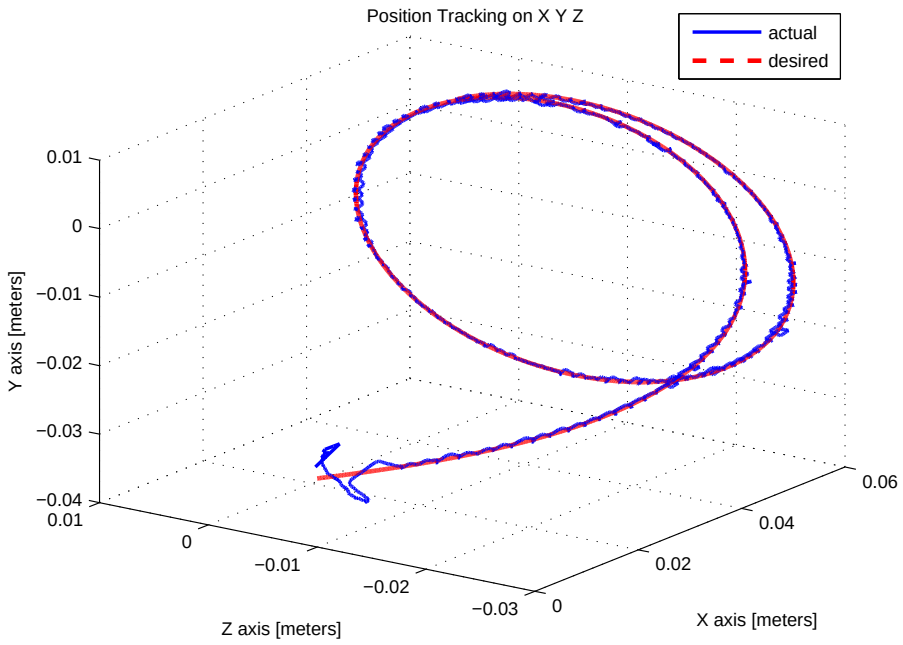

Figure 5.2. Experiment of Robust FSFB control: Desired and actual task–space trajectories.

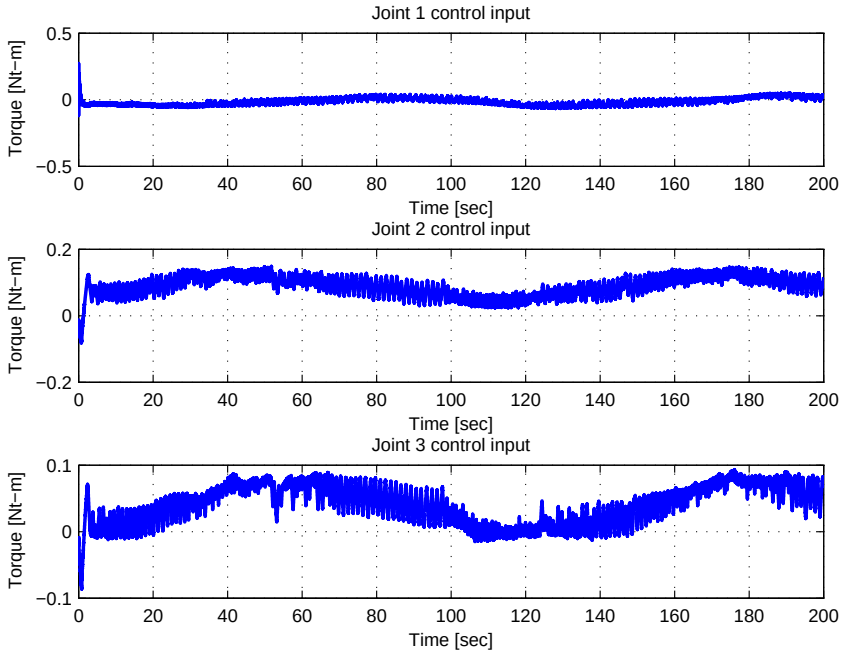

Figure 5.3. Experiment of Robust FSFB control: Control input torques $\boldsymbol{\tau}(t)$.

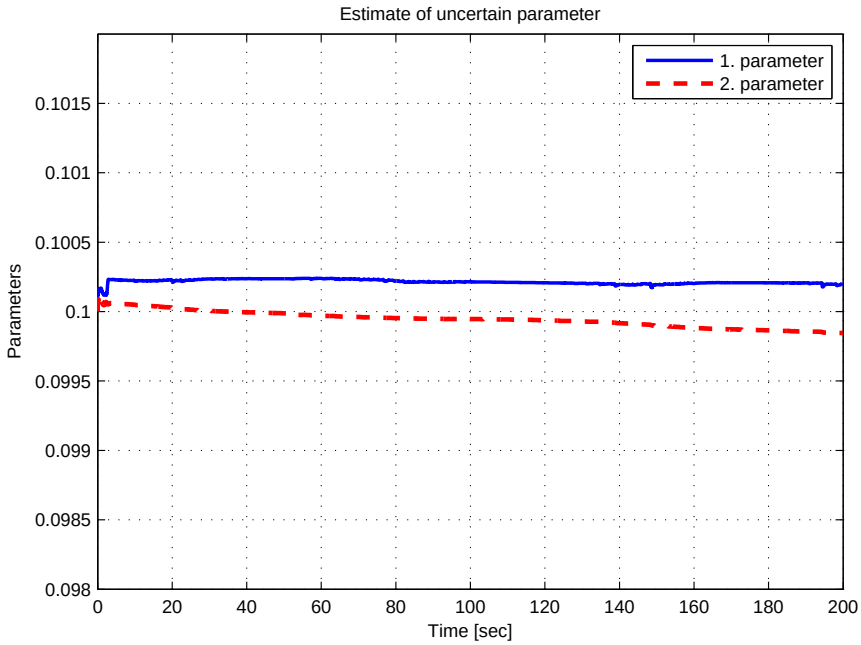

Figure 5.4. Experiment of Robust FSFB control: Estimates of uncertain parameters $\hat{\phi}_j(t)$.

CHAPTER 6

ROBUST OUTPUT FEEDBACK APPROACH FOR TASK–SPACE TRACKING CONTROL OF ROBOT MANIPULATORS

In this chapter, task–space tracking control of robot manipulators subject to uncertainties in kinematic and dynamic models is examined. The control problem is further complicated by the lack of joint velocity measurements. A robust observer based output feedback controller is proposed. While requiring only the measurement of the end–effector position of the robot manipulator, the proposed controller ensures uniformly ultimately bounded tracking result that can be adjusted arbitrarily small with high gain and compensates for both structured and unstructured uncertainties in kinematic and dynamic models of the manipulator. Simulation and experimental results are presented to demonstrate the validity of the proposed controller.

During the subsequent developments of the robust output feedback controller, the dimension of the end–effector position and the number of joints will be considered same (*i.e.,* $m = n$). So the presentation is made for non–redundant robot manipulators. However, utilizing the extended Jacobian method in Chapter 4, the proposed robust output feedback controller can easily be modified to be applicable to kinematically redundant robot manipulators as well.

6.1. Mathematical Models and Properties

In this chapter, the kinematic model of an n dof revolute joint robot manipulator given in (2.1) is utilized. Both the forward kinematics $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and the manipulator Jacobian matrix $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n}$ are considered to be uncertain. The dynamic model given in (2.33) is utilized with Property 3.

Premultiplying (2.33) by the inverse of the generalized mass matrix, joint acceleration vector can be obtained as

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{M}^{-1}\boldsymbol{\tau} - \boldsymbol{M}^{-1}\boldsymbol{N}. \tag{6.1}$$

Taking the time derivative of (2.15), the end–effector acceleration vector can be obtained

as

$$\ddot{\boldsymbol{x}} = \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}} + \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{\tau} - \boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{N} \tag{6.2}$$

where (6.1) was utilized. In (6.2), the control input $\boldsymbol{\tau}(t)$ is multiplied with $\boldsymbol{J}\boldsymbol{M}^{-1}$ where both $\boldsymbol{J}$ and $\boldsymbol{M}$ are considered as uncertain.

For (6.2), in the literature, usually, a control design with known Jacobian is preferred where researchers preferred to design $\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{u}$ with $\boldsymbol{u}$ being an auxiliary controller that results $\boldsymbol{u}$ to be premultiplied with $\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T$ which is both positive–definite and symmetric. When compared with similar works in the literature, in this chapter, both $\boldsymbol{J}$ and $\boldsymbol{M}$ being uncertain makes the control problem harder. This chapter deals with this problem by utilizing the following matrix decomposition Costa et al. (2003)

$$\boldsymbol{J}\boldsymbol{M}^{-1} = \boldsymbol{S}_{\boldsymbol{M}}\boldsymbol{D}\boldsymbol{U} \tag{6.3}$$

where $\boldsymbol{S}_{\boldsymbol{M}} \in \mathbb{R}^{n\times n}$ is a positive–definite and symmetric matrix, $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with entries ± 1 and $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is a unity upper triangular matrix. It is noted that the $\boldsymbol{S}_{\boldsymbol{M}}\boldsymbol{D}\boldsymbol{U}$ decomposition approach applied to several models in the literature and the decomposition may result in $\boldsymbol{D}$ being an identity matrix.

After premultiplying (6.3) with $\bar{\boldsymbol{M}}(\boldsymbol{\theta}) \triangleq \boldsymbol{S}_{\boldsymbol{M}}^{-1}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$, the following expression is obtained as

$$\bar{\boldsymbol{M}}\ddot{\boldsymbol{x}} = \bar{\boldsymbol{M}}\dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}} - \bar{\boldsymbol{M}}\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{N} + \boldsymbol{D}\boldsymbol{U}\boldsymbol{\tau}. \tag{6.4}$$

It is noticed that, since $\boldsymbol{S}_{\boldsymbol{M}}$ is positive–definite and symmetric, $\bar{\boldsymbol{M}}$ is also positive–definite and symmetric.

6.2. Observer and Controller Design

The control objective is to ensure that the end–effector of the robot manipulator tracks a desired task–space trajectory. The error system development and the subsequent control design will be restricted by the availability of only $\boldsymbol{x}(t)$ which is assumed to be available via some external measurement equipment such as laser interferometer, theodolite, opto-camera systems Kam Lau and Haynes (1985), Zobrist and Ho (1996).

To quantify the control objective, the task–space tracking error $\boldsymbol{e}(t) \in \mathbb{R}^n$ is defined as

$$\boldsymbol{e} \triangleq \boldsymbol{x}_{\boldsymbol{d}} - \boldsymbol{x} \tag{6.5}$$

where $\boldsymbol{x}_{\boldsymbol{d}}(t) \in \mathbb{R}^n$ denotes the desired task–space trajectory. An auxiliary error–like term,

denoted by $\boldsymbol{r}(t) \in \mathbb{R}^n$, is defined as

$$\boldsymbol{r} = \dot{\boldsymbol{e}} + 2\boldsymbol{e}. \tag{6.6}$$

The control input torque $\boldsymbol{\tau}(t)$ is designed as

$$\boldsymbol{\tau} = \boldsymbol{D}\text{Sat}\left(\boldsymbol{K}\hat{\boldsymbol{r}}\right) + \boldsymbol{D}\boldsymbol{\tau}_{\boldsymbol{ff}} \tag{6.7}$$

where $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is a constant, positive–definite, diagonal, control gain matrix, $\boldsymbol{\tau}_{\boldsymbol{ff}}(t) \in \mathbb{R}^n$ is the feed–forward component of the control input, $\text{Sat}\left(\cdot\right) \in \mathbb{R}^n$ is the vector saturation function, which is used to keep the control input bounded, and $\hat{\boldsymbol{r}}(t) \in \mathbb{R}^n$ is the observer for the auxiliary error–like term $\boldsymbol{r}(t)$ which is updated as

$$\dot{\hat{\boldsymbol{r}}} = \frac{1}{\epsilon^2}\boldsymbol{\alpha_2}(\boldsymbol{e} - \hat{\boldsymbol{e}}) \tag{6.8}$$

in which $\hat{\boldsymbol{e}}(t) \in \mathbb{R}^n$ is the observer for the task–space tracking error $\boldsymbol{e}(t)$ which is updated as

$$\dot{\hat{\boldsymbol{e}}} = \hat{\boldsymbol{r}} - 2\hat{\boldsymbol{e}} + \frac{1}{\epsilon}\boldsymbol{\alpha_1}(\boldsymbol{e} - \hat{\boldsymbol{e}}) \tag{6.9}$$

where $\epsilon \in \mathbb{R}$ is a small positive constant, $\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2} \in \mathbb{R}^{n\times n}$ are constant, positive–definite, diagonal observer gain matrices.

Theorem 6.2.1 *The controller in* (6.7) *yields a uniformly ultimately bounded tracking result in the following sense*

$$\left\|\boldsymbol{e}\left(t\right)\right\| \leq \varepsilon \tag{6.10}$$

where ε is a small constant whose value can be adjusted via changing the control gain $\boldsymbol{K}$.

Proof The reader is referred to Chen et al. (2008) for a detailed stability analysis of the full–state feedback version of the controller, and this analysis can be utilized in conjunction with the high–gain observer analysis in Chapter 12 of Khalil (2015) to prove the ultimate boundedness of task–space tracking error and observer errors.

6.3. Simulation Study

To demonstrate the performance of the proposed controller, a numerical simulation was performed by utilizing the model of the PHANToM Omni haptic device given in Section 2.2.1. The haptic device is considered to be working as a 2 dof planar manipulator with two links l_1, l_2 and two joints θ_2, θ_3 moving in Y and Z axes as shown in Figure 2.2.

The desired task–space trajectory was chosen as

$$\boldsymbol{x_d}(t) = \begin{bmatrix} Y_d \\ Z_d \end{bmatrix} = \begin{bmatrix} -0.08 - 0.02\cos(0.1t) \\ -0.04 + 0.02\sin(0.1t) \end{bmatrix} \text{(m)}. \tag{6.11}$$

The simulation studies were run on MATLAB Simulink with a data rate of 1000Hz. The control gain matrix was chosen as $\boldsymbol{K} = \text{diag}\{4; 3\}$ and no feed–forward compensation term was utilized. The observer gains were chosen as $\boldsymbol{\alpha_1} = \boldsymbol{\alpha_2} = 10\boldsymbol{I_2}$ and $\epsilon = 0.01$. The maximum and minimum values for the saturation function in the control input were set at ± 0.4 Nt-m. Output feedback control was shortly named as OFB control in the numerical results.

The simulation results are presented in Figures 6.1–6.3. In Figure 6.1, desired and actual task–space trajectories are presented. Figure 6.2 presents task–space tracking error versus observed task–space tracking error. In Figure 6.3, control input torques are presented. From Figure 6.1, it is clear that the tracking control objective was met and from Figure 6.2, it is seen that observer objective is achieved.

6.4. Experimental Study

To demonstrate the validity of the proposed controller, experimental studies with PHANToM Omni haptic device shown in Figure 2.2 were performed. In the experiments, the desired task–space trajectory in (6.11) was utilized. In addition, since the device is utilized as a 2 dof planar manipulator, the two links with the last two joints of the device were used and Y and Z axes measurements of the end–effector position were used in the experiments. The experimental studies run on MATLAB Simulink with a data rate of 100Hz. The control gain matrix was chosen as $\boldsymbol{K} = \text{diag}\{30; 25\}$ for the experiments. Feed–forward compensation term was not utilized. The observer gains were chosen as $\boldsymbol{\alpha_1} = \boldsymbol{\alpha_2} = 10\boldsymbol{I_2}$, and $\epsilon = 0.75$. The maximum and minimum values for the saturation set at ± 0.4 Nt-m.

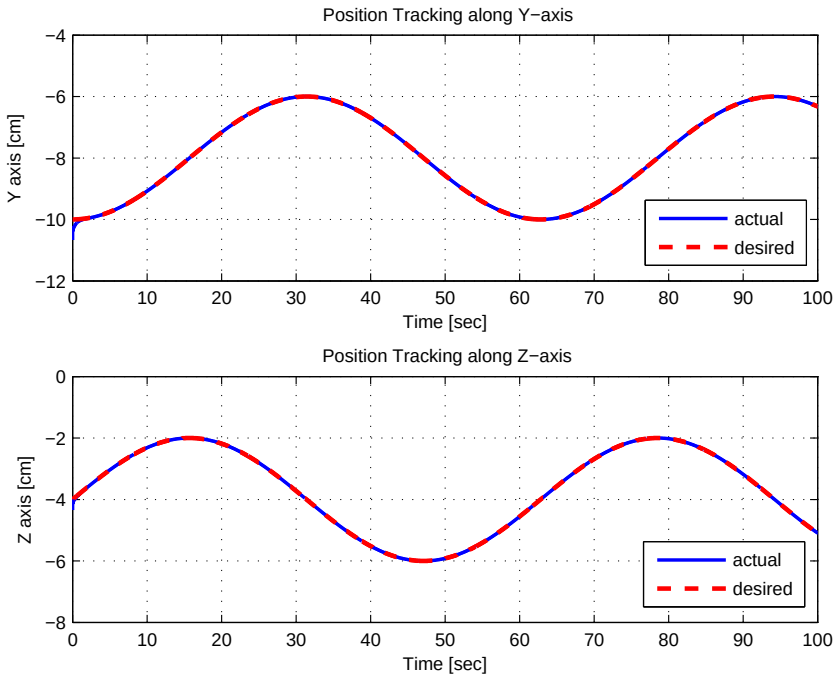

Figure 6.1. Simulation of Robust OFB: Desired and actual task–space trajectories.

The results of the experimental study are presented in Figures 6.4–6.6. In Figure 6.4, desired and actual task–space trajectories are presented. Figure 6.5 presents task–space tracking error versus observed task–space tracking error. In Figure 6.6, control input torques are presented. From Figure 6.4, it is clear that the tracking control objective was met while from Figure 6.5, it is seen that observer objective is achieved. It should be noted that if a feed–forward component is designed then better tracking may be achieved.

6.5. Conclusions

In this chapter, a new task–space tracking controller formulation was presented for robot manipulators. The proposed robust observer based output feedback controller ensured uniformly, ultimately bounded end–effector tracking performance without requiring measurements of joint variables and without the need of accurate knowledge of kinematics and dynamics of the robot manipulator. Only end–effector position measurements were used thus a simple control structure was designed with minimum requirements. Numerical simulation and experiment results confirmed the performance of the proposed controller.

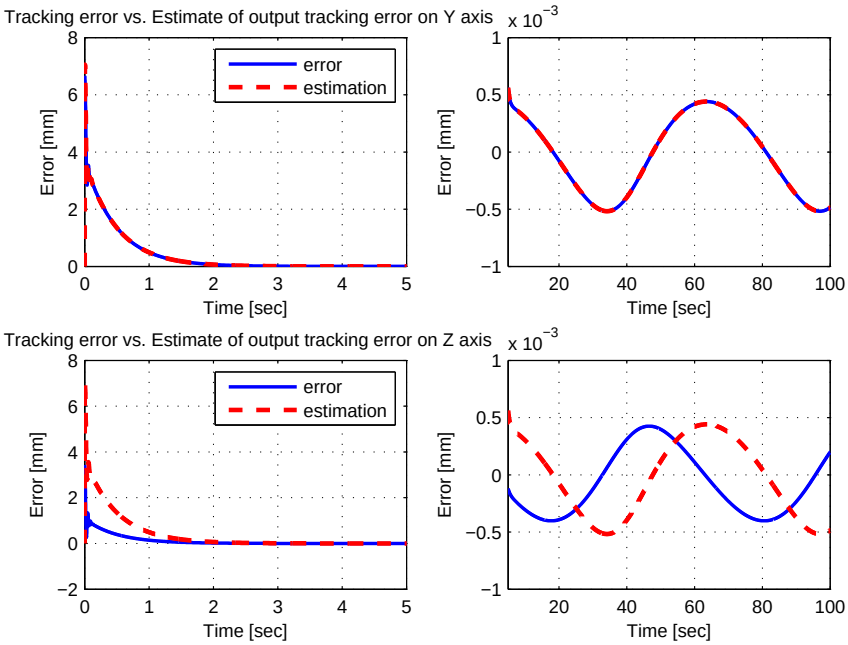

Figure 6.2. Simulation of Robust OFB: $\boldsymbol{e}(t)$ vs. $\hat{\boldsymbol{e}}(t)$.

When the proposed conroller was compared to the previous robust task–space controllers in the literature, this is one of the few robust and output feedback task–space controllers that achieved practical end–effector tracking.

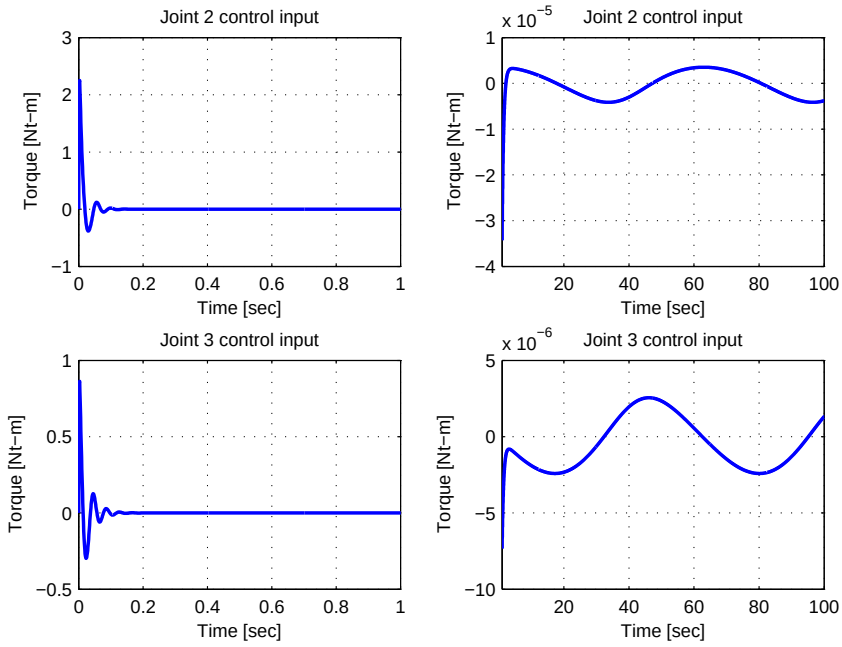

Figure 6.3. Simulation of Robust OFB: Control input torques $\boldsymbol{\tau}(t)$.

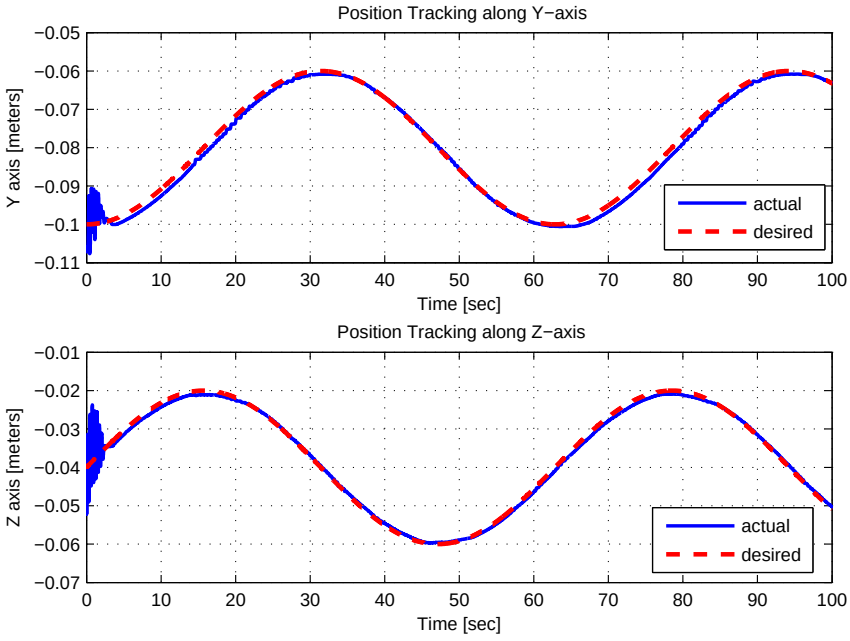

Figure 6.4. Experiment of Robust OFB: Desired and actual task–space trajectories.

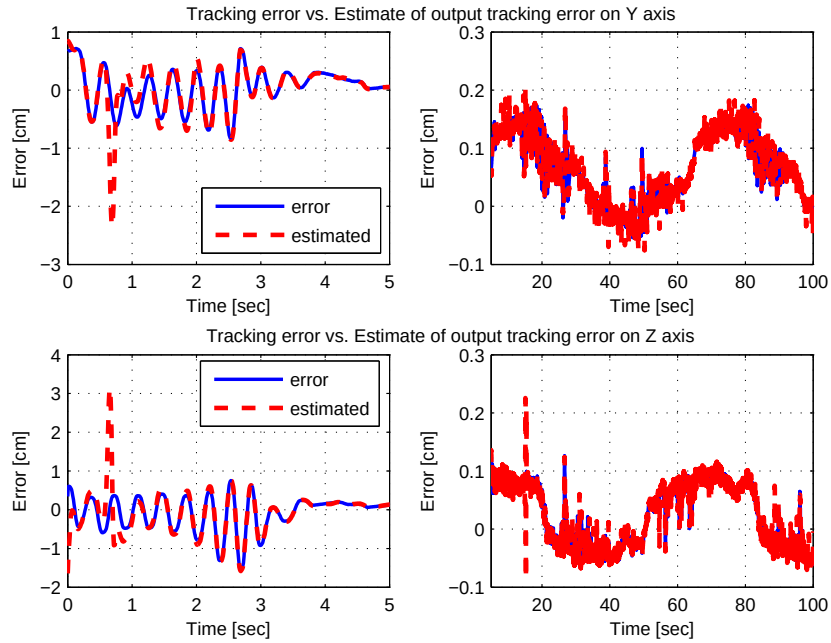

Figure 6.5. Experiment of Robust OFB: $\boldsymbol{e}(t)$ vs. $\hat{\boldsymbol{e}}(t)$.

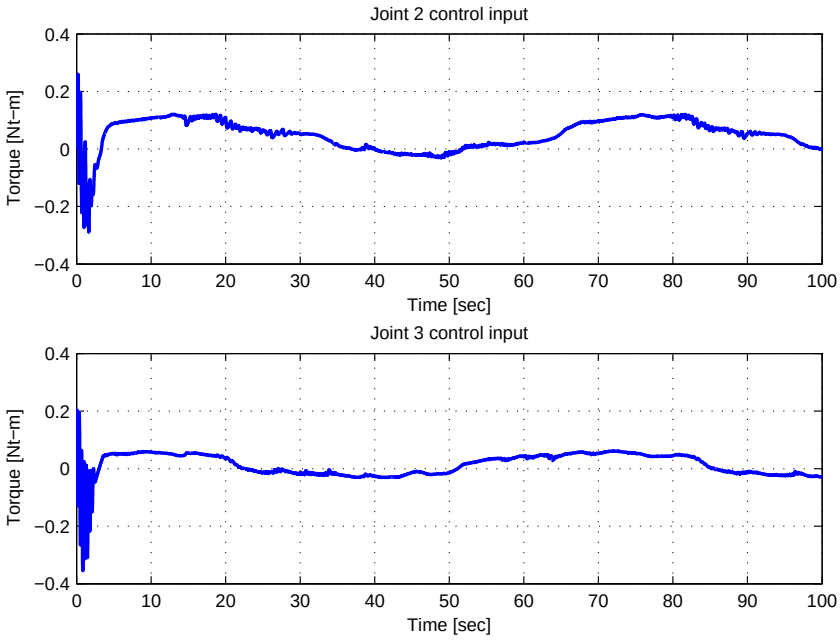

Figure 6.6. Experiment of Robust OFB: Control input torques $\boldsymbol{\tau}(t)$.

CHAPTER 7

PASSIVE DECOMPOSITION: A TASK–SPACE CONTROL APPROACH

In this chapter, a passive decomposition approach is proposed for task–space control of bilateral teleoperation systems. As shown in Figure 7.1, the proposed approach is based on availability of exact model knowledge of kinematic and dynamic models, and full–state feedback with force sensing from master and slave robots. In this study, master and slave robots are considered to be physically close so the control unit can be located at either system or in between, and thus there is no communication line induced time delay neither between the master robot and the control unit nor between the control unit and the slave robot. Considering a $2n_1$ dof bilateral teleoperation system consisting of kinematically similar two n_1 dof non–redundant master and slave robots, the closed–loop teleoperation system is decomposed in task–space into two n_1 dof subsystems as shape and locked subsystems by utilizing a decomposition matrix. A combined task–space controller is then designed. The aim of the proposed controller is to ensure that the end–effector of the slave robot tracks the end–effector of the master robot while obtaining a desired overall motion for the closed–loop teleoperation system.

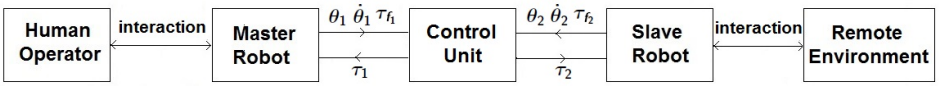

Figure 7.1. Flow diagram of a passive decomposition system.

7.1. Mathematical Models of Master and Slave Robots

In this chapter, for the kinematic models of master and slave robots, n_1 dof non–redundant robot model in (2.1) is utilized. The dynamic model in (2.44) is utilized with Properties 3 and 4. During the subsequent development of passive decomposition method, the numbers of dof of master and slave robots will be considered same (*i.e.,* $n_1 = n_2$).

Therefore, the dimension of the closed–loop teleoperation system is $2n_1$.

7.2. Problem Formulation

The $2n_1$ dof teleoperator system is decomposed into two n_1 dof subsystems according to two design criteria. The first one is obtaining coordination of master and slave robots (shape subsystem). The second one is ensuring overall motion of the closed–loop teleoperator system (locked subsystem).

A decomposition matrix, denoted by $\boldsymbol{S} \in \mathbb{R}^{2n_1 \times 2n_1}$, is defined as

$$\boldsymbol{S} \triangleq \begin{bmatrix} \left(\boldsymbol{I}_{n_1} - \bar{\phi}\right) \boldsymbol{J_1} & \bar{\phi}\boldsymbol{J_2} \\ \boldsymbol{J_1} & -\boldsymbol{J_2} \end{bmatrix} \tag{7.1}$$

where $\bar{\boldsymbol{\phi}}\left(\boldsymbol{x_1}, \boldsymbol{x_2}\right) \in \mathbb{R}^{n_1 \times n_1}$ is defined as

$$\bar{\phi} \triangleq \left(\bar{\boldsymbol{M}}_{\boldsymbol{2}}^{-1} \bar{\boldsymbol{M}}_{\boldsymbol{1}} + \boldsymbol{I}_{n_1}\right)^{-1} \tag{7.2}$$

where $\bar{\boldsymbol{M}}_{\boldsymbol{1}}(\boldsymbol{\theta_1}), \bar{\boldsymbol{M}}_{\boldsymbol{2}}(\boldsymbol{\theta_2}) \in \mathbb{R}^{n_1 \times n_1}$ denote generalized mass matrices of master and slave robots in task–space, respectively, and are defined as

$$\bar{\boldsymbol{M}}_{\boldsymbol{1}} \triangleq \boldsymbol{J}_{\boldsymbol{1}}^{-T} \boldsymbol{M}_{\boldsymbol{1}} \boldsymbol{J}_{\boldsymbol{1}}^{-1} \tag{7.3}$$

$$\bar{\boldsymbol{M}}_{\boldsymbol{2}} \triangleq \boldsymbol{J}_{\boldsymbol{2}}^{-T} \boldsymbol{M}_{\boldsymbol{2}} \boldsymbol{J}_{\boldsymbol{2}}^{-1}. \tag{7.4}$$

Then, a transformation from joint–space to task–space is defined as follows

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{L}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{E}} \end{bmatrix} = \boldsymbol{S} \begin{bmatrix} \dot{\boldsymbol{\theta}}_{\boldsymbol{1}} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{2}} \end{bmatrix} \tag{7.5}$$

where $\dot{\boldsymbol{x}}_{\boldsymbol{L}}(t), \dot{\boldsymbol{x}}_{\boldsymbol{E}}(t) \in \mathbb{R}^{n_1}$ represent task–space velocities of locked and shape subsystems, respectively.

Using the transformation in (7.5), the combined dynamics in (2.45) can now be transformed into task–space as

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{\boldsymbol{L}} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \bar{\boldsymbol{M}}_{\boldsymbol{E}} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{\boldsymbol{L}} \\ \ddot{\boldsymbol{x}}_{\boldsymbol{E}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{C}}_{\boldsymbol{L}} & \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \\ \bar{\boldsymbol{C}}_{\boldsymbol{EL}} & \bar{\boldsymbol{C}}_{\boldsymbol{E}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{L}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{E}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{L}} \\ \boldsymbol{\tau}_{\boldsymbol{E}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{f_L}} \\ \boldsymbol{\tau}_{\boldsymbol{f_E}} \end{bmatrix} \tag{7.6}$$

where $\bar{\boldsymbol{M}}_{\boldsymbol{L}}(t)$, $\bar{\boldsymbol{M}}_{\boldsymbol{E}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{L}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{LE}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{EL}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{E}}(t) \in \mathbb{R}^{n_1 \times n_1}$ and $\boldsymbol{\tau}_{\boldsymbol{f_L}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{f_E}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{L}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{E}}(t) \in \mathbb{R}^{n_1}$ are obtained from

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{\boldsymbol{L}} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \bar{\boldsymbol{M}}_{\boldsymbol{E}} \end{bmatrix} \triangleq \boldsymbol{S}^{-T} \begin{bmatrix} \boldsymbol{M}_{\boldsymbol{1}} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \boldsymbol{M}_{\boldsymbol{2}} \end{bmatrix} \boldsymbol{S}^{-1} \tag{7.7}$$

$$\begin{bmatrix} \bar{\boldsymbol{C}}_{\boldsymbol{L}} & \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \\ \bar{\boldsymbol{C}}_{\boldsymbol{EL}} & \bar{\boldsymbol{C}}_{\boldsymbol{E}} \end{bmatrix} \triangleq \boldsymbol{S}^{-T} \begin{bmatrix} \boldsymbol{C_1} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \boldsymbol{C_2} \end{bmatrix} \boldsymbol{S}^{-1} - \boldsymbol{S}^{-T} \begin{bmatrix} \boldsymbol{M_1} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \boldsymbol{M_2} \end{bmatrix} \boldsymbol{S}^{-1} \dot{\boldsymbol{S}} \boldsymbol{S}^{-1} \tag{7.8}$$

$$\begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{f_L}} \\ \boldsymbol{\tau}_{\boldsymbol{f_E}} \end{bmatrix} \triangleq \boldsymbol{S}^{-T} \begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{f_1}} \\ \boldsymbol{\tau}_{\boldsymbol{f_2}} \end{bmatrix} \tag{7.9}$$

$$\begin{bmatrix} \boldsymbol{\tau_L} \\ \boldsymbol{\tau_E} \end{bmatrix} \triangleq \boldsymbol{S}^{-T} \begin{bmatrix} \boldsymbol{\tau_1} \\ \boldsymbol{\tau_2} \end{bmatrix}. \tag{7.10}$$

From (7.6), the dynamics of locked and shape subsystems can be separately represented as

$$\bar{\boldsymbol{M}}_{\boldsymbol{L}} \ddot{\boldsymbol{x}}_{\boldsymbol{L}} + \bar{\boldsymbol{C}}_{\boldsymbol{L}} \dot{\boldsymbol{x}}_{\boldsymbol{L}} + \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} = \boldsymbol{\tau_L} + \boldsymbol{\tau}_{\boldsymbol{f_L}} \tag{7.11}$$

$$\bar{\boldsymbol{M}}_{\boldsymbol{E}} \ddot{\boldsymbol{x}}_{\boldsymbol{E}} + \bar{\boldsymbol{C}}_{\boldsymbol{E}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} + \bar{\boldsymbol{C}}_{\boldsymbol{EL}} \dot{\boldsymbol{x}}_{\boldsymbol{L}} = \boldsymbol{\tau_E} + \boldsymbol{\tau}_{\boldsymbol{f_E}}. \tag{7.12}$$

7.3. Control Design

Based on the shape subsystem dynamics in (7.12), the shape subsystem controller $\boldsymbol{\tau_E}(t)$ is designed as

$$\boldsymbol{\tau_E} = \bar{\boldsymbol{C}}_{\boldsymbol{EL}} \dot{\boldsymbol{x}}_{\boldsymbol{L}} - \boldsymbol{K}_{\boldsymbol{v}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} - \boldsymbol{K}_{\boldsymbol{p}} \boldsymbol{x}_{\boldsymbol{E}} - \boldsymbol{\tau}_{\boldsymbol{f_E}} \tag{7.13}$$

where $\boldsymbol{K}_{\boldsymbol{v}}$, $\boldsymbol{K}_{\boldsymbol{p}} \in \mathbb{R}^{n_1 \times n_1}$ are constant, positive–definite, diagonal, control gain matrices. After substituting (7.13) into (7.12), the closed-loop shape subsystem dynamics can be obtained as

$$\bar{\boldsymbol{M}}_{\boldsymbol{E}} \ddot{\boldsymbol{x}}_{\boldsymbol{E}} + \bar{\boldsymbol{C}}_{\boldsymbol{E}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} + \boldsymbol{K}_{\boldsymbol{v}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} + \boldsymbol{K}_{\boldsymbol{p}} \boldsymbol{x}_{\boldsymbol{E}} = \boldsymbol{0}_{n_1 \times 1} \tag{7.14}$$

from which it is easy to see that the first control objective for the shape subsystem is achieved in the sense that

$$\boldsymbol{x_E} = \boldsymbol{x_1} - \boldsymbol{x_2} \rightarrow \boldsymbol{0}_{n_1 \times 1} \Leftrightarrow \boldsymbol{x_1} = \boldsymbol{x_2} \tag{7.15}$$

thus perfect coordination between end–effectors of master and slave robots is achieved.

In order to ensure that the locked subsystem dynamics tracks a task–space trajectory generated by a combination of human and environmental forces, the locked subsystem controller is designed as

$$\boldsymbol{\tau_L} = \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \dot{\boldsymbol{x}}_{\boldsymbol{E}}. \tag{7.16}$$

Substituting (7.16) into (7.11) results in

$$\bar{\boldsymbol{M}}_{\boldsymbol{L}}\ddot{\boldsymbol{x}}_{\boldsymbol{L}} + \bar{\boldsymbol{C}}_{\boldsymbol{L}}\dot{\boldsymbol{x}}_{\boldsymbol{L}} = \boldsymbol{\tau}_{\boldsymbol{f_L}} \tag{7.17}$$

thus the second control objective is achieved.

7.4. Simulation Results

To illustrate the performance of the proposed controller, a numerical simulation was performed utilizing the model of the PHANToM Omni haptic device for both master and slave robots. The human forces affecting the master robot is represented as $\boldsymbol{\tau}_{\boldsymbol{f_1}} = 0.001 \times [\sin(2\pi t), \cos(2\pi t), \sin(2\pi t)]^T$ and the environmental forces affecting the slave robot is represented as $\boldsymbol{\tau}_{\boldsymbol{f_2}} = 0.001 \times [\cos(2\pi t), \sin(2\pi t), \cos(2\pi t)]$.

The simulation study was performed with sampling rate of 1000Hz. The control gains were chosen as $\boldsymbol{K}_{\boldsymbol{v}} = 10\boldsymbol{I}_{\boldsymbol{3}}$ and $\boldsymbol{K}_{\boldsymbol{p}} = 20\boldsymbol{I}_{\boldsymbol{3}}$. The master and slave robots were initialized to be at rest at joint positions $\boldsymbol{\theta_1}(0) = \boldsymbol{\theta_2}(0) = [0, 0.26, -0.37]^T$rad.

The simulation results are presented in Figures 7.2-7.4. Figure 7.2 presents the end–effector positions of master and slave robots. Figure 7.3 shows the position of the shape subsystem. The control input torques of the master and slave robots are presented in Figure 7.4. From Figures 7.2 and 7.3, it is clear that the control objective is achieved by ensuring perfect coordination between the master and slave robots.

7.5. Conclusions

In this chapter, a passive decomposition approach was proposed for task–space control of a $2n_1$ dof teleoperation system consisting two n_1 dof non–redundant master and slave robots. The $2n_1$ dof teleoperation system was decomposed into two n_1 dof locked and shape subsystems. Shape and locked subsystem controllers were designed to achieve coordination between master and slave robots in the task–space, and to obtain a desired overall motion for the closed-loop teleoperation system. A simulation study was performed by using the model of PHANToM Omni Haptic device for both master and slave robots to confirm the performance of the proposed approach.

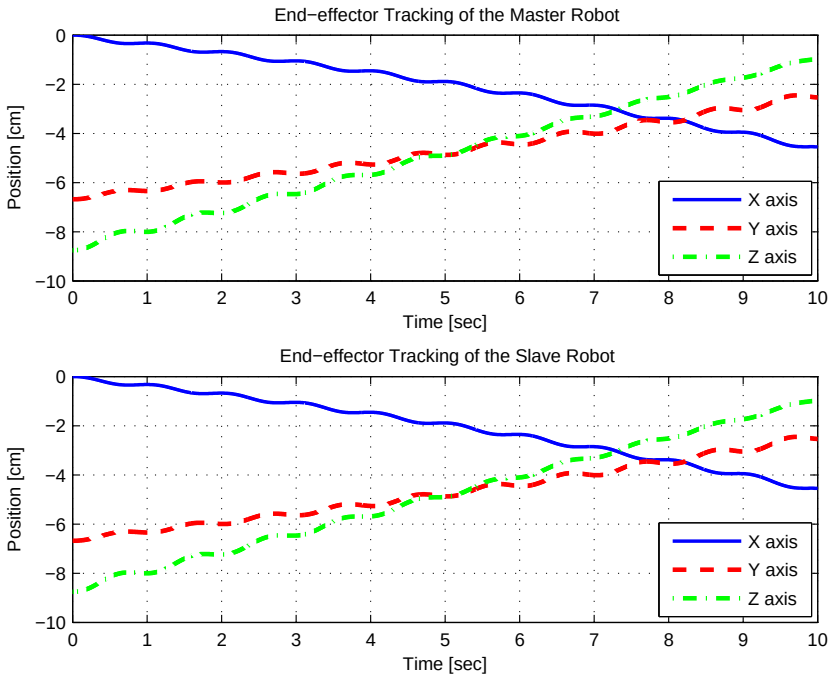

Figure 7.2. End–effector tracking of master and slave robots.

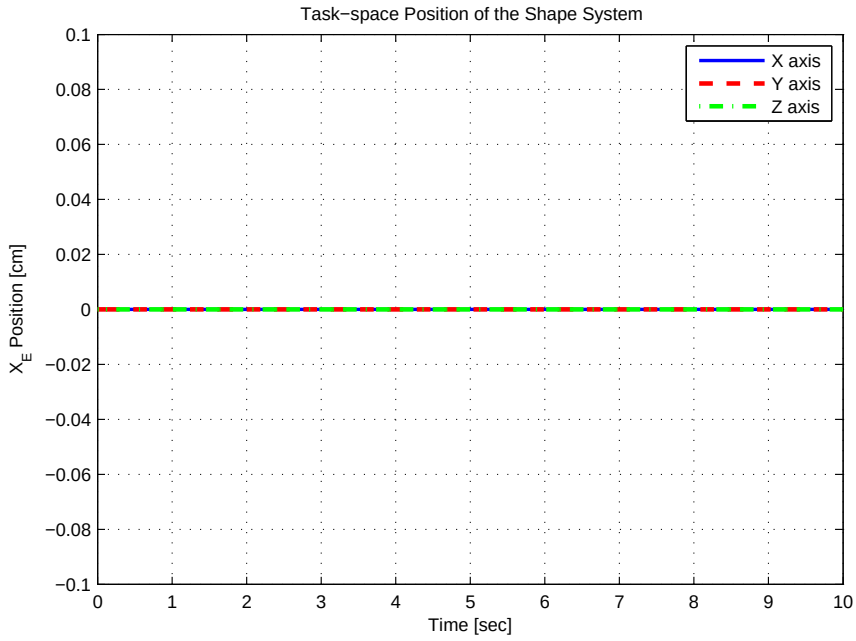

Figure 7.3. Positions of shape subsystem.

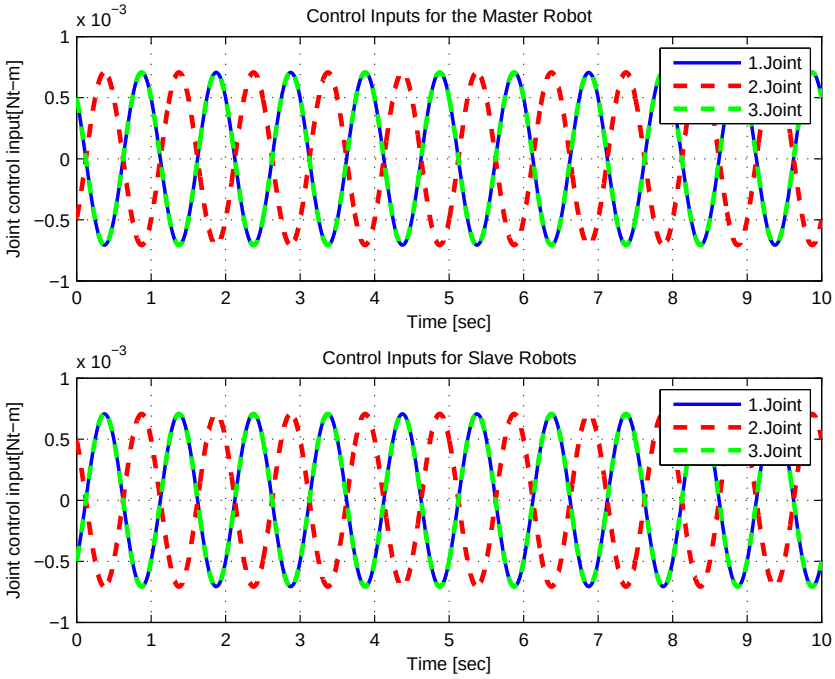

Figure 7.4. Control input torques of master and slave robots.

CHAPTER 8

PASSIVE DECOMPOSITION BASED TASK–SPACE CONTROL OF REDUNDANT TELEOPERATION SYSTEMS

In this chapter, passive decomposition based task–space control of kinematically redundant teleoperation systems is considered. The control problem is complicated by the slave robot having more dof than the master robot. As shown in Figure 7.1, the proposed approach is based on availability of exact model knowledge of kinematic and dynamic models, and full–state feedback with force sensing from master and slave robots in accordance with the previous works on passive decomposition. Similar to Chapter 7, master and slave systems are considered to be physically close. Considering an $n_1 + n_2$ dof redundant teleoperation system consisting of an n_1 dof non-redundant master robot and an n_2 dof redundant slave robot (with $n_2 > n_1$), the closed–loop teleoperation system is decomposed in task–space into two n_1 dof subsystems as shape and locked subsystems by utilizing a non-square decomposition matrix. A combined task–space controller is then designed. The controller ensures that the end–effector of the slave robot tracks the end–effector of the master robot while obtaining a desired overall motion for the closed–loop teleoperation system. The $n_2 - n_1$ redundant dof in the slave robot are made use of via the design of a null–space controller. Specifically, to solve the redundancy resolution, the pseudo–inverse Jacobian method in Siciliano (1990) is utilized. Experimental studies are conducted to illustrate the performance of the proposed control approach.

8.1. Mathematical Models of Master and Slave Robots

In this chapter, the kinematic models in (2.1) and (2.14) are utilized for the n_1 dof non-redundant master and n_2 dof redundant slave robots, respectively. The dynamic model in (2.44) and (2.45) are utilized with Properties 3 and 4.

8.2. Problem Formulation

In order to decompose the model of the redundant teleoperator system into two n_1 dof subsystems according to two design criteria: 1) ensuring coordination of end–effectors of master and slave robots (shape subsystem); 2) obtaining a desired overall motion for the closed-loop teleoperator (locked subsystem), a decomposition matrix, denoted by $\boldsymbol{S}(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) \in \mathbb{R}^{2n_1 \times (n_1+n_2)}$, is defined as follows

$$\boldsymbol{S} \triangleq \begin{bmatrix} \left(\boldsymbol{I}_{n_1} - \bar{\boldsymbol{\Phi}}\right) \boldsymbol{J_1} & \bar{\boldsymbol{\Phi}} \boldsymbol{J_2} \\ \boldsymbol{J_1} & -\boldsymbol{J_2} \end{bmatrix} \tag{8.1}$$

where $\bar{\boldsymbol{\Phi}}\left(\boldsymbol{x_1}, \boldsymbol{x_2}\right) \in \mathbb{R}^{n_1 \times n_1}$ is defined as

$$\bar{\boldsymbol{\Phi}} \triangleq \left(\bar{\boldsymbol{M}}_{\boldsymbol{2}}^{-1} \bar{\boldsymbol{M}}_{\boldsymbol{1}} + \boldsymbol{I}_{\boldsymbol{n_1}}\right)^{-1} \tag{8.2}$$

where $\bar{\boldsymbol{M}}_{\boldsymbol{1}}(\boldsymbol{\theta_1}), \bar{\boldsymbol{M}}_{\boldsymbol{2}}(\boldsymbol{\theta_2}) \in \mathbb{R}^{n_1 \times n_1}$ denote generalized mass matrices of master and slave robots in task–space, respectively, and are defined as

$$\bar{\boldsymbol{M}}_{\boldsymbol{1}} \triangleq \boldsymbol{J}_{\boldsymbol{1}}^{-T} \boldsymbol{M_1} \boldsymbol{J}_{\boldsymbol{1}}^{-1} \tag{8.3}$$

$$\bar{\boldsymbol{M}}_{\boldsymbol{2}} \triangleq \boldsymbol{J}_{\boldsymbol{2}}^{+T} \boldsymbol{M_2} \boldsymbol{J}_{\boldsymbol{2}}^{+}. \tag{8.4}$$

Then a transformation is defined from joint–space to task–space as

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{L}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{E}} \end{bmatrix} = \boldsymbol{S} \begin{bmatrix} \dot{\boldsymbol{\theta}}_{\boldsymbol{1}} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{2}} \end{bmatrix} \tag{8.5}$$

where $\dot{\boldsymbol{x}}_{\boldsymbol{L}}(t), \dot{\boldsymbol{x}}_{\boldsymbol{E}}(t) \in \mathbb{R}^{n_1}$ represent task–space velocities of locked and shape subsystems, respectively. Pseudo–inverse of the decomposition matrix in (8.5), denoted by $\boldsymbol{S}^{+}(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) \in \mathbb{R}^{(n_1+n_2) \times 2n_1}$, is also required to transform the combined dynamics in joint–space to task–space and is defined as

$$\boldsymbol{S}^{+} \triangleq \boldsymbol{S}^{T} (\boldsymbol{S}\boldsymbol{S}^{T})^{-1}. \tag{8.6}$$

Using the transformation in (8.5) with (8.6), the combined dynamics in (2.45) can now be transformed into task–space as

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{\boldsymbol{L}} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \bar{\boldsymbol{M}}_{\boldsymbol{E}} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{\boldsymbol{L}} \\ \ddot{\boldsymbol{x}}_{\boldsymbol{E}} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{C}}_{\boldsymbol{L}} & \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \\ \bar{\boldsymbol{C}}_{\boldsymbol{EL}} & \bar{\boldsymbol{C}}_{\boldsymbol{E}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{L}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{E}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{L}} \\ \boldsymbol{\tau}_{\boldsymbol{E}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{f_L}} \\ \boldsymbol{\tau}_{\boldsymbol{f_E}} \end{bmatrix} \tag{8.7}$$

where $\bar{\boldsymbol{M}}_{\boldsymbol{L}}(t)$, $\bar{\boldsymbol{M}}_{\boldsymbol{E}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{L}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{LE}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{EL}}(t)$, $\bar{\boldsymbol{C}}_{\boldsymbol{E}}(t) \in \mathbb{R}^{n_1 \times n_1}$ and $\boldsymbol{\tau}_{\boldsymbol{f_L}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{f_E}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{L}}(t)$, $\boldsymbol{\tau}_{\boldsymbol{E}}(t) \in \mathbb{R}^{n_1}$ are obtained from

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{\boldsymbol{L}} & \boldsymbol{0}_{n_1 \times n_1} \\ \boldsymbol{0}_{n_1 \times n_1} & \bar{\boldsymbol{M}}_{\boldsymbol{E}} \end{bmatrix} \triangleq \boldsymbol{S}^{+T} \begin{bmatrix} \boldsymbol{M_1} & \boldsymbol{0}_{n_1 \times n_2} \\ \boldsymbol{0}_{n_2 \times n_1} & \boldsymbol{M_2} \end{bmatrix} \boldsymbol{S}^{+} \tag{8.8}$$

$$\begin{bmatrix} \bar{\boldsymbol{C}}_{\boldsymbol{L}} & \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \\ \bar{\boldsymbol{C}}_{\boldsymbol{EL}} & \bar{\boldsymbol{C}}_{\boldsymbol{E}} \end{bmatrix} \triangleq \boldsymbol{S}^{+T} \begin{bmatrix} \boldsymbol{C_1} & \boldsymbol{0}_{n_1 \times n_2} \\ \boldsymbol{0}_{n_2 \times n_1} & \boldsymbol{C_2} \end{bmatrix} \boldsymbol{S}^{+} - \boldsymbol{S}^{+T} \begin{bmatrix} \boldsymbol{M_1} & \boldsymbol{0}_{n_1 \times n_2} \\ \boldsymbol{0}_{n_2 \times n_1} & \boldsymbol{M_2} \end{bmatrix} \boldsymbol{S}^{+} \dot{\boldsymbol{S}} \boldsymbol{S}^{+} \tag{8.9}$$

$$\begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{f_L}} \\ \boldsymbol{\tau}_{\boldsymbol{f_E}} \end{bmatrix} \triangleq \boldsymbol{S}^{+T} \begin{bmatrix} \boldsymbol{\tau}_{\boldsymbol{f_1}} \\ \boldsymbol{\tau}_{\boldsymbol{f_2}} \end{bmatrix} \tag{8.10}$$

$$\begin{bmatrix} \boldsymbol{\tau_L} \\ \boldsymbol{\tau_E} \end{bmatrix} \triangleq \boldsymbol{S}^{+T} \begin{bmatrix} \boldsymbol{\tau_1} \\ \boldsymbol{\tau_2} \end{bmatrix}. \tag{8.11}$$

From (8.7), the dynamics of locked and shape subsystems can be separately represented as

$$\bar{\boldsymbol{M}}_{\boldsymbol{L}} \ddot{\boldsymbol{x}}_{\boldsymbol{L}} + \bar{\boldsymbol{C}}_{\boldsymbol{L}} \dot{\boldsymbol{x}}_{\boldsymbol{L}} + \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} = \boldsymbol{\tau_L} + \boldsymbol{\tau}_{\boldsymbol{f_L}} \tag{8.12}$$

$$\bar{\boldsymbol{M}}_{\boldsymbol{E}} \ddot{\boldsymbol{x}}_{\boldsymbol{E}} + \bar{\boldsymbol{C}}_{\boldsymbol{E}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} + \bar{\boldsymbol{C}}_{\boldsymbol{EL}} \dot{\boldsymbol{x}}_{\boldsymbol{L}} = \boldsymbol{\tau_E} + \boldsymbol{\tau}_{\boldsymbol{f_E}}. \tag{8.13}$$

8.3. Control Design

Based on the shape subsystem dynamics in (8.13), the shape subsystem controller $\boldsymbol{\tau_E}(t)$ is designed as

$$\boldsymbol{\tau_E} = \bar{\boldsymbol{C}}_{\boldsymbol{EL}} \dot{\boldsymbol{x}}_{\boldsymbol{L}} - \boldsymbol{K}_{\boldsymbol{v}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} - \boldsymbol{K}_{\boldsymbol{p}} \boldsymbol{x}_{\boldsymbol{E}} - \boldsymbol{\tau}_{\boldsymbol{f_E}} \tag{8.14}$$

where $\boldsymbol{K}_{\boldsymbol{v}}$, $\boldsymbol{K}_{\boldsymbol{p}} \in \mathbb{R}^{n_1 \times n_1}$ are constant, positive–definite, diagonal, control gain matrices. After substituting (8.14) into (8.13), the closed-loop shape subsystem dynamics can be obtained as

$$\bar{\boldsymbol{M}}_{\boldsymbol{E}} \ddot{\boldsymbol{x}}_{\boldsymbol{E}} + \bar{\boldsymbol{C}}_{\boldsymbol{E}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} + \boldsymbol{K}_{\boldsymbol{v}} \dot{\boldsymbol{x}}_{\boldsymbol{E}} + \boldsymbol{K}_{\boldsymbol{p}} \boldsymbol{x}_{\boldsymbol{E}} = \boldsymbol{0}_{n_1 \times 1} \tag{8.15}$$

from which it is easy to see that the first control objective for the shape subsystem is achieved in the sense that

$$\boldsymbol{x_E} = \boldsymbol{x_1} - \boldsymbol{x_2} \to \boldsymbol{0}_{n_1 \times 1} \Leftrightarrow \boldsymbol{x_1} = \boldsymbol{x_2} \tag{8.16}$$

thus perfect coordination between end–effectors of master and slave robots is achieved.

In order to ensure that the locked subsystem dynamics tracks a task–space trajectory generated by a combination of human and environmental forces, the locked subsystem controller is designed as

$$\boldsymbol{\tau_L} = \bar{\boldsymbol{C}}_{\boldsymbol{LE}} \dot{\boldsymbol{x}}_{\boldsymbol{E}}. \tag{8.17}$$

Substituting (8.17) into (8.12) results in

$$\bar{\boldsymbol{M}}_{\boldsymbol{L}}\ddot{\boldsymbol{x}}_{\boldsymbol{L}} + \bar{\boldsymbol{C}}_{\boldsymbol{L}}\dot{\boldsymbol{x}}_{\boldsymbol{L}} = \boldsymbol{\tau}_{\boldsymbol{f_L}} \tag{8.18}$$

thus the second control objective is achieved.

Since the slave robot is kinematically redundant, this redundancy can be utilized to satisfy a sub–task objective. To achieve a sub–task objective, the modified control input torque, denoted by $\underline{\boldsymbol{\tau}}_{\boldsymbol{2}}(t) \in \mathbb{R}^{n_2}$, is designed by fusing $\boldsymbol{\tau_2}(t)$ with a null-space controller, denoted by $\boldsymbol{\tau_{null}}(\boldsymbol{\theta_2}) \in \mathbb{R}^{n_2}$, as

$$\underline{\boldsymbol{\tau}}_{\boldsymbol{2}} = \boldsymbol{\tau_2} + (\boldsymbol{I}_{\boldsymbol{n_2}} - \boldsymbol{J}_{\boldsymbol{2}}^{+}\boldsymbol{J_2})\boldsymbol{\tau_{null}}. \tag{8.19}$$

Here, $(\boldsymbol{I}_{\boldsymbol{n_2}} - \boldsymbol{J}_{\boldsymbol{2}}^{+}\boldsymbol{J_2})$ represents an orthogonal projection matrix into the null–space of $\boldsymbol{J_2}$. In the literature, there are many redundancy resolution methods utilizing null–space projection Nakanishi et al. (2008). In this study, the null–space controller is designed as

$$\boldsymbol{\tau_{null}} = -\boldsymbol{K_{null}}\nabla \boldsymbol{g} \tag{8.20}$$

where $\boldsymbol{K_{null}} \in \mathbb{R}^{n_2 \times n_2}$ is a constant, positive–definite, diagonal, control gain matrix, and $\nabla \boldsymbol{g} \in \mathbb{R}^{n_2}$ is the gradient of a scalar sub–task function $g(\boldsymbol{\theta_2}) \in \mathbb{R}$.

8.4. Experimental Results

To illustrate the performance of the proposed controller, a 2 dof revolute joint planar robot manipulator was utilized as master robot and the model of a 3 dof revolute joint planar robot manipulator was used as virtual slave. For the master robot, PHANToM Omni haptic device shown in Fig. 2.2 was used as a 2 dof planar manipulator after mechanically fixing the first joint and utilizing the last two joints of the device. As can be seen from Figure 2.2, as a result of not using the first joint, the device only moves in YZ plane. It should be noted that since the device was considered to be a 2 dof planar manipulator, the first joint angle and motion on X axis were not utilized in the mathematical models in Section 2.2.1.

For redundant slave robot, the kinematic and dynamic models of 3 dof planar robot manipulator given in Section 2.2.2 is virtually implemented on the computer and run on MATLAB Simulink.

In the experiments, a torque sensor was not used to measure forces on the master robot, and instead a force observer was designed to estimate the applied forces by the

human operator. In order to design a force observer, the joint acceleration can be written from (2.44) as

$$\ddot{\boldsymbol{\theta}}_{\mathbf{1}} = \boldsymbol{M}_{\mathbf{1}}^{-1}(\boldsymbol{\tau}_{\mathbf{1}} - \boldsymbol{C}_{\mathbf{1}}\dot{\boldsymbol{\theta}}_{\mathbf{1}}) + \boldsymbol{M}_{\mathbf{1}}^{-1}\boldsymbol{\tau}_{\boldsymbol{f}_{\mathbf{1}}}. \tag{8.21}$$

The estimated joint acceleration, denoted by $\ddot{\hat{\boldsymbol{\theta}}}_{\mathbf{1}}(t) \in \mathbb{R}^2$, is designed as

$$\ddot{\hat{\boldsymbol{\theta}}}_{\mathbf{1}} = \boldsymbol{M}_{\mathbf{1}}^{-\mathbf{1}}(\boldsymbol{\tau}_{\mathbf{1}} - \boldsymbol{C}_{\mathbf{1}}\dot{\boldsymbol{\theta}}_{\mathbf{1}}) + \hat{\boldsymbol{\tau}}_{\boldsymbol{f}.\mathbf{1}} \tag{8.22}$$

where $\hat{\boldsymbol{\tau}}_{\boldsymbol{f}.\mathbf{1}}(t) \in \mathbb{R}^2$ represents the observer for the vector $\boldsymbol{M}_{\mathbf{1}}^{-1}\boldsymbol{\tau}_{\boldsymbol{f}_{\mathbf{1}}}$. The joint velocity observation error, denoted by $\boldsymbol{e}_{\boldsymbol{f}}(t) \in \mathbb{R}^2$, is defined as

$$\boldsymbol{e}_{\boldsymbol{f}} \triangleq \dot{\boldsymbol{\theta}}_{\mathbf{1}} - \dot{\hat{\boldsymbol{\theta}}}_{\mathbf{1}}. \tag{8.23}$$

The force observer is designed as

$$\hat{\boldsymbol{\tau}}_{\boldsymbol{f}.\mathbf{1}} = (\boldsymbol{K}_{\boldsymbol{f}} + \boldsymbol{I}_{\mathbf{2}})(\boldsymbol{e}_{\boldsymbol{f}}(t) + \int_0^t \boldsymbol{e}_{\boldsymbol{f}}(\sigma)d\sigma) + \boldsymbol{K}_{\boldsymbol{l}} \int_0^t \text{Tanh}(\boldsymbol{e}_{\boldsymbol{f}}(\sigma))d\sigma \tag{8.24}$$

where $\boldsymbol{K}_{\boldsymbol{f}}, \boldsymbol{K}_{\boldsymbol{l}} \in \mathbb{R}^{2\times 2}$ are constant, positive–definite, diagonal, observer gain matrices, and $\text{Tanh}(\cdot) \in \mathbb{R}^2$ represents the vector hyperbolic tangent function. Provided that $\boldsymbol{K}_{\boldsymbol{f}}$ and $\boldsymbol{K}_{\boldsymbol{l}}$ are chosen appropriately $\hat{\boldsymbol{\tau}}_{\boldsymbol{f}.\mathbf{1}}$ approaches to the neighborhood of $\boldsymbol{M}_{\mathbf{1}}^{-1}\boldsymbol{\tau}_{\boldsymbol{f}_{\mathbf{1}}}$ Dasdemir and Zergeroglu (2015).

For the virtual slave, an environmental force feedback was modeled by utilizing a spring-damper model, denoted by $\boldsymbol{f}_{\boldsymbol{sd}} \in \mathbb{R}^2$, as

$$\boldsymbol{f}_{\boldsymbol{sd}} = \boldsymbol{K}_{\boldsymbol{d}}\dot{\boldsymbol{x}}^* + \boldsymbol{K}_{\boldsymbol{s}}\boldsymbol{x}^* \tag{8.25}$$

where $\boldsymbol{K}_{\boldsymbol{s}}, \boldsymbol{K}_{\boldsymbol{d}} \in \mathbb{R}^{2\times 2}$ are constant, positive–definite, diagonal, spring and damper gain matrices, respectively, $\boldsymbol{x}^* \in \mathbb{R}^2$ is the position between the end–effector and the position of the virtually defined surface, and $\dot{\boldsymbol{x}}^* \in \mathbb{R}^2$ is the velocity after applying force on the virtual surface. The applied force on the joints of the slave robot can be written as follows

$$\boldsymbol{\tau}_{\boldsymbol{f}_{\mathbf{2}}} = \boldsymbol{J}_{\mathbf{2}}^T \boldsymbol{f}_{\boldsymbol{sd}}. \tag{8.26}$$

The sub–task function was selected as $\boldsymbol{g} = (\theta_{2.2} - \theta_{2.3})^2$ which is minimized when the second and third joints have the same angular positions.

The experimental study was performed with a sampling rate of 100Hz. The control gains were chosen as $\boldsymbol{K}_{\boldsymbol{v}} = \text{diag}\{15; 10\}$ and $\boldsymbol{K}_{\boldsymbol{p}} = \text{diag}\{7.5; 5\}$. The master and slave robots were initialized to be at rest at joint positions $\boldsymbol{\theta}_{\mathbf{1}}(0) = [0.55, 0.43]^T$rad. and $\boldsymbol{\theta}_{\mathbf{2}}(0) = [0, 1.57, 1.57]^T$rad., respectively. The force observer gains were chosen as $\boldsymbol{K}_{\boldsymbol{f}} = 10 \times \text{diag}\{5; 4\}$ and $\boldsymbol{K}_{\boldsymbol{l}} = 0.5 \times \text{diag}\{5; 4\}$. The corner of position of the virtual

surface was defined at $[0; -0.2]$(m) in Y and Z coordinates of the workspace of the virtual slave system. The spring and damper gain matrices were chosen as $\boldsymbol{K_s} = \boldsymbol{K_d} = 0.1\boldsymbol{I_2}$. The null–space control gain matrix was chosen as $\boldsymbol{K_{null}} = 1.2\boldsymbol{I_3}$.

The results of the experimental study are presented in Figures 8.1–8.6. Figure 8.1 presents the end–effector positions of master robot and virtual slave. Figure 8.2 presents the end–effector trajectories of master robot and virtual slave. Figure 8.3 presents position of shape subsystem $\boldsymbol{x_E}(t)$. The control input torques applied on master robot and virtual slave are presented in Figure 8.4. In Figure 8.5, joint positions of virtual slave are presented from which it is clear that the sub–task objective was met as $\theta_{2.2}$ is very close to $\theta_{2.3}$. Figure 8.6 presents estimation of human force applied on master robot and the modeled environmental force affecting virtual system when the end–effector of the virtual slave contacted with the determined virtual surface. From Figures 8.1, 8.2, and 8.3, it is clear that the main control objective is achieved by ensuring perfect coordination between master robot and virtual slave.

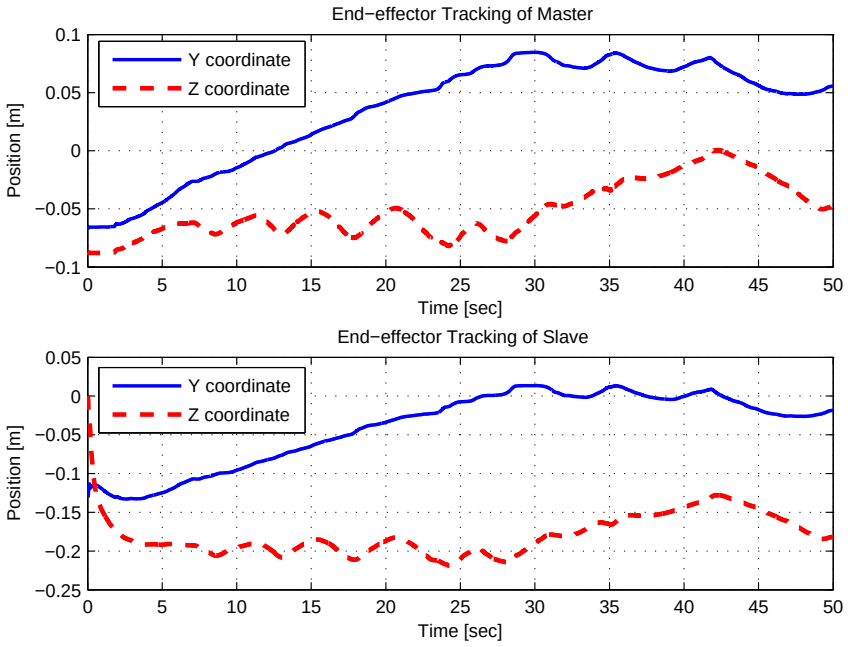

Figure 8.1. End–effector positions of master robot and virtual slave.

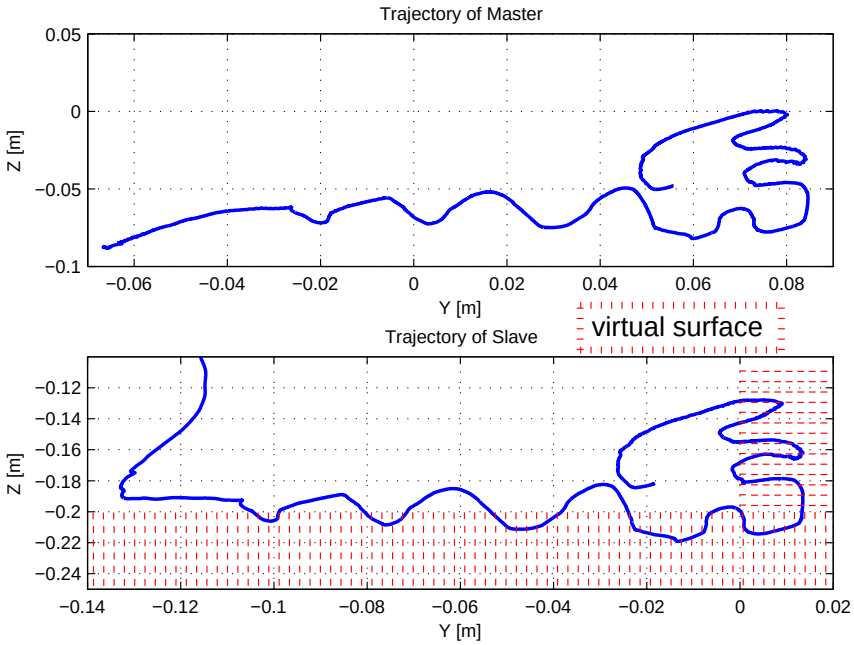

Figure 8.2. End–effector trajectories of master robot and virtual slave.

8.5. Conclusions

This chapter proposed a passive decomposition approach for task–space control of an n_1+n_2 dof bilateral teleoperation system consisting of an n_1 dof non-redundant master robot and an n_2 dof redundant slave robot. Provided that exact model knowledge of kinematic and dynamic models being available and full-state feedback with force sensing from both master and slave robots, the $n_1 + n_2$ dof teleoperation system was decomposed into two n_1 dof locked and shape subsystems. After introducing a non-square decomposition matrix, shape and locked subsystem controllers were designed to achieve coordination between master and slave robots in the task–space, and to obtain a desired overall motion for the closed–loop teleoperation system. To make use of the redundancy of the slave subsystem, its controller was fused with a null–space controller to minimize a quadratic scalar function of joint positions to achieve a secondary objective. An experimental study was conducted by using PHANToM Omni Haptic device as the non–redundant master robot and the model of a 3 dof planar robot manipulator as the redundant slave system to illustrate the performance of the proposed approach. The experiment results confirmed the performance of the proposed approach.

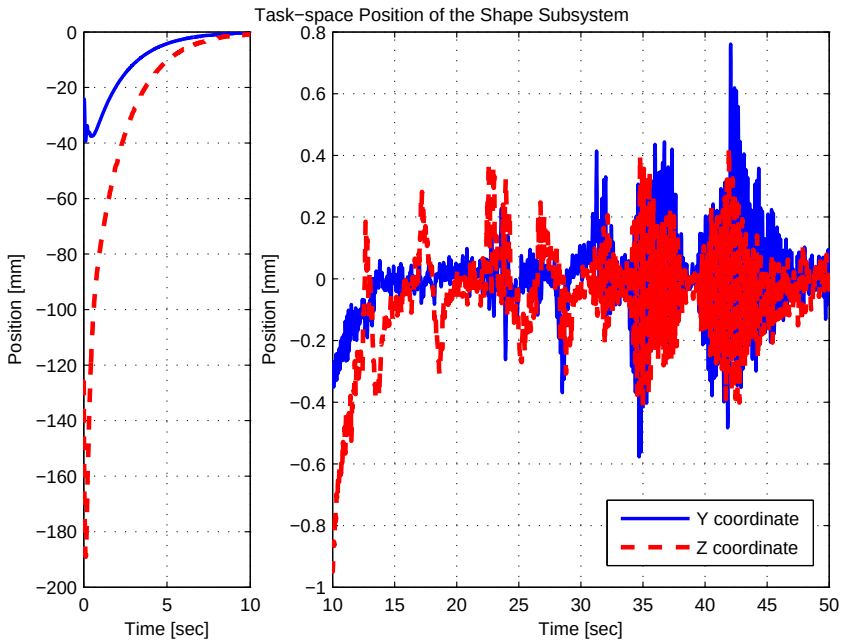

Figure 8.3. Positions of shape subsystem $\boldsymbol{x_E}(t)$.

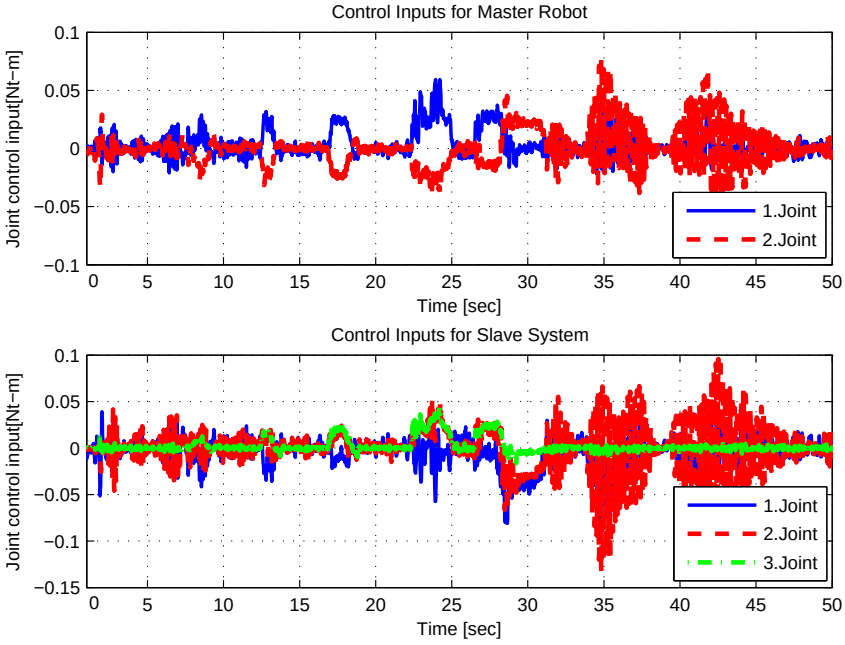

Figure 8.4. Control input torques of master robot and virtual slave.

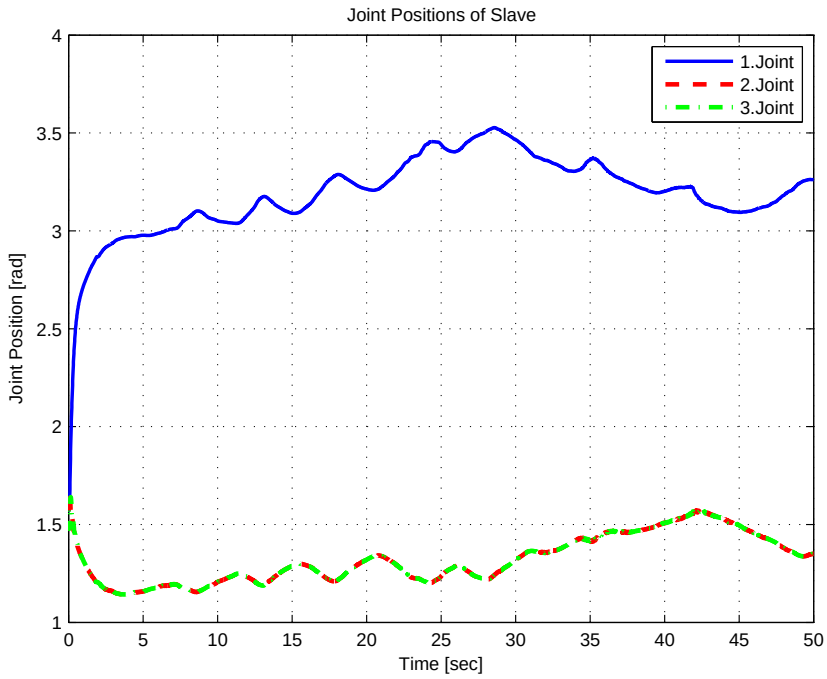

Figure 8.5. Joint positions of virtual slave.

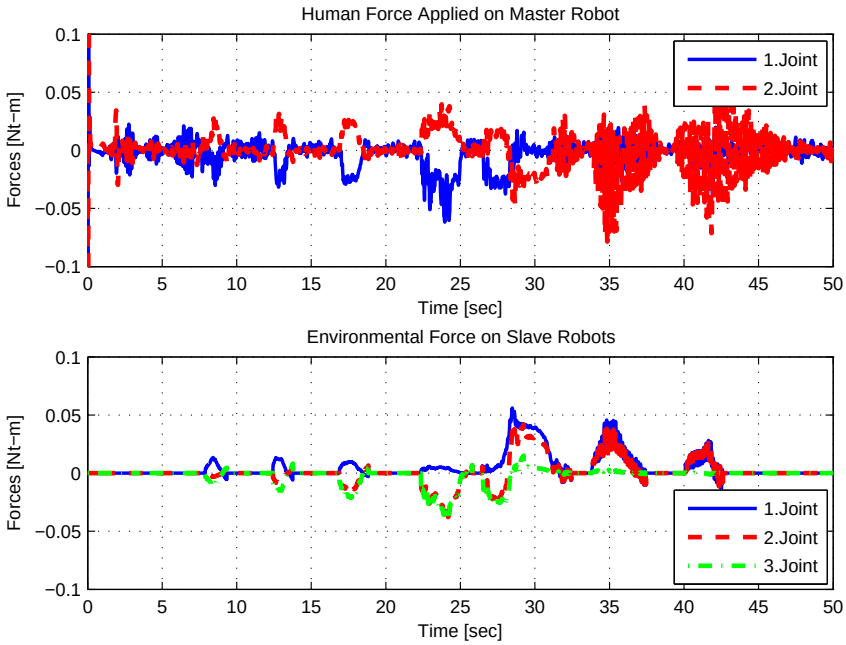

Figure 8.6. Estimate of human force (top) and environmental force in (8.25) (bottom).

CHAPTER 9

CONCLUSIONS

This thesis presented a three–stage framework for task–space control of robot manipulators with telerobotic applications. The first stage mainly focused on the design of an asymptotically stable sub–task controller for kinematically redundant robot manipulators. In the second stage, two robust control methods were proposed for task–space tracking of robot manipulators. The third stage aimed to develop passive decomposition methods for task–space control of bilateral teleoperation systems.

In order to design an asymptotically stable sub–task controller for kinematically redundant manipulators, two different approaches have been discussed in Chapters 3 and 4. Chapter 3 presented design and the corresponding stability analysis of two task–space controllers (one exact model knowledge and one adaptive) for redundant robot manipulators that utilizes the redundant dof for a predefined sub–task. The task–space controllers included a **novel** null–space controller. The sub–task function has been designed as an error–like function of joint positions. A combined stability analysis that ensured stability and convergence of both end–effector and sub–task function have been presented. To our best knowledge, the proposed combined stability analysis is **novel** when compared to the existing literature on control of redundant robot manipulators. Numerical simulations were conducted for both task–space controllers with and without null–space controllers.

In Chapter 4, an extended task–space control design with sub–task objectives was presented for kinematically redundant robot manipulators. Specifically, the main aim was to construct the extended Jacobian to be full rank by integrating the manipulator Jacobian matrix with the sub–task Jacobian matrix with the amount of the extra dof. Another motivation of Chapter 4 was to propose a method that is applicable to hyper redundant robot manipulators without requiring a separate stability analysis for multiple sub–task objectives. To deal with parametric uncertainties associated with robot dynamics, an adaptive controller was proposed. Lyapunov based stability analysis ensured stability and convergence of both task–space tracking and sub–task objectives. To our best knowledge, when compared to the existing literature on control of redundant robot manipulators, the proposed extended task–space controller is **novel** with two important specifications; the first one is that ensuring asymptotic stability of both task–space tracking and sub–task objectives, and the second one is that being applicable for multiple sub–tasks on hyper re-

dundant robot manipulators. Experiments on a 3 dof planar redundant robot manipulator were conducted to demonstrate the performance of the proposed adaptive controller with a **novel** sub–task. For the sub–task, the redundant robot manipulator was considered as being equipped with a camera or a laser tracer on the first link that traces the end–effector of the robot manipulator while performing the task–space tracking objective. Consequently, while the approach in Chapter 3 was based on pseudo–inverse Jacobian method and achieved asymptotically stable sub–task control for kinematically redundant robot manipulators which have only one redundant dof, the approach in Chapter 4 is based on extended Jacobian method and achieved asymptotically stable sub–task control for hyper redundant robot manipulators.

In Chapters 5 and 6, two robust control methods were proposed for task–space tracking of robot manipulators. Chapter 5 presented a **novel** continuous task–space controller formulation for robot manipulators. The proposed robust task–space controller ensured asymptotic task–space tracking despite unstructured uncertainties in the robot dynamics and structured uncertainties in the velocity kinematics. As opposed to most robust or variable structure controllers presented in the literature, the proposed robust controller is continuous, and asymptotic tracking was ensured. The stability of the proposed controller was ensured via rigorous theoretical analysis based on Lyapunov techniques. Experimental studies performed on a PHANToM Omni Haptic device confirmed the performance of the proposed robust controller.

In Chapter 6, a robust observer based output feedback controller formulation was presented for robot manipulators. The proposed robust output feedback controller ensured uniformly, ultimately bounded end–effector tracking performance without requiring measurements of joint position and velocity and without the need of accurate knowledge of kinematics and dynamics of the robot manipulator. Only end–effector position measurements were used thus a simple model–free control structure was designed with minimum requirements. Numerical simulation and experiment results confirmed the performance of the proposed controller. When the proposed conroller was compared to the previous robust task–space controllers in the literature, this is one of the few robust and output feedback task–space controllers that achieved practical end–effector tracking. When the two robust task–space controllers are compared, the continuous robust controller in Chapter 5 ensured asymptotic task–space tracking where measurements of joint positions and velocities are required, while the robust observer based output feedback controller in Chapter 6 ensured ultimately bounded task–space tracking without requiring neither measurements of joint positions or velocities nor accurate knowledge of kinematic and dynamic models.

While the designs in Chapters 5 and 6 were for non–redundant robots, via utilizing the extended Jacobian method in Chapter 4, they can be applied to kinematically redundant robot manipulators as well.

The third objective of this thesis was to develop a passive decomposition method for task–space control of bilateral teleoperation systems. In Chapter 7, a passive decomposition approach was proposed for task–space control of a $2n_1$ dof teleoperation system consisting of two n_1 dof non–redundant master and slave robots. The $2n_1$ dof teleoperation system was decomposed into two n_1 dof locked and shape subsystems. The proposed controllers of locked and shape subsystems in Chapter 7 ensured coordination of master and slave robots while achieving overall motion of the bilateral teleoperation system.

Chapter 8 proposed a passive decomposition approach for task–space control of an $n_1 + n_2$ dof bilateral teleoperation system consisting of an n_1 dof non–redundant master robot and an n_2 dof redundant slave robot. Provided that exact model knowledge of kinematic and dynamic models being available and full–state feedback with force sensing from both master and slave robots, the $n_1 + n_2$ dof teleoperator system was decomposed into two n_1 dof locked and shape subsystems. After introducing a non–square decomposition matrix, shape and locked subsystem controllers were designed to achieve coordination between master and slave robots in task–space, and to obtain a desired overall motion for the bilateral teleoperation system. To make use of the redundancy of the slave subsystem, its controller was fused with a null–space controller to minimize a quadratic scalar function of joint positions to achieve a secondary objective. An experimental study was conducted to illustrate the performance of the proposed approach by using PHANToM Omni Haptic device as the non–redundant master robot and the model of a 3 dof planar robot manipulator as the redundant slave system. As a conclusion, in Chapters 7 and 8, the proposed passive decomposition method was considered for task–space control of bilateral teleoperation systems for the first time. While, in Chapter 7, kinematically similar master and slave robots were decomposed into locked and shape subsystems, in Chapter 8, the proposed method was modified to be applicable for kinematically dissimilar teleoperation systems by introducing a non–square decomposition matrix.

9.1. Future Works

As future works for Chapters 3 and 4, the proposed full state feedback based controllers can be extended to output feedback based controllers. In Chapter 4, the viability of the proposed extended Jacobian method was experimentally verified on a 3 dof redundant

robot manipulator by utilizing only one sub–task objective. A possible future work can be experimentally verifying the proposed method on hyper redundant robot manipulators ($n \gg m$) by utilizing multiple sub–task objectives. For the proposed continuous robust controller in Chapter 5, there are some possible future works that could be done based on adaptive, optimal or neural network methods. This full state feedback based continuous robust controller can also be extended in future work by designing an observer to compensate for the lack of joint velocity measurements. As another future work, the proposed robust observer based output feedback controller in Chapter 6 can be modified to include an adaptive compensation component. Internet–based experimental studies of the proposed passive decomposition methods in Chapters 7 and 8 can be demonstrated on real master and slave systems which are far away from each other. As another future work, time delay challenges may be dealt with as an extension of the proposed passive decomposition methods.

REFERENCES

Abdallah, C., D. M. Dawson, P. Dorato, and M. Jamshidi (1991). Survey of robust control for rigid robots. *IEEE Control Systems 11*(2), 24–30.

An, C. H., C. G. Atkeson, and J. M. Hollerbach (1988). *Model-based Control of a Robot Manipulator*. Cambridge, MA, USA: MIT Press.

Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. In *IEEE International Conference on Robotics and Automation*, Volume 2, pp. 722–728.

Braganza, D., W. E. Dixon, D. M. Dawson, and B. Xian (2008). Tracking control for robot manipulators with kinematic and dynamic uncertainty. *Int. J. Robotics and Automation 23*(2), 5293–5297.

Braganza, D., M. L. McIntyre, D. M. Dawson, and I. D. Walker (2006). Whole arm grasping control for redundant robot manipulators. In *Proc. American Control Conf.*, Minneapolis, MN, USA, pp. 3194–3199.

Buckingham, R. and A. Graham (2005). Snaking around in a nuclear jungle. *Industrial Robot: An International Journal 32*(2), 120–127.

Cheah, C. C. (2003). Approximate jacobian robot control with adaptive jacobian matrix. In *Proc. IEEE Int. Conf. Decision and Control*, Maui, HI, USA, pp. 5859–5864.

Cheah, C. C. (2008). Task-space PD control of robot manipulators: Unified analysis and duality property. *Int. J. of Robotics Research 27*(10), 1152–1170.

Cheah, C. C., M. Hirano, S. Kawamura, and S. Arimoto (2003). Approximate jacobian control for robots with uncertain kinematics and dynamics. *IEEE Tr. on Robotics & Automation 19*(4), 692–702.

Cheah, C. C., M. Hirano, S. Kawamura, and S. Arimoto (2004). Approximate jacobian control with task-space damping for robot manipulators. *IEEE Tr. on Automatic Control 49*(5), 752–757.

Cheah, C. C., S. Kawamura, S. Arimoto, and K. Lee (1999). PID control of robotic manipulator with uncertain jacobian matrix. In *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, USA, pp. 494–499.

Cheah, C. C., K. Lee, S. Kawamura, and S. Arimoto (2000). Asymptotic stability of robot control with approximate jacobian matrix and its application to visual servoing. In *Proc. IEEE Int. Conf. Decision and Control*, Sydney, Australia, pp. 3939–3944.

Chen, J., A. Behal, and D. M. Dawson (2008). Robust feedback control for a class of uncertain MIMO nonlinear systems. *IEEE Tr. on Automatic Control 53*(2), 591–596.

Colbaugh, R. and K. Glass (1995). Robust adaptive control of redundant manipulators. *J. Intelligent and Robotic Systems 14*(1), 68–88.

Conkur, E. S. and R. Buckingham (1997). Clarifying the definition of redundancy as used in robotics. *Robotica 15*, 583–586.

Costa, R. R., L. Hsu, A. K. Imai, and P. Kokotović (2003). Lyapunov-based adaptive control of MIMO systems. *Automatica 39*(7), 1251–1257.

Dasdemir, J. and E. Zergeroglu (2015). A new continuous high-gain controller scheme for a class of uncertain nonlinear systems. *Int. J. of Robust and Nonlinear Control 25*(1), 125–141.

Dawson, D., M. Bridges, and Z. Qu (1995). *Nonlinear control of robotic systems for environmental waste and restoration*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

Dawson, D., Z. Quf, and J. Duffie (1993). Robust tracking control for robot manipulators: theory, simulation, and implementation. *Robotica 11*, 201–208.

Dixon, W. E. (2003). *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Boston, MA, USA: Birkhauser.

Dogan, K. M., E. Tatlicioglu, and E. Zergeroglu (2015). Operational/task space learning control of robot manipulators with dynamical uncertainties. In *IEEE Conf. on Control Applications*, Sydney, Australia, pp. 527–532.

Hashtrudi-Zaad, K. and S. E. Salcudean (2001). Analysis of control architectures for teleoperation systems with impedance / admittance master and slave manipulators. *Int. J. of Robotics Research 20*(6), 419–445.

Hokayem, P. F. and M. W. Spong (2006). Bilateral teleoperation: An historical survey. *Automatica 42*(12), 2035–2057.

Hsu, P., J. Hauser, and S. Sastry (1989). Dynamic control of redundant manipulators. *J. of Robotic Systems 6*(3), 133–148.

Ioannou, P. and J. Sun (1996). *Robust Adaptive Control*. Englewood Cliffs, NJ, USA: Prentice Hall.

Kam Lau, R. H. and L. Haynes (1985). Robot performance measurements using automatic laser techniques. *Int. J. of Robotics & Computer-Integrated Manufacturing 2*, 227–236.

Kapadia, A., E. Tatlicioglu, and D. M. Dawson (2008). Set-point navigation of a redundant robot in uncertain environments using finite range sensors. In *Proc. IEEE Int. Conf. Decision and Control*, Cancun, Mexico, pp. 4596–4601.

Khalil, H. (2002). *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

Khalil, H. (2015). *Nonlinear Control*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

Khatib, O. (1983). Dynamic control of manipulators in operational space. In *IFTOMM Cong. Theory of Machines and Mechanisms*, New Delhi, India, pp. 1–10.

Krstic, M., I. Kanellakopoulos, and P. Kokotovic (1995). *Nonlinear and Adaptive Control Design*. New York, NY, USA: John Wiley & Sons.

Lee, D. and P. Y. Li (2002a). Passive coordination control of nonlinear bilateral teleoperated manipulators. In *IEEE Int. Conf. on Robotics and Automation*, Washington, DC, USA, pp. 3278–3283.

Lee, D. and P. Y. Li (2002b). Passive tool dynamics rendering for nonlinear bilateral teleoperated manipulators. In *IEEE Int. Conf. on Robotics and Automation*, Washington,

DC, USA, pp. 3284–3289.

Lee, D. and P. Y. Li (2003). Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators. *IEEE Tr. on Robotics and Automation 19*(3), 443–456.

Lee, D. and P. Y. Li (2007). Passive decomposition approach to formation and maneuver control of multiple rigid bodies. *J. of Dynamic Systems, Measurement, and Control 129*(5), 662–677.

Lewis, F. L., C. T. Abdallah, and D. M. Dawson (1993). *Control of robot manipulators*. New York, NY, USA: Macmillan Pub. Co.

Lewis, F. L., D. M. Dawson, and C. T. Abdallah (2003). *Robot Manipulator Control: Theory and Practice*. New York, NY, USA: Marcel Dekker.

Li, P. Y. and D. Lee (2000). Passive feedforward approach to bilateral teleoperated manipulators. In *Proc. of ASME IMECE Symphosium on Haptic Interfaces for Virtual Reality and Teleoperator Systems*, Orlando, FL, USA, pp. 1153–1160.

Luo, S. and S. Ahmad (1997). Adaptive control of kinematically redundant robots. *IMA J. of Mathematical Control & Information 14*, 225–253.

Nakamura, Y. (1991). *Advanced Robotics Redundancy and Optimization*. Reading, MA, USA: Addison-Wesley.

Nakanishi, J., R. Cory, M. Mistry, J. Peters, and S. Schaal (2008). Operational space control: A theoretical and empirical comparison. *Int. J. of Robotics Research 27*(6), 737–757.

Nenchev, D. (1989). Redundancy resolution through local optimization: A review. *J. of Robotic Systems 6*(6), 769–798.

Nicosia, S. and P. Tomei (1990). Robot control by using only joint position measurements. *IEEE Tr. on Automatic Control 35*(9), 1058–1061.

Nuno, E., L. Basanez, and R. Ortega (2011). Passivity-based control for bilateral teleop-

eration: A tutorial. *Automatica 47*(3), 485 – 495.

Nygaard, A. (2008). High-level control system for remote controlled surgical robots. Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Ozbay, U., H. T. Sahin, and E. Zergeroglu (2008). Robust tracking control of kinematically redundant robot manipulators subject to multiple self-motion criteria. *Robotica 26*(6), 711–728.

Peng, Z. and N. Adachi (1993). Compliant motion control of kinematically redundant manipulators. *IEEE Tr. on Robotics and Automation 9*(6), 831–836.

Qu, Z. and D. M. Dawson (1995). *Robust Tracking Control of Robot Manipulators*. Piscataway, NJ, USA: IEEE Press.

Sage, H. G., M. F. D. Mathelin, and E. Ostertag (1999). Robust control of robot manipulators: A survey. *Int. J. of Control 72*(16), 1498–1522.

Sahin, H. T., U. Ozbay, and E. Zergeroglu (2006). Quaternion based robust tracking control of kinematically redundant manipulators subject to multiple self-motion criteria. In *Proc. IEEE Int. Conf. Decision and Control*, San Diego, CA, USA, pp. 6462–6467.

Sansanayuth, T., I. Nilkhamhang, and K. Tungpimolrat (2012). Teleoperation with inverse dynamics control for phantom omni haptic device. In *SICE Annual Conference*, Akita, Japan, pp. 2121–2126.

Seraji, H. (1989). Configuration control of redundant manipulators: Theory and implementation. *IEEE Tr. on Robotics and Automation 5*(4), 472–490.

Seraji, H. (1991). Task options for redundancy resolution using configuration control. In *Proc. IEEE Int. Conf. Decision and Control*, pp. 2793–2798.

Seshagiri, S. and H. K. Khalil (2005). Robust output feedback regulation of minimum-phase nonlinear systems using conditional integrators. *Automatica 41*(1), 43–54.

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control*. Cam-

bridge, MA, USA: MIT Press.

Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *J. Intelligent and Robotic Systems 3*(3), 201–212.

Siciliano, B. and O. Khatib (2008). *Springer Handbook of Robotics*. Secaucus, NJ, USA: Springer.

Silva, A. J., O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver (2009). Phantom omni haptic device: Kinematic and manipulability. In *Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Morelos, Mexico, pp. 193–198.

Spong, M. W. and M. Vidyasagar (1989). *Robot Dynamics and Control*. New York, NY, USA: John Wiley & Sons Inc.

Tatlicioglu, E., D. Braganza, T. C. Burg, and D. M. Dawson (2008). Adaptive control of redundant robot manipulators with sub-task objectives. In *Proc. American Control Conf.*, Seattle, WA, USA, pp. 856–861.

Tatlicioglu, E., D. Braganza, T. C. Burg, and D. M. Dawson (2009). Adaptive control of redundant robot manipulators with sub-task objectives. *Robotica 27*, 873–881.

Tatlicioglu, E., M. L. McIntyre, D. M. Dawson, and T. C. Burg (2006). Coordination control for haptic and teleoperator systems. Technical report, Clemson University, Clemson, SC, USA.

Tatlicioglu, E., M. L. McIntyre, D. M. Dawson, and I. D. Walker (2005). Adaptive non-linear tracking control of kinematically redundant robot manipulators with sub-task extensions. In *Proc. IEEE Int. Conf. Decision and Control*, Seville, Spain, pp. 4373–4378.

Tatlicioglu, E., M. L. McIntyre, D. M. Dawson, and I. D. Walker (2008). Adaptive non-linear tracking control of kinematically redundant robot manipulators. *Int. J. Robotics and Automation 23*(2), 98–105.

Tee, K. P. and R. Yan (2011). Adaptive operational space control of redundant robot manipulators. In *Proc. American Control Conf.*, San Francisco, CA, USA, pp. 1742–

1747.

Walker, I. (1994). Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE Tr. on Robotics and Automation 10*(5), 670–683.

Xian, B., D. M. Dawson, M. S. de Queiroz, and J. Chen (2004). A continuous asymptotic tracking control strategy for uncertain nonlinear systems. *IEEE Tr. on Automatic Control 49*(7), 1206–1211.

Xian, B., M. S. de Queiroz, D. M. Dawson, and I. D. Walker (2004). Task-space tracking control of robot manipulators via quaternion feedback. *IEEE Tr. on Robotics and Automation 20*(1), 160–167.

Yazarel, H. and C. C. Cheah (2002). Task-space adaptive control of robotic manipulators with uncertainties in gravity regressor matrix and kinematics. *IEEE Tr. on Automatic Control 47*(9), 1580–1585.

Yoshikawa, T. (1994). Analysis and control of robot manipulators with redundancy. In *Robotics Research - The First International Symp.*, Cambridge, MA, USA, pp. 735–747.

Zergeroglu, E., D. Dawson, I. D. Walker, and A. Behal (2000). Nonlinear tracking control of kinematically redundant robot manipulators. In *Proc. American Control Conf.*, Chicago, IL, USA, pp. 2513–2517.

Zergeroglu, E., D. M. Dawson, I. D. Walker, and P. Setlur (2004). Nonlinear tracking control of kinematically redundant robot manipulators. *IEEE/ASME Tr. on Mechatronics 9*(1), 129–132.

Zergeroglu, E., H. T. Sahin, U. Ozbay, and U. A. Tektas (2006). Robust tracking control of kinematically redundant robot manipulators subject to multiple self-motion criteria. In *IEEE Int. Conf. on Control App.*, Munich, Germany, pp. 2860–2865.

Zobrist, G. W. and C. Y. Ho (1996). *Progress in Robotics and Intelligent Systems*. Norwood, NJ, USA: Ablex Pub. Co.

APPENDIX A

PROOFS OF BOUNDS

A.1. Upper Bound Development for $\widetilde{Q}(t)$

In view of the similar derivations in Tatlicioglu et al. (2006) and Dasdemir and Zergeroglu (2015), the development is started by rewriting (5.9) as follows

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{M}[\ddot{\hat{\boldsymbol{J}}}^{-1}(\dot{\boldsymbol{x}}_{\boldsymbol{d}}+\boldsymbol{\alpha e})+2\dot{\hat{\boldsymbol{J}}}^{-1}(\ddot{\boldsymbol{x}}_{\boldsymbol{d}}+\boldsymbol{\alpha r}+\boldsymbol{\alpha}^2\boldsymbol{e}) \\ &\quad +\hat{\boldsymbol{J}}^{-1}(\dddot{\boldsymbol{x}}_{\boldsymbol{d}}+\boldsymbol{\alpha}^2\boldsymbol{r}+\boldsymbol{\alpha}^3\boldsymbol{e}+\boldsymbol{\alpha s}-\boldsymbol{\alpha\Gamma r})] \\ &\quad +\dot{\boldsymbol{M}}(\dot{\hat{\boldsymbol{J}}}^{-1}\dot{\boldsymbol{x}}+\hat{\boldsymbol{J}}^{-1}\ddot{\boldsymbol{x}})+\dot{\boldsymbol{N}}+\boldsymbol{M}(\boldsymbol{\Gamma s}-\boldsymbol{\Gamma}^2\boldsymbol{r})+\frac{1}{2}\dot{\boldsymbol{M}}\boldsymbol{s}+\boldsymbol{r} \end{aligned} \tag{A.1}$$

where (5.2) and (5.6) were utilized. Utilizing (2.4) and its first two time derivatives, (5.10) is rewritten as follows

$$\begin{aligned} \boldsymbol{Q_d} &= \boldsymbol{M}(\bar{\boldsymbol{f}}(\boldsymbol{x_d}))(\ddot{\hat{\boldsymbol{J}}}^{-1}(\boldsymbol{x_d},\dot{\boldsymbol{x}}_{\boldsymbol{d}},\ddot{\boldsymbol{x}}_{\boldsymbol{d}})\dot{\boldsymbol{x}}_{\boldsymbol{d}}+2\dot{\hat{\boldsymbol{J}}}^{-1}(\boldsymbol{x_d},\dot{\boldsymbol{x}}_{\boldsymbol{d}})\ddot{\boldsymbol{x}}_{\boldsymbol{d}}+\hat{\boldsymbol{J}}^{-1}(\boldsymbol{x_d})\dddot{\boldsymbol{x}}_{\boldsymbol{d}}) \\ &\quad +\dot{\boldsymbol{M}}(\bar{\boldsymbol{f}}(\boldsymbol{x_d}),\hat{\boldsymbol{J}}^{-1}(\boldsymbol{x_d})\dot{\boldsymbol{x}}_{\boldsymbol{d}})(\dot{\hat{\boldsymbol{J}}}^{-1}(\boldsymbol{x_d},\dot{\boldsymbol{x}}_{\boldsymbol{d}})\dot{\boldsymbol{x}}_{\boldsymbol{d}}+\hat{\boldsymbol{J}}^{-1}(\boldsymbol{x_d})\ddot{\boldsymbol{x}}_{\boldsymbol{d}}) \\ &\quad +\dot{\boldsymbol{N}}(\bar{\boldsymbol{f}}(\boldsymbol{x_d}),\hat{\boldsymbol{J}}^{-1}(\boldsymbol{x_d})\dot{\boldsymbol{x}}_{\boldsymbol{d}},(\dot{\hat{\boldsymbol{J}}}^{-1}(\boldsymbol{x_d},\dot{\boldsymbol{x}}_{\boldsymbol{d}})\dot{\boldsymbol{x}}_{\boldsymbol{d}}+\hat{\boldsymbol{J}}^{-1}(\boldsymbol{x_d})\ddot{\boldsymbol{x}}_{\boldsymbol{d}})). \end{aligned} \tag{A.2}$$

After adding and subtracting auxilary terms, the right–hand side of (5.11) is rewritten as follows

$$\begin{aligned} \widetilde{\boldsymbol{Q}} &= [\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}}_{\boldsymbol{d}},\ddot{\boldsymbol{x}}_{\boldsymbol{d}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})-\boldsymbol{Q_d}(\boldsymbol{x_d},\dot{\boldsymbol{x}}_{\boldsymbol{d}},\ddot{\boldsymbol{x}}_{\boldsymbol{d}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})] \\ &\quad +[\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}}_{\boldsymbol{d}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})-\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}}_{\boldsymbol{d}},\ddot{\boldsymbol{x}}_{\boldsymbol{d}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})] \\ &\quad +[\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})-\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}}_{\boldsymbol{d}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})] \\ &\quad +[\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{e},\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})-\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{0}_n,\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})] \\ &\quad +[\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{e},\boldsymbol{r},\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})-\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{e},\boldsymbol{0}_n,\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})] \\ &\quad +[\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{e},\boldsymbol{r},\boldsymbol{s},\dddot{\boldsymbol{x}}_{\boldsymbol{d}})-\boldsymbol{Q}(\boldsymbol{x},\dot{\boldsymbol{x}},\ddot{\boldsymbol{x}},\boldsymbol{e},\boldsymbol{r},\boldsymbol{0}_n,\dddot{\boldsymbol{x}}_{\boldsymbol{d}})]. \end{aligned} \tag{A.3}$$

Applying the Mean Value Theorem in Khalil (2002) to (A.3) yields

$$\begin{aligned}
\widetilde{\boldsymbol{Q}} \;=\; & \frac{\partial \boldsymbol{Q}(\boldsymbol{\sigma_1}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_1}}|_{\boldsymbol{\sigma_1}=\boldsymbol{v}_1}(\boldsymbol{x}-\boldsymbol{x}_{\boldsymbol{d}}) \\
& + \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\sigma_2}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_2}}|_{\boldsymbol{\sigma_2}=\boldsymbol{v}_2}(\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}_{\boldsymbol{d}}) \\
& + \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\sigma_3}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_3}}|_{\boldsymbol{\sigma_3}=\boldsymbol{v}_3}(\ddot{\boldsymbol{x}}-\ddot{\boldsymbol{x}}_{\boldsymbol{d}}) \\
& + \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{\sigma_4}, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_4}}|_{\boldsymbol{\sigma_4}=\boldsymbol{v}_4}(\boldsymbol{e}-\mathbf{0}_n) \\
& + \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{\sigma_5}, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_5}}|_{\boldsymbol{\sigma_5}=\boldsymbol{v}_5}(\boldsymbol{r}-\mathbf{0}_n) \\
& + \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{\sigma_6}, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_6}}|_{\boldsymbol{\sigma_6}=\boldsymbol{v}_6}(\boldsymbol{s}-\mathbf{0}_n) \qquad \text{(A.4)}
\end{aligned}$$

where $\boldsymbol{v_1} \in (\boldsymbol{x_d}, \boldsymbol{x})$, $\boldsymbol{v_2} \in (\dot{\boldsymbol{x}}_{\boldsymbol{d}}, \dot{\boldsymbol{x}})$, $\boldsymbol{v_3} \in (\ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}})$, $\boldsymbol{v_4} \in (\mathbf{0}_n, \boldsymbol{e})$, $\boldsymbol{v_5} \in (\mathbf{0}_n, \boldsymbol{r})$ and $\boldsymbol{v_6} \in (\mathbf{0}_n, \boldsymbol{s})$. The right–hand side of (A.4) can be upper bounded as follows

$$\begin{aligned}
\left\|\widetilde{\boldsymbol{Q}}\right\| \;\leq\; & \left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{\sigma_1}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \sigma_1}|_{\boldsymbol{\sigma_1}=\boldsymbol{v}_1} \right\| \|\boldsymbol{e}\| \\
& + \left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\sigma_2}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_2}}|_{\boldsymbol{\sigma_2}=\boldsymbol{v}_2} \right\| \|\dot{\boldsymbol{e}}\| \\
& + \left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\sigma_3}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_3}}|_{\boldsymbol{\sigma_3}=\boldsymbol{v}_3} \right\| \|\ddot{\boldsymbol{e}}\| \\
& + \left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{\sigma_4}, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_4}}|_{\boldsymbol{\sigma_4}=\boldsymbol{v}_4} \right\| \|\boldsymbol{e}\| \\
& + \left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{\sigma_5}, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_5}}|_{\boldsymbol{\sigma_5}=\boldsymbol{v}_5} \right\| \|\boldsymbol{r}\| \\
& + \left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{\sigma_6}, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_6}}|_{\boldsymbol{\sigma_6}=\boldsymbol{v}_6} \right\| \|\boldsymbol{s}\| \,. \qquad \text{(A.5)}
\end{aligned}$$

The partial derivative terms in (A.4) can be calculated as follows

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{\sigma_1}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_1}} = \frac{\partial [\boldsymbol{M}(\ddot{\hat{\boldsymbol{J}}}^{-1}\dot{\boldsymbol{x}}_{\boldsymbol{d}} + 2\dot{\hat{\boldsymbol{J}}}^{-1}\ddot{\boldsymbol{x}}_{\boldsymbol{d}} + \hat{\boldsymbol{J}}^{-1}\dddot{\boldsymbol{x}}_{\boldsymbol{d}})]}{\partial \boldsymbol{\sigma_1}} + \frac{\partial [\dot{\boldsymbol{M}}(\dot{\hat{\boldsymbol{J}}}^{-1}\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \hat{\boldsymbol{J}}^{-1}\ddot{\boldsymbol{x}}_{\boldsymbol{d}})]}{\partial \boldsymbol{\sigma_1}} + \frac{\partial \dot{\boldsymbol{N}}}{\partial \boldsymbol{\sigma_1}} \tag{A.6}$$

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\sigma_2}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_2}} = \frac{\partial [\boldsymbol{M}(\ddot{\hat{\boldsymbol{J}}}^{-1}\dot{\boldsymbol{x}}_{\boldsymbol{d}} + 2\dot{\hat{\boldsymbol{J}}}^{-1}\ddot{\boldsymbol{x}}_{\boldsymbol{d}})]}{\partial \boldsymbol{\sigma_2}} + \frac{\partial [\dot{\boldsymbol{M}}(\dot{\hat{\boldsymbol{J}}}^{-1}\boldsymbol{\sigma_2})]}{\partial \boldsymbol{\sigma_2}} + \frac{\partial \dot{\boldsymbol{N}}}{\partial \boldsymbol{\sigma_2}} \tag{A.7}$$

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\sigma_3}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_3}} = \frac{\partial [\boldsymbol{M}\ddot{\hat{\boldsymbol{J}}}^{-1}\dot{\boldsymbol{x}}_{\boldsymbol{d}}]}{\partial \boldsymbol{\sigma_3}} + \frac{\partial [\dot{\boldsymbol{M}}\hat{\boldsymbol{J}}^{-1}\boldsymbol{\sigma_3}]}{\partial \boldsymbol{\sigma_3}} + \frac{\partial \dot{\boldsymbol{N}}}{\partial \boldsymbol{\sigma_3}} \tag{A.8}$$

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{\sigma_4}, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_4}} = \frac{\partial [\boldsymbol{M}(\ddot{\hat{\boldsymbol{J}}}^{-1}\boldsymbol{\alpha}\boldsymbol{\sigma_4} + 2\dot{\hat{\boldsymbol{J}}}^{-1}\boldsymbol{\alpha}^2\boldsymbol{\sigma_4} + \hat{\boldsymbol{J}}^{-1}\boldsymbol{\alpha}^3\boldsymbol{\sigma_4})]}{\partial \boldsymbol{\sigma_4}} \tag{A.9}$$

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{\sigma_5}, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_5}} = \frac{\partial [\boldsymbol{M}(2\dot{\hat{\boldsymbol{J}}}^{-1}\boldsymbol{\alpha}\boldsymbol{\sigma_5} + \hat{\boldsymbol{J}}^{-1}\boldsymbol{\alpha}^2\sigma_5 - \hat{\boldsymbol{J}}^{-1}\boldsymbol{\alpha}\boldsymbol{\Gamma}\boldsymbol{\sigma_5})]}{\partial \boldsymbol{\sigma_5}} - \frac{\partial [\boldsymbol{M}\boldsymbol{\Gamma}^2\boldsymbol{\sigma_5}]}{\partial \boldsymbol{\sigma_5}} + \boldsymbol{I_n} \tag{A.10}$$

$$\frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{\sigma_6}, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_6}} = \frac{\partial [\boldsymbol{M}(\hat{\boldsymbol{J}}^{-1}\boldsymbol{\alpha}\boldsymbol{\sigma_6} + \boldsymbol{\Gamma}\boldsymbol{\sigma_6})]}{\partial \boldsymbol{\sigma_6}} + \frac{1}{2}\frac{\partial [\dot{\boldsymbol{M}}\sigma_6]}{\partial \boldsymbol{\sigma_6}}. \tag{A.11}$$

Defining $\boldsymbol{v_1} \triangleq \boldsymbol{x} - c_1(\boldsymbol{x} - \boldsymbol{x_d})$, $\boldsymbol{v_2} \triangleq \dot{\boldsymbol{x}} - c_2(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\boldsymbol{d}})$, $\boldsymbol{v_3} \triangleq \ddot{\boldsymbol{x}} - c_3(\ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}_{\boldsymbol{d}})$, $\boldsymbol{v_4} \triangleq \boldsymbol{e} - c_4(\boldsymbol{e} - \mathbf{0})$, $\boldsymbol{v_5} \triangleq \boldsymbol{r} - c_5(\boldsymbol{r} - \mathbf{0})$, $\boldsymbol{v_6} \triangleq \boldsymbol{s} - c_6(\boldsymbol{s} - \mathbf{0})$ where $c_i \in (\mathbf{0}_n, 1) \forall i = 1, 2, ..., 6$, and the following upper bounds can be written for (A.6)–(A.11)

$$\left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{\sigma_1}, \dot{\boldsymbol{x}}_{\boldsymbol{d}}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_1}} |_{\boldsymbol{\sigma_1} = \boldsymbol{v_1}} \right\| \leq \rho_1(\boldsymbol{x}) \tag{A.12}$$

$$\left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\sigma_2}, \ddot{\boldsymbol{x}}_{\boldsymbol{d}}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_2}} |_{\boldsymbol{\sigma_2} = \boldsymbol{v_2}} \right\| \leq \rho_2(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{A.13}$$

$$\left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\sigma_3}, \mathbf{0}_n, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_3}} |_{\boldsymbol{\sigma_3} = \boldsymbol{v_3}} \right\| \leq \rho_3(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}) \tag{A.14}$$

$$\left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{\sigma_4}, \mathbf{0}_n, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_4}} |_{\boldsymbol{\sigma_4} = \boldsymbol{v_4}} \right\| \leq \rho_4(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}) \tag{A.15}$$

$$\left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{\sigma_5}, \mathbf{0}_n, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_5}} |_{\boldsymbol{\sigma_5} = \boldsymbol{v_5}} \right\| \leq \rho_5(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{A.16}$$

$$\left\| \frac{\partial \boldsymbol{Q}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{e}, \boldsymbol{r}, \boldsymbol{\sigma_6}, \dddot{\boldsymbol{x}}_{\boldsymbol{d}})}{\partial \boldsymbol{\sigma_6}} |_{\boldsymbol{\sigma_6} = \boldsymbol{v_6}} \right\| \leq \rho_6(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{A.17}$$

where $\rho_i(\cdot)$ $\forall i = 1, 2, ..., 6$ are positive non–decreasing functions of $\boldsymbol{x}(t)$, $\dot{\boldsymbol{x}}(t)$ and $\ddot{\boldsymbol{x}}(t)$. After substituting (A.12)-(A.17) into (A.5), the following expression can be obtained

$$\begin{aligned} \left\| \widetilde{\boldsymbol{Q}} \right\| \leq\ & \rho_1(\|\boldsymbol{e}\|)\|\boldsymbol{e}\| + \rho_2(\|\boldsymbol{e}\|, \|\boldsymbol{r}\|)\|\dot{\boldsymbol{e}}\| + \rho_3(\|\boldsymbol{e}\|, \|\boldsymbol{r}\|, \|\boldsymbol{s}\|)\|\ddot{\boldsymbol{e}}\| \\ +\ & \rho_4(\|\boldsymbol{e}\|, \|\boldsymbol{r}\|, \|\boldsymbol{s}\|)\|\boldsymbol{e}\| + \rho_5(\|\boldsymbol{e}\|, \|\boldsymbol{r}\|)\|\boldsymbol{r}\| + \rho_6(\|\boldsymbol{e}\|, \|\boldsymbol{r}\|)\|\boldsymbol{s}\| \end{aligned} \tag{A.18}$$

where (5.2) and (5.6) were utilized. The right–hand side of (A.18) can be rewritten in a compact form as in (5.12).

A.2. Proof of Non–negativeness of $P(t)$

In this appendix, the sufficient condition of (5.21) will be obtained. Substituting (5.6) into (5.26), and then integrating $\eta(t)$ in time, the following expression is obtained

$$\begin{aligned}\int_0^t \eta(\sigma)d\sigma &= \int_0^t \boldsymbol{r}^T(\sigma)\boldsymbol{\Gamma}(\boldsymbol{Q_d}(\sigma) - \boldsymbol{\beta}\text{Sgn}(\boldsymbol{r}(\sigma)))d\sigma + \int_0^t \frac{d\boldsymbol{r}^T(\sigma)}{d\sigma}\boldsymbol{Q_d}(\sigma)d\sigma \\ &\quad - \int_0^t \frac{d\boldsymbol{r}^T(\sigma)}{d\sigma}\boldsymbol{\beta}\text{Sgn}(\boldsymbol{r}(\sigma))d\sigma \end{aligned} \tag{A.19}$$

where the symmetry of $\boldsymbol{\Gamma}$ was utilized. After evaluating second and third integrals on the right–hand side of (A.19) by parts, the following expression can be obtained

$$\begin{aligned}\int_0^t \eta(\sigma)d\sigma &= \int_0^t \boldsymbol{r}^T(\sigma)\boldsymbol{\Gamma}(\boldsymbol{Q_d}(\sigma) - \boldsymbol{\beta}\text{Sgn}(\boldsymbol{r}(\sigma)))d\sigma + \boldsymbol{r}^T(\sigma)\boldsymbol{Q_d}(\sigma)|_0^t \\ &\quad - \int_0^t \boldsymbol{r}^T(\sigma)\frac{d\boldsymbol{Q_d}(\sigma)}{d\sigma}d\sigma - \sum_{i=1}^n \beta_i \left|r_i(\sigma)\right|\,|_0^t \\ &= \int_0^t \boldsymbol{r}^T(\sigma)\boldsymbol{\Gamma}[\boldsymbol{Q_d}(\sigma) - \boldsymbol{\beta}\text{Sgn}(\boldsymbol{r}(\sigma)) - \boldsymbol{\Gamma}^{-1}\frac{d\boldsymbol{Q_d}(\sigma)}{d\sigma}]d\sigma \\ &\quad + \boldsymbol{r}^T(t)\boldsymbol{Q_d}(t) - \boldsymbol{r}^T(0)\boldsymbol{Q_d}(0) - \sum_{i=1}^n \beta_i(|r_i(t)| - |r_i(0)|). \end{aligned} \tag{A.20}$$

The right–hand side of (A.20) can be upper bounded as follows

$$\begin{aligned}\int_0^t \eta(\sigma)d\sigma &\leq \int_0^t \sum_{i=1}^n \Gamma_i \left|r_i(\sigma)\right| [Q_{di}(\sigma) - \beta_i + \frac{1}{\Gamma_i}\left|\frac{dQ_{di}(\sigma)}{d\sigma}\right|]d\sigma \\ &\quad + \sum_{i=1}^n |r_i(t)|\,(|Q_{di}(t)| - \beta_i) + \zeta_P. \end{aligned} \tag{A.21}$$

If the entries of the gain matrix $\boldsymbol{\beta}$ are chosen to satisfy (5.21), then the following expression can be obtained

$$\int_0^t \eta(\sigma)d\sigma \leq \zeta_P \tag{A.22}$$

and hence, it is clear from (5.25) that $P(t) \geq 0$.

VITA

Kamil Cetin was born in Karabuk, Turkey, on April 26, 1981. In June 2002, he received a Bachelor of Science Degree in Electrical and Electronics Engineering from Dumlupinar University, Kutahya, Turkey. In April 2007, he graduated with a Master of Science Degree in Electrical and Electronics Engineering from Dokuz Eylul University, Izmir, Turkey. After his industrial career as an electronic engineer in Makro Healthcare Company in Izmir from July 2004 to April 2008, he went on to pursue his academic career as a research assistant in the Department of Electronic Systems at Aalborg University, Aalborg, Denmark between June 2008 and September 2011. During his post master studies in Aalborg University, he also worked within ASPIRE and ISISEMD projects, collaborative FP7 projects funded by EU. Since September 2012, he has been a Ph.D student in the Electronics and Communication Engineering Doctoral Program at Izmir Institute of Technology, Izmir, Turkey. During his Ph.D. studies, he worked within one national project supported by The Scientific and Technological Research Council of Turkey. His research interests include robot manipulators, telerobotic systems, nonlinear systems, robust control, and adaptive control methods.