ADVANCED ROBOT CONTROL ALGORITHMS BASED ON MODEL-BASED NONLINEAR VELOCITY-OBSERVERS AND ADAPTIVE FRICTION COMPENSATION

XIA QING HUA
M.Sc

A DISSERTATION SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2007

ACKNOWLEDGMENTS

Time flies, four years ago, my supervisors Prof. Marcelo H. Ang Jr. and Dr Lim Ser Yong encouraged me to join the exciting and challenging robotic world. Throughout my four years's pleasant journey, I have been supported by many people. Now it is a pleasure to extend my sincere gratitude to all of those who have offered valuable help. I hope I don't forget anyone.

First and foremost, I would like to thank my supervisors, Prof. Marcelo H. Ang Jr. and Dr Lim Ser Yong, who have provided valuable guidance and suggestions in the course of my research.

Second, my research has been supported and funded by Singapore Institute of Manufacturing Technology and National University of Singapore, I am grateful for the support and the excellent research environment provided.

Third, I would also like to thank our collaboration research project advisor, Prof. Oussama Khatib from Stanford University for his guidance and great operational space framework.

This dissertation would not have been possible without the experimentation and implementation that is at its core. Therefore, my appreciation also goes to Dr Lin Wei, Lim Tao Ming, Lim Chee Wang, Dr Denny Oetomo, Mana Saedan, and Li Yuan Ping, for their support in software, hardware, and facilities.

Finally, I would like to recognize the support of my wife, my parents, and my son, their love for me and their encouragement. Special thanks to my son, who doubted my PhD qualification when I was unable to answer his funny questions, which made me realize that I need to accumulate more knowledge. Anyway, he is proud of having a father with a doctor's degree although he does not know well what PhD means.

TABLE OF CONTENTS

Page

Acknowledgments ii

Summary xi

Nomenclature xiv

List of Tables xv

List of Figures xviii

Chapters:

1. Introduction 1

    1.1 Robot Control Algorithms 1

        1.1.1 Observer-Controller 2

        1.1.2 Friction Identification and Compensation 5

        1.1.3 Force Control 6

    1.2 Objective and Summary of Contributions 8

2. Theoretical Background 9

2.1 Robot Dynamic Model 9

2.2 Robot Dynamic Model with Friction 10

2.3 Operational Space Formulation 11

2.3.1 Motion Control 13

2.3.2 Force Control 14

2.3.3 Unified Force and Motion Control 15

3. Control Algorithm 1: Observer-Controller Formulation 19

3.1 Introduction 19

3.2 Observer-Controller Formulation 20

3.2.1 Formulation of Velocity Observer 21

3.2.2 Formulation of Observer-Based Controller 22

3.3 Overall System Stability Result and Analysis 23

3.3.1 Observer Stability Analysis 24

3.3.2 Tracking Error System Stability Analysis 25

3.3.3 Controller Stability Analysis 26

3.3.4 Overall System Stability Analysis 28

3.4 Estimation Error Formulation 29

3.5 Experimental Results 31

3.5.1 Tracking Error Formulation 32

3.5.2 Experimental Results under Parametric Uncertainty 34

3.5.3 Experimental Results under Payload Variations 35
3.5.4 Quality of the Observed Velocities 38
3.6 Conclusions 46

4. Control Algorithm 2: Robust Observer-Controller Formulation 48

4.1 Introduction 48
4.1.1 Formulation of Robust Velocity Observer 48
4.1.2 Formulation of Robust Observer-Based Controller 49
4.2 Overall System Stability Result and Analysis 50
4.2.1 Lyapunov Function for Observation Error $\tilde{x}$ and $\dot{\tilde{x}}$ 51
4.2.2 Lyapunov Function for Tracking Error e 52
4.2.3 Lyapunov Function for η_p 53
4.2.4 Overall System Stability Analysis 54
4.3 Experimental Results 55
4.3.1 Experimental Results under Parametric Uncertainty 56
4.3.2 Experimental Results under Payload Variations 57
4.3.3 Quality of the Observed Velocities 60
4.4 Conclusions 67

5. Control Algorithm 3: Adaptive Friction Identification and Compensation via Robust Observer-Controller 68

5.1 Introduction 68
5.2 Adaptive Observer-Controller Formulation 68

5.2.1 Formulation of Operational Space Velocity Observer 69
5.2.2 Formulation of Friction Adaptation Law 69
5.2.3 Formulation of Operational Space Controller 72
5.3 Overall System Stability Result and Analysis 73
5.3.1 Lyapunov Function for Observation Error $\tilde{x}$, $\dot{\tilde{x}}$ and $\tilde{\theta}$ 74
5.3.2 Lyapunov Function for Tracking Error e 75
5.3.3 Lyapunov Function for η_p 76
5.3.4 Overall System Stability Analysis 77
5.4 Implementation of Friction Adaptation Law 78
5.5 Experimental Results 78
5.5.1 Friction Identification and Compensation Performance . . . 79
5.6 Conclusions 85

6. Control Algorithm 4: Adaptive Friction Identification and Compensation via Filtered Velocity 86

6.1 Introduction 86
6.2 Adaptive Controller Formulation 86
6.2.1 Formulation of Friction Adaptation Law 88
6.2.2 Formulation of Operational Space Controller 88
6.3 Overall System Stability Analysis 89
6.4 Experimental Results 91
6.4.1 Experimental Result without Friction Adaptation 91
6.4.2 Experimental Result with Friction Adaptation 93
6.5 Conclusions 96

7. Control Algorithm 5: Adaptive Friction Identification and Compensation Using Both Observed and Desired Velocity 98

7.1 Introduction 98

7.2 Adaptive Observer-Controller Formulation 99

7.2.1 Formulation of Robust Velocity Observer 99

7.2.2 Formulation of Friction Adaptation Law 99

7.2.3 Formulation of Operational Space Controller 101

7.3 Overall System Stability Result and Analysis 102

7.3.1 Lyapunov Function for Observation Error $\tilde{x}$, $\dot{\tilde{x}}$ and $\tilde{\theta}$ 103

7.3.2 Lyapunov Function for Tracking Error e 104

7.3.3 Lyapunov Function for η_p 104

7.3.4 Overall System Stability Analysis 105

7.4 Implementation of Friction Adaptation Law 106

7.5 Experimental Results 107

7.5.1 Friction Identification and Compensation Performance . . . 107

7.6 Conclusions 113

8. Control Algorithm 6: Parallel Force and Motion Control Using Observed Velocity 115

8.1 Introduction 115

8.2 Parallel Force and Motion Control 115

8.2.1 Formulation of Robust Velocity Observer 118

8.2.2 Formulation of Robust Observer-Based Controller 118
8.3 Overall System Stability Result and Analysis 119
8.3.1 Lyapunov Function for Observation Error $\tilde{x}$ and $\dot{\tilde{x}}$ 119
8.3.2 Lyapunov Function for Tracking Error e and η_p 120
8.3.3 Overall System Stability 123
8.4 Experimental Setup and Results 124
8.4.1 Damping Control Algorithm 125
8.4.2 Experimental Results 126
8.5 Conclusions 131

9. Control Algorithm 7: Parallel Force and Motion Control Using Adaptive Observer-Controller 133

9.1 Introduction 133
9.2 Parallel Force and Motion Control 133
9.2.1 Formulation of Operational Space Velocity Observer 133
9.3 Overall System Stability Result and Analysis 134
9.3.1 Lyapunov Function for Observation Error $\tilde{x}$, $\dot{\tilde{x}}$ and $\tilde{\theta}$ 135
9.3.2 Lyapunov Function for Tracking Error e and η_p 136
9.3.3 Overall System Stability Analysis 138
9.4 Experimental Results 139
9.5 Conclusions 142

10. Contributions & Future Works 143

Appendices:

A. Properties of Robot Dynamic Model 146

B. Lemmas for Stability Analysis 148

Bibliography 151

SUMMARY

This dissertation presents the development of advanced control algorithms based on model-based nonlinear velocity-observers. Several controllers have been developed and used for trajectory tracking, joint friction identification and compensation, and force control.

In Chapter 3, the first operational space observer-controller for trajectory tracking is introduced. The controller is designed in conjunction with a velocity observer using an observed integrator-backstepping procedure. With link position measurements only, the overall observer-controller system achieves semi-global exponential stability for the position, orientation and velocity tracking errors as well as velocity observation errors. Experimental results indicate that compared with the estimated velocities obtained from the backward difference algorithm used in conjunction with a lowpass filter, the observed velocities using the proposed velocity observer are less noisy. Under parametric uncertainties and payload variations, the proposed observer-controller can achieve higher position tracking accuracy than the controller employing filtered velocity.

Based on the formulation of the observer-controller introduced in Chapter 3, a robust observer-controller is presented in Chapter 4. The overall robust observer-controller system achieves semi-global exponential stability result for the position and velocity tracking errors as well as position and velocity observation errors. Under

system uncertainties, by adjusting controller gains, position and orientation estimation errors can be confined within a narrow boundary so that the variation of the observed velocity can be much smaller, hence the velocity observer becomes more robust.

To make use of the merits of the "cleaner" observed velocity proposed in Chapter 4, an observer-controller with adaptive friction compensation capability is introduced in Chapter 5. The adaptive observer-controller consists of a model-based velocity observer, a controller that is formulated in operational space, plus friction adaptation law. Experimental results using PUMA 560 indicate that the proposed adaptive controller is able to achieve higher tracking accuracy than the observer-controller without friction compensation.

In Chapter 6, an adaptive controller using filtered velocity for friction identification and compensation is presented. The overall adaptive control system can achieve a global asymptotical stability for the position and velocity tracking errors in the presence of uncertainties in friction coefficients.

In Chapter 7, an adaptive observer-controller incorporating both observed and desired velocity is presented. The adaptive controller is designed to make use of the merits of "cleaner" observed velocity and smoother desired velocity. Without velocity measurements, the overall adaptive observer-controller can achieve a semi-global asymptotic stability for the position and velocity tracking errors, and position and velocity estimation errors, with estimated friction coefficients converging asymptotically.

Both the adaptive controllers proposed in Chapters 5 and 7 can achieve higher tracking accuracy than the adaptive controller presented in 6, which verify the effectiveness of the controllers using observed velocity information.

In Chapter 8, a parallel force and motion controller employing observed velocity is proposed. The controller can achieve better control performance in both force and motion subspace as compared with the controller using filtered velocity.

Finally, an adaptive parallel force and motion controller using observed velocity is proposed in Chapter 9. The controller is able to perform friction adaptation and compensation, and at the same time, achieve better control performance in both force and motion subspaces as compared with the controller using filtered velocity and the adaptive parallel force and motion controller using observed velocity without friction adaptation capability.

NOMENCLATURE

q	Joint positions vector
$\dot{q}$	Joint velocities vector
$\ddot{q}$	Joint accelerations vector
A	Joint space inertia matrix
B	Centrifugal and Coriolis matrix in joint space
g	Gravity vector in joint space
Γ	Joint torques vector
x	Positions vector of an end-effector
$\dot{x}$	Velocities vector of an end-effector
$\ddot{x}$	Accelerations vector of an end-effector
Λ	Kinetic energy matrix
Ψ	Centrifugal and Coriolis matrix expressed in operational space
p	Gravity vector in operational space
F	Forces vector at the operational point
$\boldsymbol{F}$	Forces vector at the operational point
J	Basic Jacobian matrix
τ_f	Frictions vector in joint space
τ_{vis}	Diagonal coefficient matrix of viscous frictions in joint space
f	Frictions vector in operational space
f_{vis}	Diagonal coefficient matrix of viscous frictions in operational space
Ω	Task specification matrix, on which axes are in force and which in motion control

LIST OF TABLES

Table **Page**

3.1 Observer controller - Maximum tracking errors under parametric uncertainty 36

3.2 Observer controller - Maximum tracking errors under payload variations 36

4.1 Robust observer controller - Maximum tracking errors under parametric uncertainty 58

4.2 Robust observer controller - Maximum tracking errors under parametric uncertainty 58

5.1 Adaptive friction identification and compensation - Maximum tracking errors with adaptive friction compensation 80

5.2 Adaptive friction identification and compensation - Identified friction coefficients of each joint 81

5.3 Adaptive friction identification and compensation - Maximum tracking errors without friction compensation 85

6.1 Adaptive friction identification and compensation via filtered velocity - Tracking errors without friction compensation 92

6.2 Adaptive friction identification and compensation via filtered velocity - Tracking errors with adaptive friction compensation 93

6.3 Adaptive friction identification and compensation via filtered velocity - Identified friction coefficients of each joint (J_i) 93

7.1 Adaptive friction identification and compensation using both observed and desired velocity - Maximum tracking errors with adaptive friction compensation 109

7.2 Adaptive friction identification and compensation using both observed and desired velocity - Identified friction coefficients of each joint . . . 110

7.3 Adaptive friction identification and compensation using both observed and desired velocity - Maximum tracking errors without friction compensation 111

8.1 Tracking errors during impact moment with hard surface - Using observed velocity 126

8.2 Tracking errors during impact moment with hard surface - Using filtered velocity 132

9.1 Tracking errors during impact with hard surface - Using adaptive observer-controller 142

LIST OF FIGURES

Figure **Page**

2.1 Friction model 11

2.2 Tool frame assignment 16

3.1 Schematic Diagram of the proposed observer-controller 20

3.2 Observer controller - Tracking errors under parametric uncertainty . . 35

3.3 Observer controller - Tracking errors under payload variations 37

3.4 Observer controller - Tracking delay due to low cutoff frequency . . . 40

3.5 Observer controller - Joint velocities obtained from the velocity observer 41

3.6 Observer controller - Joint velocities obtained from filtering method . 42

3.7 Observer controller - Pseudo velocity tracking errors using the velocity observer 43

3.8 Observer controller - Pseudo velocity tracking errors using filtering method 44

4.1 Robust observer controller - Tracking errors under parametric uncertainty 57

4.2 Robust observer controller - Tracking errors under payload variations 59

4.3 Robust observer controller - Velocities obtained from the velocity observer (f=0.1Hz) 61

4.4 Robust observer controller - Velocities obtained from filtering method (f=0.1Hz) 62

4.5 Robust observer controller - Velocities obtained from the velocity observer (f=0.5Hz) 63

4.6 Robust observer controller - Velocities obtained from filtering method (f=0.5Hz) 64

4.7 Robust observer controller - Velocities obtained from the velocity observer (f=1.0Hz) 65

4.8 Robust observer controller - Velocities obtained from filtering method (f=1.0Hz) 66

5.1 Adaptive friction identification and compensation - Initial tracking errors with adaptive friction compensation 80

5.2 Adaptive friction identification and compensation - Initial identified joints friction coefficients 81

5.3 Adaptive friction identification and compensation - Tracking errors with adaptive friction compensation 82

5.4 Adaptive friction identification and compensation - Final identified joints friction coefficients 83

5.5 Adaptive friction identification and compensation - Tracking errors without friction compensation 84

6.1 Adaptive friction identification and compensation via filtered velocity - Tracking errors without friction compensation 92

6.2 Adaptive friction identification and compensation via filtered velocity - Initial tracking errors of the system 94

6.3 Adaptive friction identification and compensation via filtered velocity - Tracking errors with friction compensation 95

7.1 Adaptive friction identification and compensation using both observed and desired velocity - Initial tracking errors with adaptive friction compensation 108

7.2 Adaptive friction identification and compensation using both observed and desired velocity - Initial identified joints friction coefficients . . . 109

7.3 Adaptive friction identification and compensation using both observed and desired velocity - Tracking errors with adaptive friction compensation 110

7.4 Adaptive friction identification and compensation using both observed and desired velocity - Final identified joints friction coefficients 111

7.5 Adaptive friction identification and compensation using both observed and desired velocity - Tracking errors without friction compensation . 112

8.1 Force control response - Using observed velocity (lower graph shows response immediately after impact) 127

8.2 End-effector tracking errors - Using observed velocity 128

8.3 End-effector tracking errors - No damping control 129

8.4 Force control response - Using filtered velocity (lower graph shows response immediately after impact) 130

8.5 End-effector tracking errors - Using filtered velocity 131

9.1 Force control response - Using adaptive observer-controller 140

9.2 End-effector tracking errors - Using adaptive observer-controller . . . 141

CHAPTER 1

INTRODUCTION

1.1 Robot Control Algorithms

At present, most commercial robotic control systems use linear controllers, such as conventional PID controllers. These controllers can control a robot with moderate accuracy. However, in precision motion control, high performance force control, and applications requiring high speed in robot motion, nonlinearities cannot be ignored because they can greatly degrade the system performance. Linear control theory cannot adequately cope with nonlinearities such as dead zone or friction. Hence, linear controllers are simply not capable of providing satisfactory performance and robustness against parameter variations and many nonlinearities. In order to fully realize the capabilities of robotic systems, existing development algorithms that appear in research community should be developed, or existing approaches should be modified to improve the performance of robotic control systems.

This thesis cover the topics of the estimation of velocity and its applications in robot control, adaptive friction identification and compensation using observed velocity, and force control with the help of a velocity observer. In the following sections, the literature survey on these three aspects are presented.

1.1.1 Observer-Controller

Advanced nonlinear controllers for robot manipulators have been dealt with in great detail by many robotics researchers. Although the design of many of these controllers is elegant, their implementation is hindered by the fact that they often require the measurements of both link position and link velocity, even for the implementation of a simple PD controller. But most robot manipulators are only equipped with link position sensors (e.g. optical encoder) as they give us very accurate measurements of joint position. Measurement of the link velocity is possible by using a velocity sensor, e.g. a tachometer, but the measurements are often contaminated by noise. This will reduce the dynamic performance of the manipulator, since, in practice, the values of the controller gains are limited by the noise present in the velocity measurements [1]. Besides, the addition of tachometers makes the whole robotic system more complex.

To provide for a means of incorporating link velocity information into a control algorithm, most researchers resort to filtering (e.g. a backwards difference algorithm used in conjunction with a low pass filter) of the joint position information to estimate the link velocity. However, this approach cannot guarantee the closed-loop stability of the overall system. Moreover, it ignores the dynamic effect because of the position linearization across each sampling interval.

To overcome this drawback, some researchers have proposed advanced robot controllers that do not rely on link velocity measurements. For example, Nicosia et al. [2] designed an exact knowledge model-based observer-controller that yielded an asymptotic stability result for the closed loop observer-tracking error system. In addition to Nicosia et al. [3], experimental work was presented to verify the feasibility of using high gain observers in conjunction with several different control techniques. Based

on the mechanical model of the robot manipulator and link position measurements, the estimates of link velocity have been achieved using a nonlinear, second-order observer [4, 5].

For the compensation of system uncertainty, Canudas de Wit et al. developed variable structure model-based observers for the design of adaptive [6] and robust [7] controllers. Erlic et al. designed reduced-order observers for use in an exact knowledge-based controller [8] and an adaptive controller [9] that were shown to have applications for impedance control [10].

Zhu et al. presented a variable structure controller that utilized a model-based observer with fixed parameter estimates [11]. Combing the controller development with the observer design, Berghuis et al. developed a robust control [12] that utilized the velocity estimates from a linear high-gain observer and an observer-controller combination [13, 14] for robotic manipulators based on Lyapunov and passivity type arguments.

Using an observed backstepping approach, Lim et al. [15] presented theoretical development and experimental results for an output feedback position tracking robot controller that incorporated an exact knowledge model-based velocity observer, this controller can achieve a semi-global exponential stability (SGES) result for the link position tracking error and the velocity observation error. Semi-global means that controller gains must satisfy certain condition in order to make a system stable. Exponential stability means that the link position tracking error and the velocity observation error will approach zero exponentially. The controller has then been further extended to the robot manipulator models that include actuator dynamics [16–18]. Hsu et al. [19] proposed a variable structure adaptive control scheme without

velocity measurements. Yuan et al. [20] applied a filtering scheme to the position signal to create a new signal that is used to design a robust controller consisting of a linear feedback term and a nonlinear feedforward term with fixed parameter estimates. Burg et al. [21] used a similar filtering scheme to develop an adaptive controller that yielded a semi-global asymptotic stability result for the link position tracking error. Kaneko et al. [22] used repetitive and adaptive motion control schemes for rigid-link robot manipulators.

The above mentioned controllers were designed in joint space. However, in many robotic applications, tasks are defined in operational space [23]. The basic idea in the operational space approach is to control motions and contact forces through the use of control forces that act directly at the end-effector. Task specification for motion and contact forces, dynamics, and force sensing feedback, are most closely linked to the end-effector's motion. Thus, high performance control of motions and contact forces requires the formulation of controllers directly in operational space. Many works have been done on the formulation of controllers in operational space based on the availability of actual link velocity measurements [24, 25]. But it seems that little work has been done with regards to the development of observer-controllers in operational space. Only recently, a method for task space position tracking via quaternion feedback was presented in [26]. Pagilla et al. designed an adaptive observer-controller which is shown to be semi-global asymptotically stable [27]. An observer-controller design for task space tracking control using unit quaternion was proposed in [28].

1.1.2 Friction Identification and Compensation

Due to mechanical contact, friction will present in robot servo-mechanisms and causes tracking lags, steady state errors and undesired stick-slip motion. In most industrial robots, motor torques are transformed through gears to links. The dynamic behavior of robots is significantly affected by gears, which introduce significant friction.

Precise control of robot manipulators in the presence of friction-related effects is a very challenging task. The coefficients of the various friction-related effects are usually very difficult to measure. In addition, the friction-related coefficients usually exhibit time-dependent characteristics; therefore, effective compensation for friction effects via adaptive control seems are well motivated. Friction in robot manipulators is one of the major limitations in achieving high precision motion control. It has many diverse aspects giving rise to control problems such as steady state errors, tracking errors, limit cycles, and stick-slip. If not compensated properly, it may cause stability problems. For these reasons, friction modelling, identification, and compensation have been addressed by a number of researchers. For example, a dynamic friction compensator was derived for position-force visual servoing [29], two discrete-time models of friction for the purpose of fixed-step numerical simulations were proposed in [30], an adaptive controller that considers both static and dynamic friction effects was proposed in [31], a robust adaptive friction compensation in the presence of bounded disturbances and/or modelling uncertainties was addressed in [32], and a variable structure control scheme for the robot with nonlinear friction and dynamic backlash was investigated in [33]. To deal with stick-slip friction, an integrated adaptive-robust approach along with a smooth friction compensation strategy was presented in [34],

and a robust nonlinear controller was designed for the regulation of a rigid robot with internal joint stick-slip friction [35]. In terms of flexible manipulators, some researchers investigated limit cycles phenomena in flexible joint mechanisms [36], and friction compensation algorithm based on LuGre's model was proposed in [37]. To achieve precise tracking control, virtual friction field and iterative learning control architecture to compensate friction effect were developed [38], [39].

1.1.3 Force Control

Research on robot force control has flourished in the past two decades. Such a wide interest is motivated by the general desire of providing robotic systems with enhanced sensory capabilities. The purpose of force control could be quite diverse, such as applying a controlled force needed for a manufacturing process (e.g. deburring or grinding), pushing an external object using a controlled force, or dealing with geometric uncertainty by establishing controlled contacts (e.g. in assembly).

The two most common basic approaches to force control are Hybrid force/position control, and impedance control. Both approaches can be implemented in many different ways. Hybrid control is based on the decomposition of the workspace into purely motion controlled directions and purely force controlled directions [23, 40]. Many tasks, such as inserting a peg into a hole, and force-controlled deburring are described in the ideal case by such task decomposition [41]. Impedance control, on the other hand, does not regulate motion or force directly, but instead regulates the ratio of force to motion, which is the mechanical impedance [42–44]. Both Hybrid control and impedance control are highly idealized control architectures. The decomposition into purely motion controlled and purely force controlled directions is

based on the assumption of ideal constraints, i.e., rigid and frictionless contacts with perfectly known geometry. In order to overcome some of the fundamental limitations of the basic approaches, the following improvements have been proposed: The combination of force and motion control in a single direction has been introduced in the Hybrid control approach [45, 46], where a feedforward motion command was injected in a force controlled direction. Parallel force/position control schemes were also presented in [46–49], where the force and motion commands coexist in the force control direction, with force command dominating the force control performance.

In [50], hybrid and impedance control was combined into hybrid impedance control to simultaneously regulate impedance and either force or motion.

To deal with uncertainty, some adaptive and robust force/position controllers were presented in [51, 52]. An adaptive compliant control algorithm was presented in [53]. An adaptive parallel force/position control scheme was presented in [54]. Because sliding mode control is insensitive against system perturbation and modeling uncertainties, recently, some researchers used this control scheme for robotic force control, e.g., a force controller with an inner-loop position-based sliding mode controller, and an outer-loop force compensator was presented in [55], a sliding mode controller for a robot in contact with an isotropic and homogenous environment was presented in [56].

All the above mentioned force and position control schemes require full-state feedback of the contact force and the joint position and velocity. A problem exist, however, for those robots having only joint encoders or resolvers for measuring positions, but no tachometers for measuring joint velocities. Recently, an output feedback parallel force/position regulator for a robot manipulator was presented in [57, 58], and the

use of a nonlinear observer does not compromise the tracking and steady-state performance of the system, and thus presents a valid solution when joint velocities are not available.

1.2 Objective and Summary of Contributions

Filtered velocity has two limitations: first, the introduction of a low-pass filter will cause tracking delay; second, even with the help of a low-pass filter, the noise in filtered velocity cannot be completely removed. The objective of the Ph.D research is to design a velocity observer without use of a low-pass filter, and at the same time, to minimize the noise level in observed velocity.

Based on this objective, two observer-controllers have been developed. Experimental results indicate that, the noise level in observed velocity is lower than the filtered velocity. Under parametric uncertainty and payload variations, the proposed observer-controllers can achieve higher position tracking accuracy than the controller employing filtered velocity.

Encouraged by the performance of the observer-controllers, two adaptive observer-controllers have been designed to perform friction identification and compensation function. Experimental results indicate that both the proposed adaptive observer-controller are able to achieve much higher tracking accuracy than the observer-controller without friction compensation.

Velocity observers can also help in force control applications. Two parallel force and motion controllers using observed velocity have been developed. Experimental results also show their better control performance in both force and motion subspace as compared with the controller using filtered velocity.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Robot Dynamic Model

The dynamic equation of a robot in free motion (no contact with environment) can be expressed in joint space

$$A(q)\ddot{q} + B(q,\dot{q})\dot{q} + g(q) = \Gamma \tag{2.1}$$

where Γ is the $n \times 1$ vector of joint torques, q is the $n \times 1$ vector of joint positions, $A(q)$ is the $n \times n$ inertial matrix, $B(q,\dot{q})$ is the $n \times n$ centrifugal and Coriolis matrix, and $g(q)$ is the $n \times 1$ vector of gravitational torques. For a non-redundant robot, the corresponding end-effector equation of motion (in operational space) can be expressed as [23]

$$\Lambda(x)\ddot{x} + \Psi(x,\dot{x})\dot{x} + p(x) = F \tag{2.2}$$

where F is the $n \times 1$ operational forces vector, x is the $n \times 1$ vector describing the position and orientation of the end-effector, $\Lambda(x)$ is the $n \times n$ kinetic energy matrix, $\Psi(x,\dot{x})$ is the $n \times n$ centrifugal and Coriolis matrix expressed in operational space, and $p(x)$ is the $n \times 1$ vector of gravitational forces. In the nonsingular region and the domain of one to one mapping of a robot, the relationships between the components

of the joint space dynamic model and those of the operational space dynamic model can be expressed as [59]:

$$\begin{array}{lcl} \Lambda(x) & = & J^{-T}(q)A(q)J^{-1}(q) \\ \Psi(x,\dot{x}) & = & J^{-T}(q)\left[B - A(q)J^{-1}(q)\dot{J}(q,\dot{q})\right]J^{-1}(q) \\ p(x) & = & J^{-T}(q)g(q) \\ \Gamma & = & J^{T}(q)F \end{array} \tag{2.3}$$

where $J(q)$ is the basic Jacobian of the robot.

These equations apply when the robot is operating in a non-singular region and the mapping between joint (q) and operational space (x) coordinates is one to one. These conditions are necessary for the operational space (x) to be considered generalized coordinates.

2.2 Robot Dynamic Model with Friction

In the presence of joints friction, Eq (2.1) can be written as:

$$A(q)\ddot{q} + B(q,\dot{q})\dot{q} + g(q) + \tau_f = \Gamma \tag{2.4}$$

where τ_f is the $n \times 1$ vector of friction torques.

In this thesis, the following friction model is used (as shown in Fig. 2.1) [60]:

$$\tau_f = \tau_{vis}\dot{q} + \left[\tau_{cou} + \tau_{sti}\exp(-\tau_{dec}\dot{q}^2)\right]\mathrm{sgn}(\dot{q}) \tag{2.5}$$

where τ_{vis} denotes the diagonal coefficient matrix of viscous friction; τ_{cou} denotes the Coulomb friction-related diagonal coefficient matrix; τ_{sti} denotes the static friction-related diagonal coefficient matrix; τ_{dec} is a positive diagonal coefficient matrix corresponding to Stribeck effect; and the signum function $\mathrm{sgn}(\cdot)$ is defined as:

$$\mathrm{sgn}(\dot{q}) = \begin{cases} +1, & \dot{q} > 0 \\ 0, & \dot{q} = 0 \\ -1, & \dot{q} < 0 \end{cases} \tag{2.6}$$

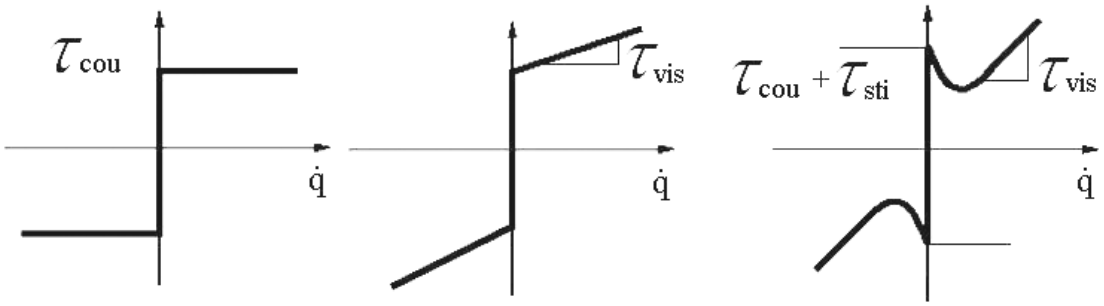

Figure 2.1: Friction model

In the presence of joint friction, the corresponding end-effector equation of motion in operational space can be expressed as [23]:

$$\Lambda(x)\ddot{x} + \Psi(x,\dot{x})\dot{x} + p(x) + f = F \tag{2.7}$$

where f is the $n \times 1$ friction vector expressed in operational space.

The relationships between the joint space and operational space friction vector can be expressed as:

$$f = J^{-T}(q)\tau_f \tag{2.8}$$

2.3 Operational Space Formulation

The Operational Space Formulation [61] is a control approach where free motion and contact forces are expressed in operational space (Cartesian space as seen from the end-effector or tool), and transformed into operational space forces that includes the dynamic effects of the manipulator. This force is then transformed into equivalent torque values to be exerted by each joint to result in the desired operational forces at the end effector.

The force is obtained by multiplying the mass/inertia of the robot with the desired acceleration. The mass/inertia of the robot can be obtained by experiments as described in [62, 63] and can also be verified in [64]. In free motion, the desired acceleration is generated by the control law that minimizes the error between the desired and the actual trajectories. Other dynamic parameters can be included into the generated force, such as the gravity, Coriolis, and Centrifugal forces to better model the dynamics of the robot.

An obvious advantage of this formulation is that it is a very natural framework for combined position and force control, which is used when the end effector comes into contact with the environment. Forces are generally expressed in the Cartesian space, and having free motion generated as forces in the Cartesian space provides an elegant framework for a hybrid motion/force control.

The total force f is therefore a combination of the force for free motion control and force for constrained motion (force control). It is then converted to joint torques by

$$\begin{aligned} \tau &= \mathbf{J}^T f + \mathcal{N}^T \tau_0 \\ \mathcal{N} &= [I - J^{\#} J] \end{aligned} \tag{2.9}$$

where τ is the joint torque command vector, and J is the Jacobian matrix. $\mathcal{N}$ and τ_0 are used to control the null space motion of the Jacobian and is useful when the manipulator is redundant with respect to the task. They will be elaborated in the later parts of the chapter. $J^{\#}$ is a generalized inverse of the J matrix.

2.3.1 Motion Control

The control law to generate the required force is computed from the required acceleration, f^*_{motion}:

$$f^*_{motion} = I\ddot{x}_d - K_{vm}(\dot{x} - \dot{x}_d) - K_{pm}(x - x_d) \tag{2.10}$$

f^*_{motion} is then multiplied with the inertia matrix $\hat{\Lambda}$, and added with the Coriolis and Centrifugal forces $\hat{\mu}$ and gravitational vector $\hat{p}$, to yield the required force:

$$f_{motion} = \hat{\Lambda}(x) f^*_{motion} + \hat{\mu}(x, \dot{x}) + \hat{p}(x) \tag{2.11}$$

The operational space control can be compared with computed-torque control in joint space, which is described as:

$$\tau = \hat{A}(q)\ddot{q} + \hat{b}(q, \dot{q}) + \hat{g}(q) \tag{2.12}$$

where $\hat{A}$ is the joint space inertial matrix of the manipulator, $\hat{b}(q, \dot{q})$ is the Coriolis and centrifugal vector, and $\hat{g}$ is the gravity compensation vector in joint space. Methods of dynamics identification can be found in [62] and [63, 65]. In the work involved in this dissertation, we use the PUMA 560 manipulator as a test bed. The dynamic model of PUMA 560 is obtained from [63].

The " ˆ " above the parameter represents our estimate of actual dynamic parameters. The actual dynamic model of the robot is represented by:

$$f_{motion} = \Lambda(x)\ddot{x} + \mu(x, \dot{x}) + p(x) \tag{2.13}$$

2.3.2 Force Control

As the robot end-effector is in contact with the environment, reaction forces and moments are generated at the end-effector. These forces/moments are then transmitted to the robot joints where the driving torques can be generated to impose the desired contact forces/moments to the robot environment.

The force control in operational space can be transformed to the robot joint space by the same transformation as the operational space motion control.

The operational space force applied at the end-effector can be expressed as

$$f_{force} = \hat{\Lambda}(x) f^*_{force} + \hat{\mu}(x, \dot{x}) + \hat{p}(x) + f_{contact} \tag{2.14}$$

where

$$f^*_{force} = K_{pf}(f_d - f_{contact}) + K_{if} \sum (f_d - f_{contact}) \tag{2.15}$$

is the control law and $f_{contact}$ is the force exerted on the environment and is related to the force sensor reading, f_{sensor}, by

$$f_{contact} = -f_{sensor} \tag{2.16}$$

Note that the force sensor reading is the force exerted by the environment on the end-effector.

The $\hat{\mu}$ and $\hat{p}$ vectors are the Coriolis and centrifugal vector and gravitational vector as defined in motion control. With contact to the environment, the actual dynamic model becomes

$$f_{force} = \Lambda(x)\ddot{x} + \mu(x, \dot{x}) + p(x) + f_{contact} \tag{2.17}$$

2.3.3 Unified Force and Motion Control

In unified force and motion control, operational space is divided into two subspaces: force control and motion control subspaces. We need to specify which degrees-of-freedom will be assigned for force and motion control. Appropriate control algorithms are then applied respectively.

The resulting force and motion control is done by selecting the desired force or motion response of the robot and adding them together to get the effective robot response (Fig. 2.2). This is expressed as

$$f = f_{motion} + f_{force} \tag{2.18}$$

where

$$f_{motion} = \hat{\Lambda}(x)\Omega f^*_{motion} + \hat{\mu}(x, \dot{x}) + \hat{p}(x) \tag{2.19}$$

and

$$f_{force} = \hat{\Lambda}(x)\bar{\Omega} f^*_{force} - f_{sensor}. \tag{2.20}$$

f^*_{motion} and f^*_{force} are the force applied for motion and force control respectively (defined in (2.10), (2.15)). Ω and $\bar{\Omega}$ are the selection matrices to switch the application between force or motion whichever is desired and to specify the direction of application. $\hat{\mu}(x, \vartheta)$ represents the estimated Coriolis and centrifugal forces, and $\hat{p}(x)$ the estimated Gravitational force, which are the same as those defined for force and motion control, and are therefore only included once.

To specify the selection matrices, consider a reference Frame $\{P\}$ at the operational point that is always parallel to the base (global) reference Frame $\{O\}$ (see Fig. 2.2). We then consider an operational space (tool) force Frame $\{T\}$ whose orientation is obtained from Frame {P} by the 3×3 rotation matrix ${}^{P}R_{T}$. Frame $\{T\}$ is attached

to the end-effector while the origin of Frame $\{P\}$ translates with the operational point and always coincides with the origin of Frame $\{T\}$.

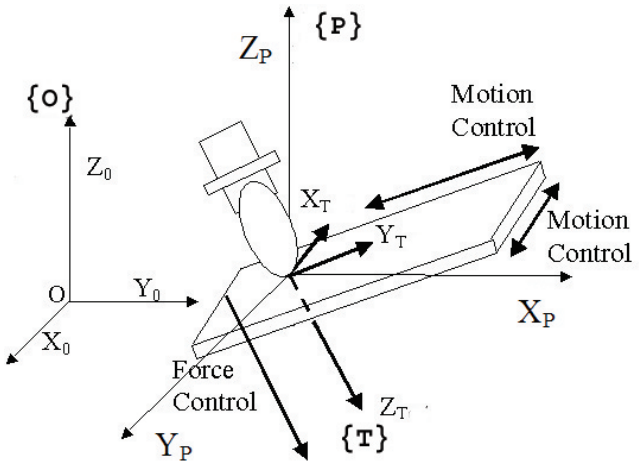

Figure 2.2: Tool frame assignment

The generalized task specification matrices Ω is then defined as

$$\Omega = \begin{pmatrix} {}^{T}R_{P}^{T}\, S_F\, {}^{T}R_{P} & 0 \\ 0 & {}^{T}R_{P}^{T}\, S_M\, {}^{T}R_{P} \end{pmatrix} \tag{2.21}$$

where

$$\mathcal{S}_{\mathcal{F}} = \begin{pmatrix} \sigma_{FX} & 0 & 0 \\ 0 & \sigma_{FY} & 0 \\ 0 & 0 & \sigma_{FZ} \end{pmatrix} \tag{2.22}$$

$$\mathcal{S}_{\mathcal{M}} = \begin{pmatrix} \sigma_{MX} & 0 & 0 \\ 0 & \sigma_{MY} & 0 \\ 0 & 0 & \sigma_{MZ} \end{pmatrix} \tag{2.23}$$

and

σ_{FX}, σ_{FY}, σ_{FZ}, σ_{MX}, σ_{MY}, σ_{MZ} are binary values where "1" signifies application of free motion (motion control) along the corresponding axis and "0" for constraint motion (force control) along the corresponding axis.

Eq 2.21 was derived to consistently match the frames that different components are expressed in. $\mathcal{S}_\mathcal{F}$ and $\mathcal{S}_\mathcal{M}$ are expressed in the end-effector frame (Frame{T}. However, f^*_{motion} and f^*_{force} are all expressed in Frame{0}, consistent with system dynamics expressed in Frame {P} (which is parallel to Frame{0}). Therefore, they have to be first transformed to Frame{T} (by ${}^TR_P$) before the application of $\mathcal{S}$. They are then transformed back to Frame{P} by ${}^PR_T$ after the application of $\mathcal{S}$. $\bar{\Omega}$ is obtained using $\bar{\mathcal{S}}_\mathcal{F}$ and $\bar{\mathcal{S}}_\mathcal{M}$ which are the complements of $\mathcal{S}_\mathcal{F}$ and $\mathcal{S}_\mathcal{M}$.

The equations are reproduced below for convenience.

$$f^*_{motion} = I\ddot{x}_d - K_{vm}(\dot{x} - \dot{x}_d) - K_{pm}(x - x_d) \tag{2.10}$$

$$f^*_{force} = K_{pf}(f_d - f_{contact}) + K_{if}\sum(f_d - f_{contact}) \tag{2.15}$$

These commands are then compensated for the dynamic effect according to the dynamic model of the manipulator and assigned to its associated degrees-of-freedom. The resulting forces are added together to form the total force to be displayed at the end-effector.

$$f_{motion} = \hat{\Lambda}(x)\Omega f^*_{motion} + \hat{\mu}(x, \dot{x}) + \hat{p}(x) \tag{2.19}$$

$$f_{force} = \hat{\Lambda}(x)\bar{\Omega} f^*_{force} - f_{sensor} \tag{2.20}$$

$$f = f_{motion} + f_{force} \tag{2.18}$$

The force is then converted to joint space command to be sent to the respective joints, by:

$$\tau = J^T f + \mathcal{N}^T \tau_0 \tag{2.9}$$

The operational space formulation is capable of unified force and motion control. Desired contact force with the environment and the desired end-effector motion is

generated by task specification. The task specification also includes the description of which degrees-of-freedom are to be assigned to force control and which to motion control. The control law that compares the input and the generated output forces and motion at the end-effector provides the actuation command in task space required to close the tracking error. For more details in unified motion and force control, please refer to [61].

CHAPTER 3

CONTROL ALGORITHM 1: OBSERVER-CONTROLLER FORMULATION

3.1 Introduction

The control objective is to develop an end-effector position and orientation tracking controller in operational space so that the end-effector position and orientation tracking errors and velocity observation (velocity estimation) errors can be driven to zero with link position information only. Based on the assumption that the exact model is available, we design a velocity observer to estimate the robot end-effector velocities, the observed velocities will be used for the development of a force input controller. The force generated by the force input controller will then be used to drive the velocity observer and to generate the corresponding torque that is used to drive the robot. The schematic diagram of the proposed observer-controller is shown in Fig. 3.1. The end-effector position and orientation information can be obtained from link position measurements by forward kinematics (only for serial link mechanism) of the robot.

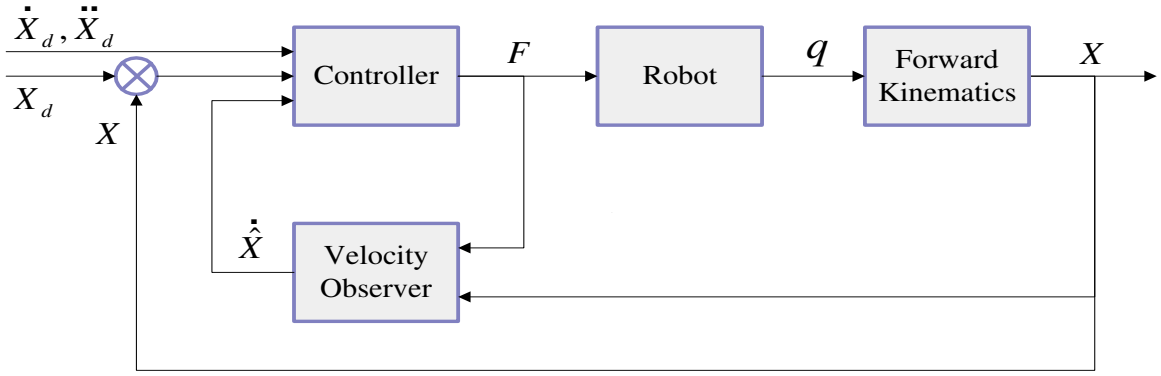

Figure 3.1: Schematic Diagram of the proposed observer-controller

3.2 Observer-Controller Formulation

To begin the controller development, we define the $n \times 1$ end-effector position and orientation tracking error as:

$$e = x_d - x \tag{3.1}$$

where x_d represents the desired end-effector position and orientation trajectory. For the system to be stable, the requirement for the trajectory is that x_d and its first and second derivatives are all bounded functions of time. More specifically, for the desired velocity, we may assume that:

$$\|\dot{x}_d\| \leq \zeta_d \qquad \forall t \tag{3.2}$$

where ζ_d is a known positive scalar constant. This property will be used for stability analysis in Sec 3.3.

3.2.1 Formulation of Velocity Observer

To estimate the end-effector velocity, we utilize the following second order velocity observer:

$$\dot{\hat{x}} = y + k\tilde{x}, y(0) = -k\tilde{x}(0) \tag{3.3}$$

$$\dot{y} = \Lambda^{-1}\Big[F - \Psi(x, \dot{\hat{x}})\dot{\hat{x}} - p(x)\Big] \tag{3.4}$$

where

$$\tilde{x} = x - \hat{x} \tag{3.5}$$

$\dot{\hat{x}}$ and $\hat{x}$ are the estimates of a robot end-effector velocity and position using the proposed velocity observer.

The motivation of this controller is to mimic the dynamic behavior of a robot as indicated by (2.2). Here x is the position and orientation of the end-effector, which can be obtained from the forward kinematics of a robot.

The variable $\dot{y}$ is an $n \times 1$ auxiliary variable that can be viewed as acceleration. When a robot starts to move from standstill, we can set $y(0) = -k\tilde{x}(0)$ so that the estimated initial end-effector velocity $\dot{\hat{x}}(0)$ is 0. F is the force control input to the observer, it is the force generated by the controller (as will be indicated in (3.8)). Here k is a positive scalar constant defined by:

$$k = \frac{1}{m_1}\left[\zeta_c\zeta_d + \zeta_c k_0 + \zeta_c k_s k_0 + k_s + 2k_n\right] \tag{3.6}$$

where k_0, k_s and k_n are positive scalar control gains, ζ_c and ζ_d are defined in (A.4) and (3.2) respectively. m_1 is a known positive scalar constant as defined in (A.1). The values of all these gains are difficult to estimate due to the complexity of a robot

dynamic model. However, we can still choose controller gains that are large enough to make a system stable, as indicated by Theorem 1 in this chapter. The inequality expressed (A.1) will be used in system stability analysis.

To facilitate the subsequent analysis, (3.5) is differentiated with respect to time to form the following velocity observation error:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \tag{3.7}$$

3.2.2 Formulation of Observer-Based Controller

Based on the structure of the above observer, and the subsequent stability analysis, we propose the following controller to generate the required driving force:

$$F = (k_s + k_{nd})\eta_p + w_e \tag{3.8}$$

where k_{nd} is a positive controller gain defined as:

$$k_{nd} = 2k_n + \zeta_c k_0 + (k_s m_2 + k m_2)^2 k_n \tag{3.9}$$

m_2 is a known positive scalar constant as defined in (A.1). The inequality expressed (A.1) will be used in system stability analysis. The $n \times 1$ observed filtered tracking error signal η_p and w_e are defined as:

$$\eta_p = \dot{x}_d + k_s e - \dot{\hat{x}} \tag{3.10}$$

$$w_e = \Lambda(x)[\ddot{x}_d + k_s(\dot{x}_d - \dot{\hat{x}})] + \Psi(x, \dot{\hat{x}})(\dot{x}_d + k_s e) + p(x) \tag{3.11}$$

The force command F will be used in the observer indicated by (3.4). And the torque commands for driving the robot can be obtained by:

$$\Gamma = J^T F$$

3.3 Overall System Stability Result and Analysis

For the observer and controller presented in the previous section, if the exact model of a robot is known, then the position and orientation tracking error defined in (3.1) is semi-global exponentially stable according to the following theorem:

Theorem 1 *Provided the observer-controller gains satisfy the following sufficient conditions:*

$$\begin{array}{rcl} k_s & > & \frac{1}{k_n} \\ k_0 & > & \sqrt{\frac{\lambda_2}{\lambda_1}} \parallel err(0) \parallel \end{array} \tag{3.12}$$

the closed-loop observation tracking error system is exponentially stable:

$$\parallel err(t) \parallel \leq \sqrt{\frac{\lambda_2}{\lambda_1}} \parallel err(0) \parallel e^{-\lambda t} \tag{3.13}$$

where

$$\begin{array}{rcl} \lambda_1 & = & \min\{m_1, 1\} \\ \lambda_2 & = & \max\{m_2, 1\} \\ \lambda_3 & = & k_s - \frac{1}{k_n} \\ \lambda & = & \frac{\lambda_3}{\lambda_2} \end{array} \tag{3.14}$$

and

$$err = \begin{bmatrix} \eta_p^T & e^T & \dot{\tilde{x}}^T \end{bmatrix}^T \in \Re^{3n} \tag{3.15}$$

m_1 and m_2 are known positive scalar constants that satisfy the inequality indicated by (A.1) in Appendix A. The values of m_1 and m_2 will affect the stability of the system, and k_0 needs to be large enough, as indicated by (3.12). The values of m_2, k_s, and k_n will affect the convergence speed of err, as indicated by (3.13) and (3.14). k_n, k_s and k_0 are the controller gains indicated by (3.6) and (3.9). The gains need to satisfy (3.12), thus the observer-controller achieves semi-global stability. Eq (3.13) indicates that $err(t)$ approach zero exponentially. Hence, the whole system is a semi-global exponentially stable system.

It is noted from (3.13) that, with $err(0) = 0$, $err(t) = 0$ for all t. However, if $err(0)$ is not zero initially, as indicated by this theorem, $err(t)$ will approach zero as t approaches infinity. But (3.13) only holds with perfect knowledge of robot dynamics (as shown in the proof). It is therefore practical to choose gains indicated by (3.12) for robustness, assuming a bound for $err(0)$.

In the stability analysis, we will break up this analysis into three parts, i.e., observer stability analysis, tracking error system stability analysis, and controller stability analysis. We will then combine the three analysis to complete the overall stability proof.

To prove the stability result indicated by *Theorem 1*, we define the following composite Lyapunov function:

$$V = V_0 + V_1 + V_2 \tag{3.16}$$

and $\dot{V}$ can be obtained by:

$$\dot{V} = \dot{V}_0 + \dot{V}_1 + \dot{V}_2 \tag{3.17}$$

V_0, V_1, and V_2 are used for observer stability analysis, tracking error system analysis, and controller stability analysis respectively.

3.3.1 Observer Stability Analysis

To analyze the stability of the above observer error system, the following sub-Lyapunov function is defined:

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} \tag{3.18}$$

From second-order observer shown in (3.3), we differentiate it and then substitute the expression for $\dot{y}$ in (3.4) to give:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} + p(x) - k\Lambda(x)\dot{\tilde{x}} = F \tag{3.19}$$

We use properties of dynamic model (in Appendix A) in the subsequent derivation.

Subtract (3.19) from the robot dynamic model (2.2), use (A.3) and (3.7) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x,\dot{x})\dot{\tilde{x}} + \Psi(x,\dot{\tilde{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} = 0 \tag{3.20}$$

Differentiate V_0 in (3.18) with respect to time and combine (3.20) to get:

$$\begin{aligned} \dot{V}_0 &= \dot{\tilde{x}}^T \left[-\Psi(x,\dot{x})\dot{\tilde{x}} - \Psi(x,\dot{\tilde{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} \right] \\ &\quad + \frac{1}{2}\dot{\tilde{x}}^T \dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i \tilde{x} \end{aligned} \tag{3.21}$$

Utilizing (3.20) and dynamic property (A.2) to get:

$$\dot{V}_0 = -\dot{\tilde{x}} \left[\Psi(x,\dot{\tilde{x}}) + k\Lambda(x) \right] \dot{\tilde{x}} \tag{3.22}$$

Properties (A.1) and (A.4) are used to get the upper bound of $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c \|\dot{\tilde{x}}\| - k m_1 \right) \|\dot{\tilde{x}}\|^2 \tag{3.23}$$

Substituting $\dot{\tilde{x}}$ from (3.10) into (3.23), and using (3.2) results in a new upper bound for $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c \zeta_d + \zeta_c \|\eta_p\| + \zeta_c k_s \|e\| - k m_1 \right) \|\dot{\tilde{x}}\|^2 \tag{3.24}$$

3.3.2 Tracking Error System Stability Analysis

From the form of (3.24), we are motivated to design a controller which ensures that the $\|e\|$ and $\|\eta_p\|$ terms in (3.24) are both driven to zero; hence, we are motivated to develop tracking error systems and the corresponding sub-Lyapunov functions to facilitate the goal. The position tracking error system can be formed by differentiating (3.1) with respect to time to yield:

$$\dot{e} = \dot{x}_d - \dot{x}$$

Since $\dot{x}$ is not measurable, we use the estimated term $\dot{\hat{x}}$ to eliminate $\dot{x}$ and get the following equation:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} - \dot{\tilde{x}} \tag{3.25}$$

Add and subtract a fictitious controller [66] to the right-hand side of (3.25) to yield:

$$\dot{e} = \dot{x}_d - [\dot{x}_d + k_s e] + [\dot{x}_d + k_s e] - \dot{\hat{x}} - \dot{\tilde{x}} \tag{3.26}$$

where k_s is a positive controller gain defined in (3.6).

Equ (3.26) can be simplified by utilizing (3.10) and get:

$$\dot{e} = -k_s e + \eta_p - \dot{\tilde{x}} \tag{3.27}$$

To analyze the stability of the above position tracking error system, the following sub-Lyapunov function is defined:

$$V_1 = \frac{1}{2} e^T e \tag{3.28}$$

Differentiating V_1 with respect to time and get:

$$\dot{V_1} = e^T \dot{e} \tag{3.29}$$

The upper bound for the time derivative of V_1 along (3.27) is given by:

$$\dot{V_1} \leq -k_s \|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{3.30}$$

3.3.3 Controller Stability Analysis

From the form of (3.30) and the fact that the form of (3.24) indicates that $\|\dot{\tilde{x}}\|$ can be driven to zero, we are motivated to design a force input controller which ensure that η_p can be driven to zero.

The tracking error system for η_p can be formed by differentiating (3.10) with respect to time, multiplying both sides of the resulting expression by $\Lambda(x)$, and substituting $\ddot{\tilde{x}}$ obtained from (3.19) to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= \Lambda(x)\ddot{x}_d + k_s\Lambda(x)(\dot{x}_d - \dot{x}) - k\Lambda(x)\dot{\tilde{x}} \\ &\quad +\Psi(x,\dot{\tilde{x}})\dot{\tilde{x}} + p(x) - F \end{aligned} \tag{3.31}$$

Substituting the force input given by (3.8) into (3.31), and using the definitions of w_e and η_p, we get:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -(k_s + k_{nd})\eta_p - (k + k_s)\Lambda(x)\dot{\tilde{x}} \\ &\quad -\Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{3.32}$$

Rewrite the term $\Psi(x,\dot{\tilde{x}})\eta_p$ on the right-hand side of (3.32) in terms of $\dot{\tilde{x}}$, and utilize (A.3) and (3.7) results in:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -\Psi(x,\dot{x})\eta_p - (k_s + k_{nd})\eta_p \\ &\quad -(k_s + k_{nd})\Lambda(x)\dot{\tilde{x}} + \Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{3.33}$$

Define another sub-Lyapunov function:

$$V_2 = \frac{1}{2}\eta_p^T\Lambda(x)\eta_p \tag{3.34}$$

Differentiating V_2 along (3.33), and utilizing (A.2) results in:

$$\begin{aligned} \dot{V}_2 &= -(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &\quad +\eta_p^T\Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{3.35}$$

From (3.35), and using (A.1) and (A.4), we obtain the following upper bound for $\dot{V}_2$:

$$\begin{aligned}\dot{V}_2 &\leq -(k_s + k_{nd})\|\eta_p\|^2 + (k + k_s)m_2\|\|\eta_p\|\dot{\tilde{x}}\| \\ &\quad +\zeta_c\|\eta_p\|^2\|\dot{\tilde{x}}\| \end{aligned} \tag{3.36}$$

3.3.4 Overall System Stability Analysis

From the results indicated in Sections 3.3.1, 3.3.2, and 3.3.3, we can obtain the following inequality:

$$\frac{1}{2}\lambda_1(\left\|\dot{\tilde{x}}\right\|^2 + \|e\|^2 + \left\|\eta_p\right\|^2) \leq V \leq \frac{1}{2}\lambda_2(\left\|\dot{\tilde{x}}\right\|^2 + \|e\|^2 + \left\|\eta_p\right\|^2) \tag{3.37}$$

Using the definitions of λ_1, λ_2 and $err(t)$ defined in (3.14) and (3.15), we place the following bounds on V:

$$\frac{1}{2}\lambda_1 \|err\|^2 \leq V \leq \frac{1}{2}\lambda_2 \|err\|^2 \tag{3.38}$$

Using the upper bounds of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$, and using (3.6), (3.9), and (3.15), we can form the upper bound on $\dot{V}$:

$$\begin{aligned}\dot{V} \leq\ & -k_s \|e\|^2 - k_s \left\|\eta_p\right\|^2 - k_s \left\|\dot{\tilde{x}}\right\|^2 \\ & + \left[\left\|\eta_p\right\| \left(\|e\| - 2k_n \left\|\eta_p\right\|\right)\right] + \left[\left\|\dot{\tilde{x}}\right\| \left(\|e\| - 2k_n \left\|\dot{\tilde{x}}\right\|\right)\right] \\ & + \left[(k + k_s)\, m_2 \left\|\eta_p\right\| \left(\left\|\dot{\tilde{x}}\right\| - (k + k_s)\, m_2 k_n \left\|\eta_p\right\|\right)\right] \qquad - km_1\|\dot{\tilde{x}}\|^2 \\ & + \left(\zeta_c\zeta_d \left\|\dot{\tilde{x}}\right\|^2 - \zeta_c\zeta_d \left\|\dot{\tilde{x}}\right\|^2\right) \\ & - (k_o - \|err\|) \left(\zeta_c \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c \left\|\eta_p\right\|^2\right)\end{aligned} \tag{3.39}$$

where we have used the fact derived from (3.15) that $\|err\| \geq \|e\|, \|\eta\|$, and $\|\dot{\tilde{x}}\|$.

By applying *Lemma 2* on the three bracketed terms in (3.39), a new upper bound on can be formed as:

$$\begin{aligned}\dot{V} \leq \; & -(k_s - \tfrac{1}{k_n}) \left\|e\right\|^2 - k_s \left\|\eta_p\right\|^2 - (k_s - \tfrac{1}{k_n}) \left\|\dot{\tilde{x}}\right\|^2 \\ & - (k_o - \left\|err\right\|) \left(\zeta_c \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c \left\|\eta_p\right\|^2 \right)\end{aligned} \tag{3.40}$$

From (3.40), it is easy to get:

$$\dot{V} \leq \; -\lambda_3 \left\|err\right\|^2 \quad \text{for} \;\; \left\|err(t)\right\| \leq k_0 \tag{3.41}$$

where λ_3 and $err(t)$ are defined in (3.14) and (3.15), respectively.

Finally, from (3.38), we can obtain:

$$\dot{V} \leq \; -\tfrac{2\lambda_3}{\lambda_2} V \quad \text{for} \;\; \sqrt{\frac{2V(t)}{\lambda_1}} < k_0 \tag{3.42}$$

Standard Lyapunov method [67] can now be applied to (3.38) and (3.42) to yield the result indicated by *Theorem 1.*

3.4 Estimation Error Formulation

To facilitate the estimation error formulation, we rewrite (3.3) as follows:

$$\dot{\hat{x}} = y + k\tilde{x} = y + k(x - \hat{x}) \tag{3.43}$$

It is easy to see that for some kinds of robots with simple configuration, the formulation of estimation error $\tilde{x}$ is straightforward, for example, a three-link planar robot, from $\dot{\hat{x}}$ we can get $\hat{x}$, which is the estimated end-effector position in X and Y direction, and the estimated end-effector orientation about Z axis; from the forward kinematics of a robot, we can get the actual end-effector position information, and the summation of the three joint angles gives us the actual orientation of the end-effector.

From the above information, we can easily formulate the position and orientation estimation error $\tilde{x}$ in closed form. But for other kinds of robots, like 6DOF PUMA-like robots, if use (3.43) directly, the formulation of orientation estimation error is not straightforward; from $\dot{\hat{x}}$ we can get $\hat{x}$, which is the estimated position in X, Y, and Z direction, and the estimated relative rotation changes about X, Y, and Z axis; from the robot homogeneous transformation matrix, we can get the actual end-effector position information in X, Y, and Z direction, and the actual end-effector orientation information, but expressed by a 3×3 rotation matrix. So the formulation of orientation estimation error in closed form is not trivial.

To formulate the position and orientation estimation error in closed form, we rewrite (3.43) as following:

$$J(q)\dot{\hat{q}} = y + k\left[x(q) - \hat{x}(\hat{q})\right] \tag{3.44}$$

where $J(q)$ is the basic Jacobian of the robot, $\hat{q}$ is the $n \times 1$ vector of the estimated joint positions, $\dot{\hat{q}}$ is the vector of the observed joint velocities.

We assume that a robot works in a non-singular region, so the inverse of $J(q)$ is always possible, and for a non-redundant robot, the mapping from $\dot{\hat{x}}$ to $\dot{\hat{q}}$ is unique. (3.44) is therefore rewritten as:

$$\dot{\hat{q}} = J^{-1}(q)\left\{y + k\left[x(q) - \hat{x}(\hat{q})\right]\right\} \tag{3.45}$$

From here we can get $\hat{q}$ through integrating (3.45) with respect to time, and this $\hat{q}$ will be used in (3.49).

The relationship between joint space velocity and operational space velocity can be expressed as:

$$\dot{x} = J\dot{q} \tag{3.46}$$

If we assume that the sampling time Δt is small, (3.46) can be written as:

$$\frac{\Delta x}{\Delta t} = J\frac{\Delta q}{\Delta t} \tag{3.47}$$

Both sides multiplied by Δt to obtain:

$$\Delta x = J\Delta q \tag{3.48}$$

From (3.48) and $\hat{q}$ obtained from the integration of (3.45), the closed form of the instantaneous estimation error $\tilde{x}$ can be formulated as:

$$\tilde{x} = J(q)(q - \hat{q}) \tag{3.49}$$

Using (3.49), we can rewrite (3.43) as:

$$\dot{\hat{x}} = y + k\tilde{x} = y + kJ(q)(q - \hat{q}) \tag{3.50}$$

3.5 Experimental Results

The experiments were performed using PUMA 560 robot. In the experiments, we bypassed the original motion controller card of PUMA 560 and used Servo To Go (www.servotogo.com) data acquisition card to control the robot. The real-time robotic control software runs under Windows NT 4.0 with RTX 4.3.2.1 (Venturcom Real-Time Extension). The PC used is a dual-processor 800MHz PC, and the sampling time is selected to be 1ms.

Our task is to move the end-effector in XYZ direction with the following desired position trajectory while maintaining the initial end-effector orientation constant all

the time.

$$\begin{aligned} p_{x_d} &= p_{x_0} + 50.0\sin(2\pi ft)\left(1 - e^{-0.05t^3}\right)\text{mm} \\ p_{y_d} &= p_{y_0} + 50.0\cos(2\pi ft)\left(1 - e^{-0.05t^3}\right)\text{mm} \\ p_{z_d} &= p_{z_0} + 50.0\cos(2\pi ft)\left(1 - e^{-0.05t^3}\right)\text{mm} \\ x_d &= \left[\begin{array}{ccc} p_{x_d} & p_{y_d} & p_{z_d} \end{array}\right]^T \end{aligned} \tag{3.51}$$

where $(p_{x_0}, p_{y_0}, p_{z_0})$ is the initial position of the robot. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

In our experiments, we use the combination of (3.45) and (3.4) to form our velocity observer, and use (3.49) to form the position and orientation estimation error.

3.5.1 Tracking Error Formulation

For tracking control in operational space, the tracking error e consists of position and orientation tracking errors:

$$e = \left[\begin{array}{cccccc} e_x & e_y & e_z & e_{\phi_x} & e_{\phi_y} & e_{\phi_z} \end{array}\right]^T \tag{3.52}$$

Here we define e_{pos} as the 3×1 vector of position tracking errors along X, Y, and Z axis, and e_{ori} as the 3×1 vector of orientation tracking errors about X, Y, and Z axis, respectively:

$$e_{pos} = \left[\begin{array}{ccc} e_x & e_y & e_z \end{array}\right]^T \tag{3.53}$$

$$e_{ori} = \left[\begin{array}{ccc} e_{\phi_x} & e_{\phi_y} & e_{\phi_z} \end{array}\right]^T \tag{3.54}$$

In section 3.2.2, the proposed controller needs to use $x_d, \dot{x}_d$ and $\ddot{x}_d$ to generate the required force commands. For position control, we can get the desired end-effector position, linear velocities and accelerations from (3.51). The instantaneous position tracking errors e_{pos} can be easily obtained by:

$$e_{pos} = \left[\begin{array}{ccc} p_{x_d} - p_x & p_{y_d} - p_y & p_{z_d} - p_z \end{array}\right]^T \tag{3.55}$$

where p_{xd}, p_{yd} and p_{zd} are the desired positions, and p_x, p_y and p_z are the positions of the robot.

The direction cosines representation of the initial end-effector orientation is given by:

$$\begin{array}{rcl} x_{rini} & = & [\quad n_{ini}^T \quad o_{ini}^T \quad a_{ini}^T \quad]^T \\ n_{ini} & = & [\quad n_{xini} \quad n_{yini} \quad n_{zini} \quad]^T \\ o_{ini} & = & [\quad o_{xini} \quad o_{yini} \quad o_{zini} \quad]^T \\ a_{ini} & = & [\quad a_{xini} \quad a_{yini} \quad a_{zini} \quad]^T \end{array} \tag{3.56}$$

The direction cosines representation of the end-effector orientation is given by:

$$\begin{array}{rcl} x_r & = & [\quad n^T \quad o^T \quad a^T \quad]^T \\ n & = & [\quad n_x \quad n_y \quad n_z \quad]^T \\ o & = & [\quad o_x \quad o_y \quad o_z \quad]^T \\ a & = & [\quad a_x \quad a_y \quad a_z \quad]^T \end{array} \tag{3.57}$$

For orientation control, since the task is to maintain the initial end-effector orientation during position tracking, the desired end-effector angular velocities and accelerations are just zeros. And the instantaneous orientation tracking errors e_{ori} (in the unit of radian) can be expressed as:

$$e_{ori} = \frac{1}{2}(\widehat{s}_1 n_{ini} + \widehat{s}_2 o_{ini} + \widehat{s}_3 a_{ini}) \tag{3.58}$$

where

$$\widehat{s}_1 = \tfrac{1}{2}\begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} \tag{3.59}$$

$$\widehat{s}_2 = \tfrac{1}{2}\begin{bmatrix} 0 & -o_z & o_y \\ o_z & 0 & -o_x \\ -o_y & o_x & 0 \end{bmatrix} \tag{3.60}$$

$$\widehat{s}_3 = \tfrac{1}{2}\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{3.61}$$

3.5.2 Experimental Results under Parametric Uncertainty

In the work involved in this dissertation, we use the PUMA 560 manipulator as a test bed. The dynamic model of PUMA 560 is obtained from [63]. This model does not include friction model. For our experiments, there are definite parameter uncertainties.

To test what are the smallest tracking errors that can be achieved by the proposed controller, we increased the controller gains until the robot started to vibrate, the gains used are listed in (3.62):

$$\begin{array}{rcl} k_{nd} & = & diag\{220, 220, 220, 66, 66, 66\} \\ k & = & diag\{200, 200, 200, 60, 60, 60\} \\ k_s & = & diag\{180, 180, 180, 54, 54, 54\} \end{array} \tag{3.62}$$

all these gains are in the unit of 1/s.

In order to make fair comparison, we also implemented the controller indicated in Section 3.2.2, but the observed velocities $\dot{\hat{x}}$ were replaced by that obtained from backwards difference plus a low pass filter with cutoff frequency of 100Hz. We refer to this as the backwards difference controller. During the experiments, we found that, due to large inherent noise ripple of filtered velocity, the controller gains values indicated in (3.62) were too high for the backwards difference controller, they caused the robot to vibrate. In order to reduce vibration, the possibly highest gains were selected as listed in (3.63):

$$\begin{array}{rcl} k_{nd} & = & diag\{20, 20, 20, 12, 12, 12\} \\ k_s & = & diag\{90, 90, 90, 30, 30, 30\} \end{array} \tag{3.63}$$

The experimental results are shown in Fig. 3.2 and Table 3.1. Where OC stands for the observer-controller, and BD for the backwards difference controller. The position tracking errors e_x, e_y, and e_z, orientation tracking errors e_{ϕ_x}, e_{ϕ_y}, and e_{ϕ_z} are defined in (3.52).

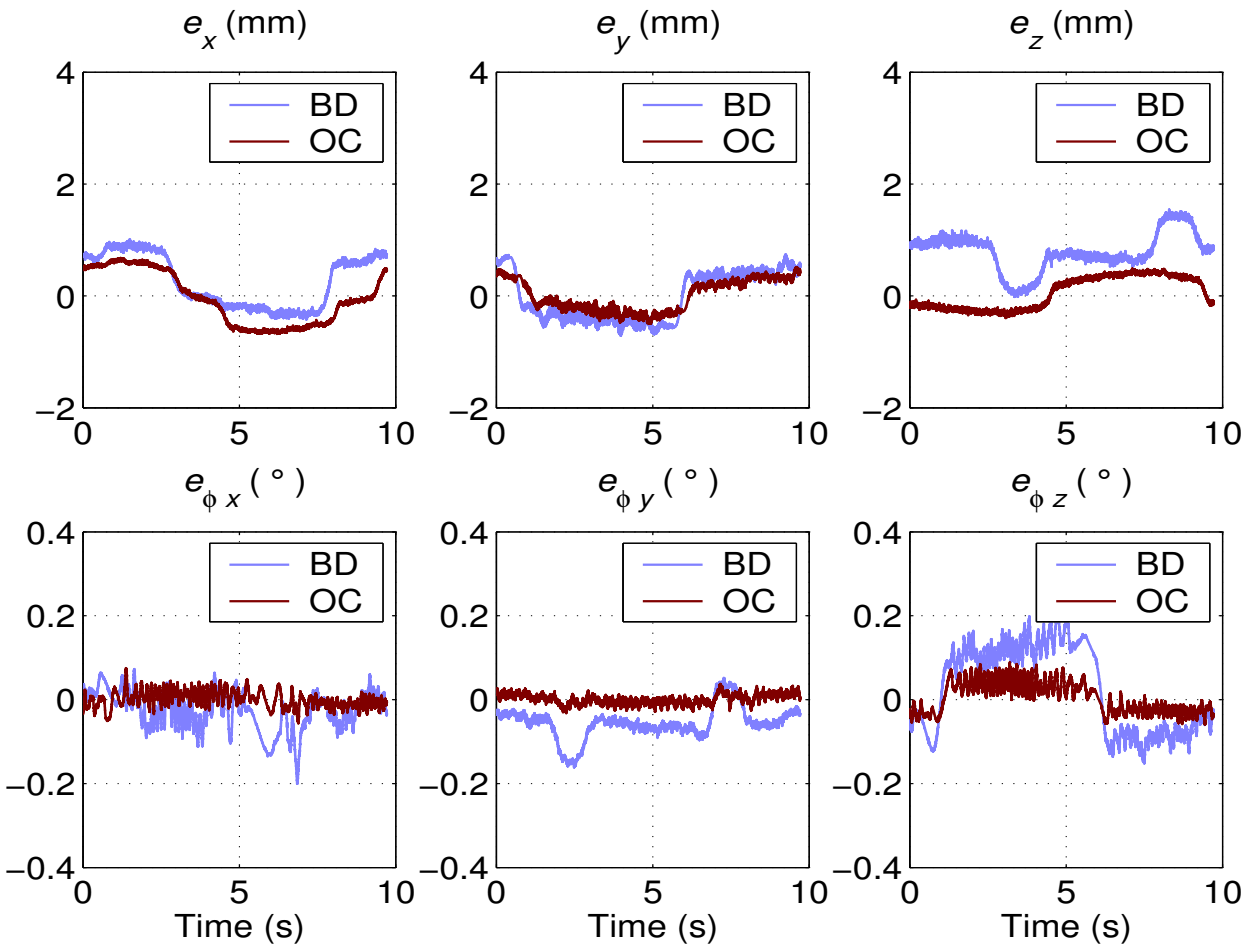

Figure 3.2: Observer controller - Tracking errors under parametric uncertainty

The results show that, using the proposed observer-controller, the maximum position and orientation tracking errors are about 1.4 to 3.5 times smaller than that of the backwards difference controller.

3.5.3 Experimental Results under Payload Variations

To examine the performance of the proposed observer-controller and the backwards difference controller under payload variations, a payload of 1.5kg, which is

Table 3.1: Observer controller - Maximum tracking errors under parametric uncertainty

	e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
OC	0.68mm	0.53mm	0.49mm	0.08^o	0.04^o	0.09^o
BD	1.02mm	0.75mm	1.55mm	0.20^o	0.14^o	0.22^o

Table 3.2: Observer controller - Maximum tracking errors under payload variations

	e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
OC	0.74mm	0.53mm	0.71mm	0.08^o	0.04^o	0.09^o
BD	1.96mm	1.40mm	3.12mm	0.30^o	0.38^o	0.33^o

about 60 percent of the maximum allowable payload of the robot, was attached to the end-effector of PUMA 560. Using the same gains as listed in (3.62) and (3.63) for the observer-controller and the backwards difference controller, respectively, the experimental results are shown in Fig. 3.3 and Table 3.2.

The results show that, using the observer-controller, the maximum position and orientation tracking errors are about 2.6 to 9.5 times smaller than that of the backwards difference controller.

Remark 1 *The experimental results indicate that, compared with the backwards difference controller, the observer-controller seems less sensitive to payload variations, which indicates that the proposed observer-controller maybe robust against parametric uncertainty and payload variations. The limitation of the backwards difference controller in compensation of parametric uncertainty and payload variations is because of its linear behavior (lack of dynamic information). While the observer-controller*

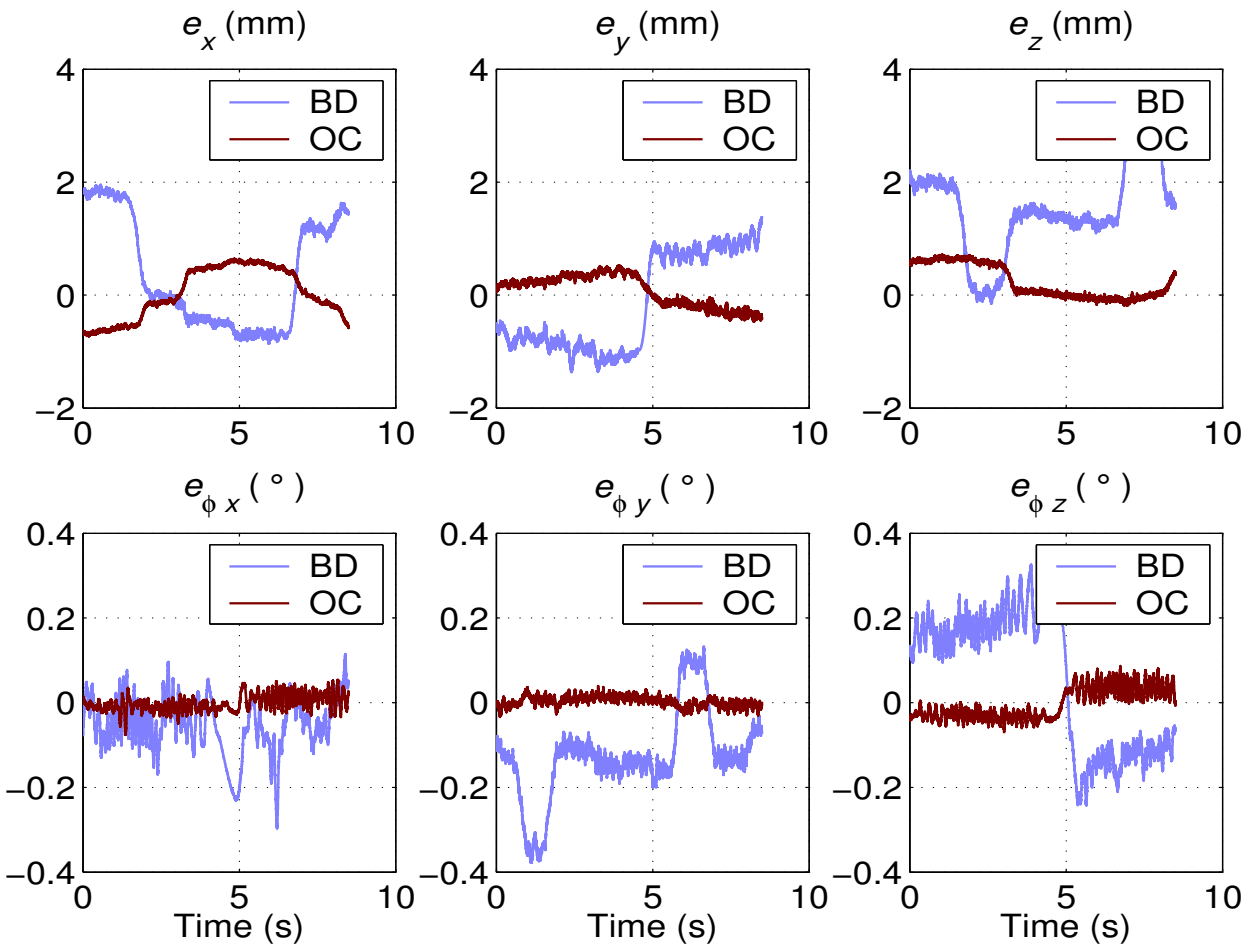

Figure 3.3: Observer controller - Tracking errors under payload variations

is designed to mimic the dynamic behavior of a robot, the position and orientation tracking performance can be better if the robot dynamic model is accurate enough.

Remark 2 *The proposed observer-controller seems a little bit complex, but it is easy to see from (3.3), (3.4) and (3.8) that this controller only requires the selection of three controllers gains k_{nd}, k, and k_s. From Theorem, 1 we know that in order to make the system stable, the controller gains k_s, k_n, and k_0 need to satisfy certain conditions. But in our experiments, even if the gains selection does not meet the above mentioned constrain, the robot still works well. We mention it here to indicate that, owing to the conservative nature of the Lyapunov stability analysis, the gains condition is treated as selection guideline rather than an absolute mandate.*

Remark 3 *The gain condition given by (3.12) implies that k_0 can be selected to cover any set of initial condition of the error vector $err(t)$. In addition, from (3.41) we can see that, the transient response of end-effector position tracking errors can be improved by increasing the controller gains k_s and k_{nd}.*

Remark 4 *As compared with the joint space observer-controller, the main advantage of the operational space observer-controller is to avoid solving the inverse kinematics problem of a robot manipulator.*

3.5.4 Quality of the Observed Velocities

To exam the velocity observation quality of the proposed velocity observer, we let the robot to follow the trajectory defined by (3.51), and at the same time, calculate the filtered joint velocities, i. e., to obtain the joint velocities using backwards difference approach used in conjunction with a lowpass filter. The backwards difference

algorithm can be expressed as:

$$\dot{\hat{q}}(kT) = \frac{q(kT) - q\left[(k-1)T\right]}{T} \tag{3.64}$$

where q represents the robot joint angles; $\dot{\hat{q}}$ represents the calculated joint velocities; k represents the sampling instant; and T represents the sampling time.

After obtaining $\dot{\hat{q}}(kT)$, we use a second order 100Hz Butterworth filter to reduce the noise in $\dot{\hat{q}}(kT)$. The reason of using 100Hz cutoff frequency is because lower cutoff frequency will introduce tracking delay further. Fig. 3.4 illustrates the delay caused by using a 10Hz Butterworth filter, where V_{xdes}, V_{ydes}, V_{zdes}, and ω_{xdes}, ω_{ydes}, ω_{zdes} are the desired end-effector linear and angular velocities, and V_x, V_y, V_z, and ω_x, ω_y, ω_z are the filtered end-effector linear and angular velocities, respectively.

Using the controllers gains given in (3.62), the joint velocities obtained from the proposed velocity observer is shown in Fig. 3.5. The joint velocities obtained from the filtering method is shown in Fig. 3.6.

It is clear that the joint velocities estimation obtained from the proposed velocity observer is smoother than that of the filtering approach, which is critical for high precision motion control.

To compare the quality of observed and filtered end-effector velocities in operational space, first we introduce a term called pseudo velocity tracking error, it is defined as:

$$\dot{e}_{pseu} = \dot{x}_d - \dot{\hat{x}} \tag{3.65}$$

where $\dot{\hat{x}}$ is either observed or filtered end-effector velocity.

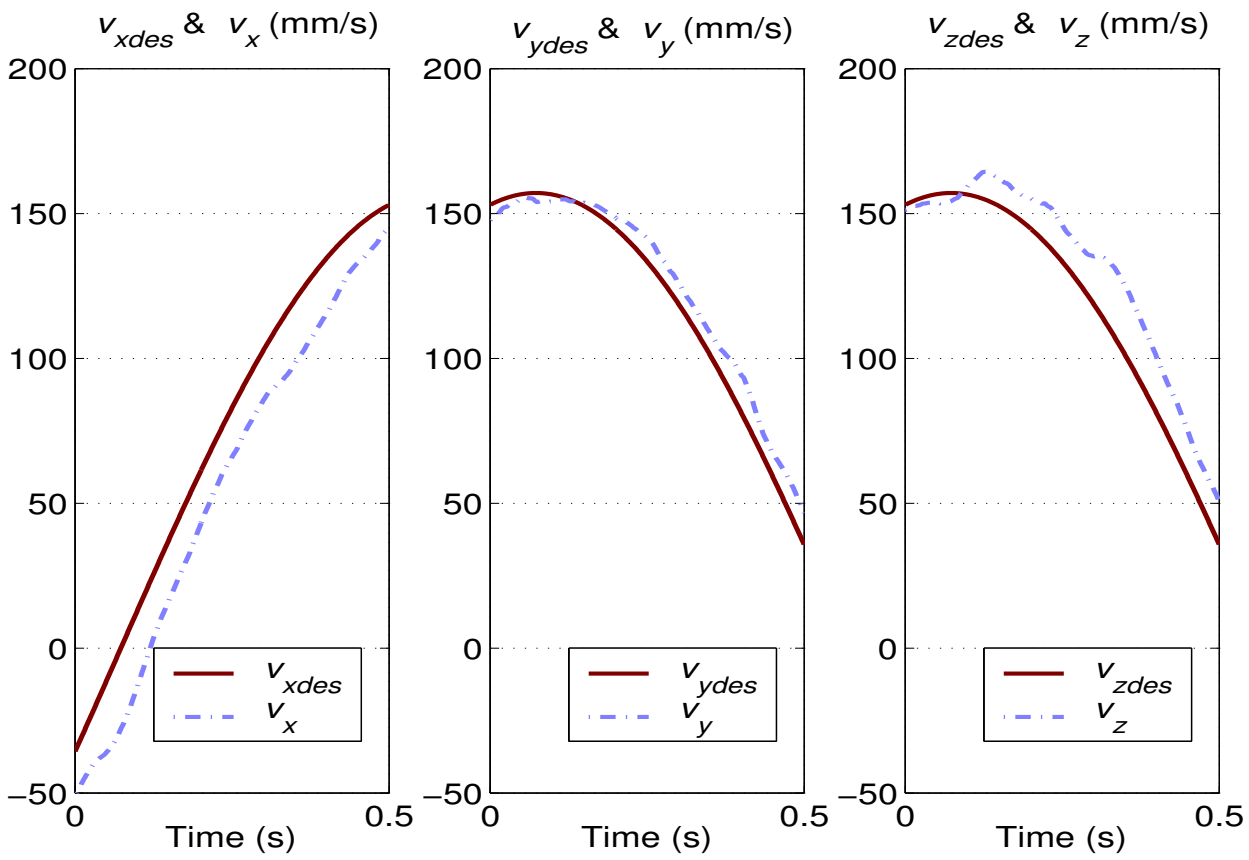

Figure 3.4: Observer controller - Tracking delay due to low cutoff frequency

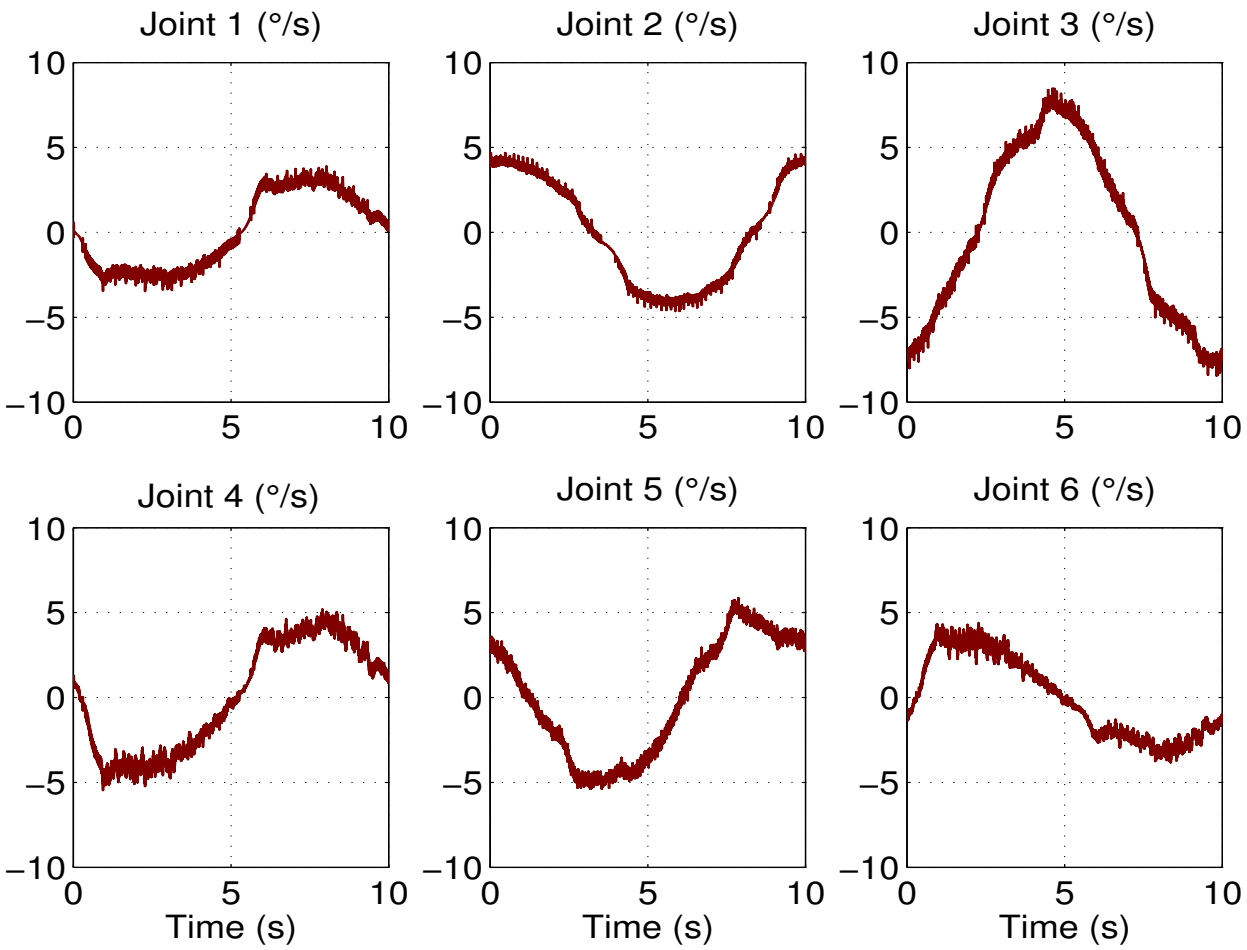

Figure 3.5: Observer controller - Joint velocities obtained from the velocity observer

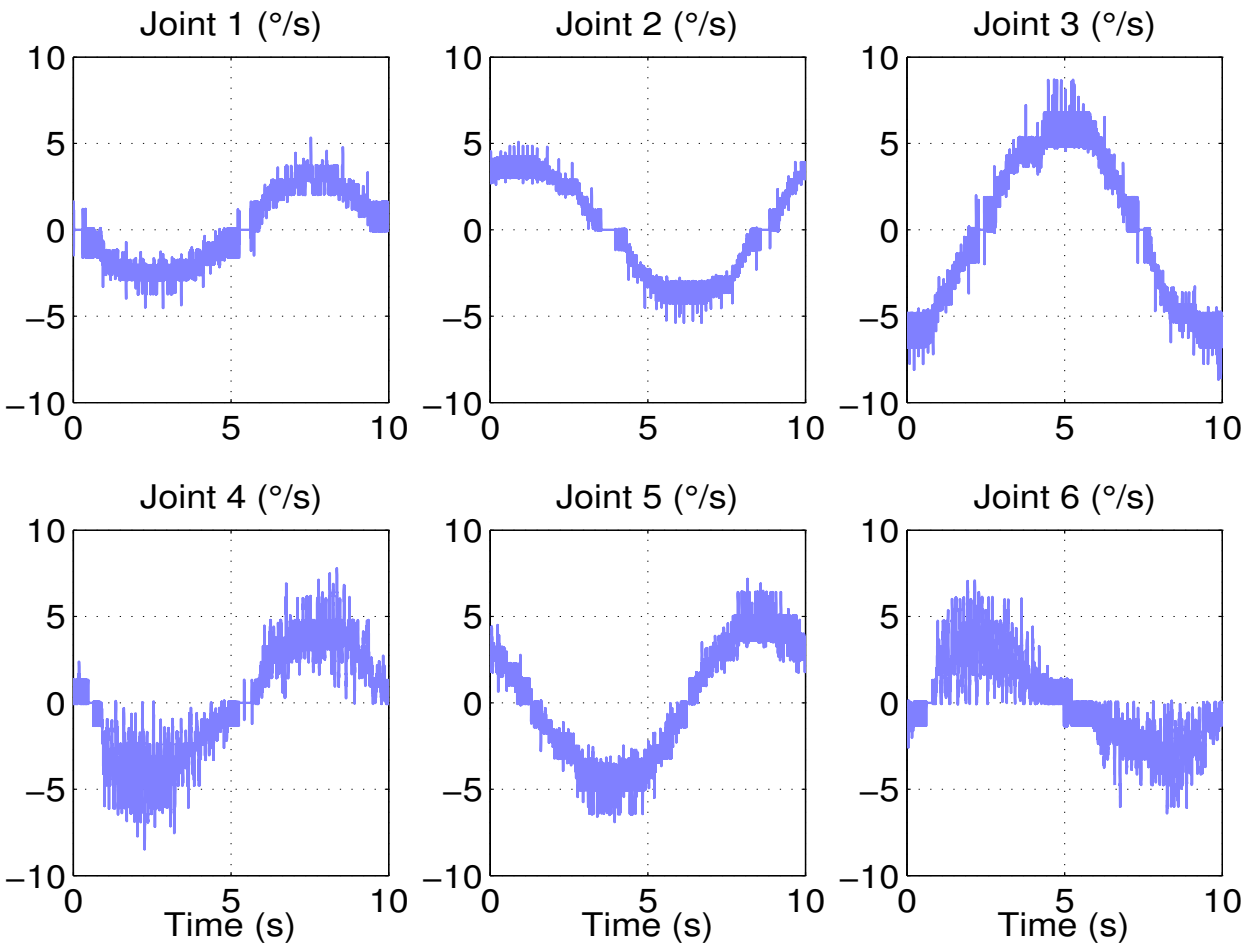

Figure 3.6: Observer controller - Joint velocities obtained from filtering method

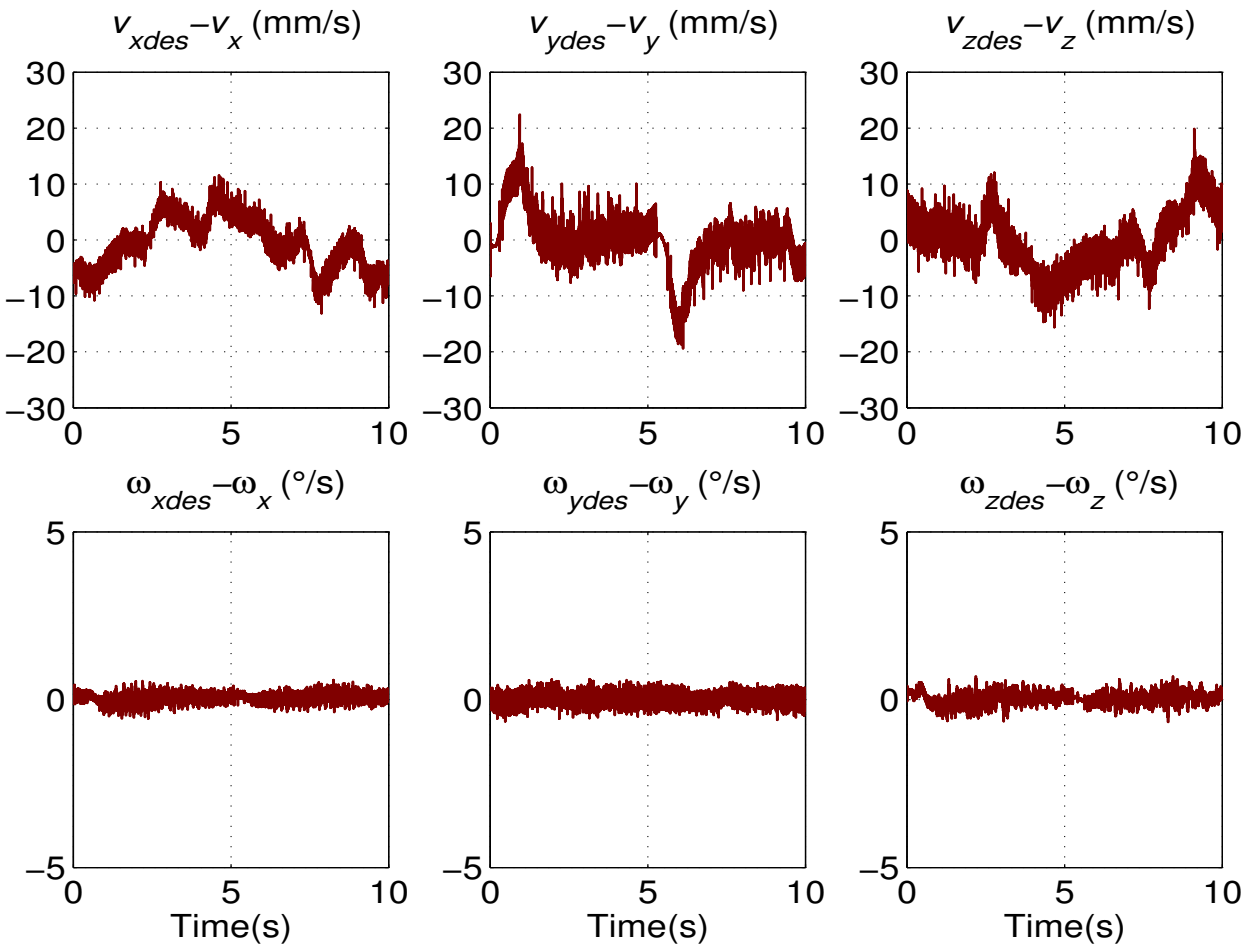

Figure 3.7: Observer controller - Pseudo velocity tracking errors using the velocity observer

The plot of the pseudo velocity tracking errors between the desired and the observed end-effector velocities is shown in Fig. 3.7, and that between the desired and the filtered velocities is shown in Fig. 3.8.

From the results we can see that the pseudo velocity tracking errors $\dot{e}_{pseu}$ obtained from the proposed velocity observer are much smoother than that from the filtered

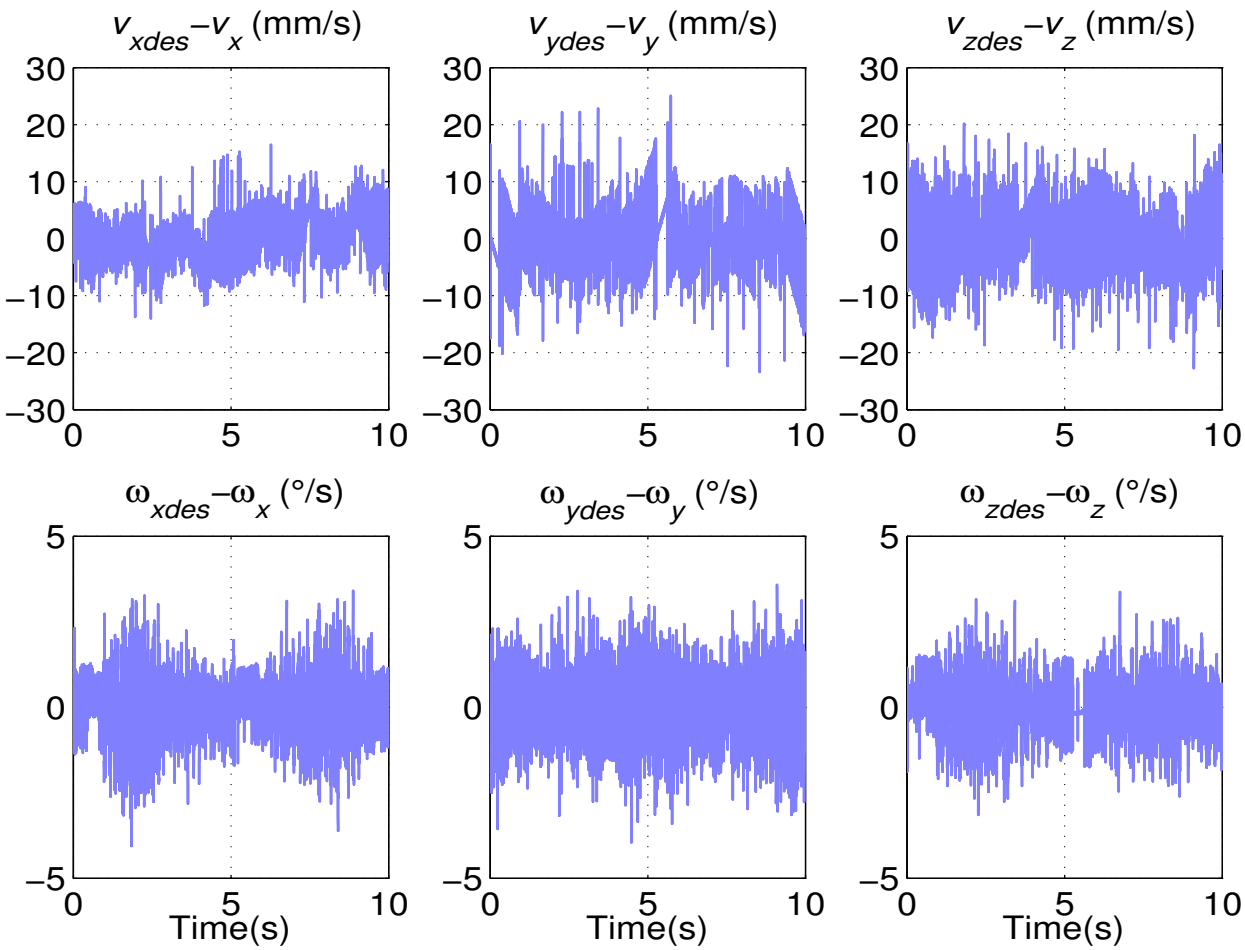

Figure 3.8: Observer controller - Pseudo velocity tracking errors using filtering method

joint velocities, hence higher gains can be used for the proposed observer-controller to achieve higher tracking accuracy.

Remark 5 *It is interesting to make comparison between the observed velocity and the filtered velocity here. There are two major differences between the two approaches. First, the observed velocity is computed based on the robot dynamic model plus feedback terms as illustrated by (3.3) and (3.4), while the filtered velocity is obtained purely from joint positions information as indicated by (3.64), there is no utilization of the robot dynamic information; second, the observed velocity is obtained from the integration of the computed end-effector acceleration $\dot{y}$ in (3.4), and the inherent feature of integration tends to suppress noise. While the filtered velocity is obtained through differentiation as shown by (3.3), and the inherent feature of differentiation tends to amplify noise. Hence, due to the two major differences, our proposed velocity observer is able to obtain more accurate and less noisy velocity information. Based on the better-quality observed velocity, we are able to choose higher gains for our observer-controller to achieve more accurate trajectory tracking, even under parametric uncertainties and payload variations, which is verified by our experimental results.*

Remark 6 *From the formulation of the proposed velocity observer we know that the estimated velocity information is obtained directly from the robot dynamic model, no filtering process, hence there is no delay in observed velocity information. However, in order to remove part of the noise component in the velocity information computed by backwards difference algorithm, a low-pass filter is needed, which introduces tracking delay as illustrated by Fig. 3.4. This is another advantage of velocity observer.*

Remark 7 *The observer-controller is developed based on perfect knowledge of robot dynamics. However, experimental results show that, in the case of modelling errors, the whole system is still a stable system. Theoretical analysis indicates that, if a robot dynamic model is exactly known, tracking errors will approach zero as time approached infinity. Due to parametric uncertainties and payload variations, tracking errors will converge to certain values, but won't approach zero, as shown in Figs. 3.2 and 3.3.*

Remark 8 *In our experiments, fourth order Runge-Kutta algorithm was used to solve the nonlinear differential equation, thereby allowing us to estimate the joint velocity and position.*

3.6 Conclusions

In this chapter, we proposed an operational space observer-controller to achieve velocity observation and position tracking. This operational space controller is more practical as compared with the controller used in joint space. Under parametric uncertainties and payload variations, the proposed observer-controller is able to achieve higher tracking accuracy than the controller using filtered velocity. The experimental results done on PUMA 560 verify the effectiveness of the proposed nonlinear model-based controller. Compared with the estimated velocities obtained from backwards difference algorithm plus a lowpass filter, the observed velocities using the proposed velocity observer are less noisy, hence high gains can be used to achieve higher tracking accuracy. The result of the controller has been published in [68, 69].

The observer-based controllers introduced in this thesis is semi-global stable, its gains selection have some conditions. Under large degree of parametric uncertainties, payload variation, and contact force, the controller gains need to be large enough in

order to make the system stable. However, a robot may vibrate if the gains become too high, hence, it may not be suitable to use the controller under large degree of uncertainties.

CHAPTER 4

CONTROL ALGORITHM 2: ROBUST OBSERVER-CONTROLLER FORMULATION

4.1 Introduction

The control objective is to further improve the performance of the observer-controller presented in Chapter 3 so that it is more robust under parametric uncertainties and payload variations.

4.1.1 Formulation of Robust Velocity Observer

The velocity observer discussed in Chapter 3 is modified by adding a term $k_i\tilde{x}$ in Eq (3.4) to ensure robustness in terms of parametric uncertainties and payload variations. The following equations describe the robust velocity observer where equations in Chapter 3 are repeated and assigned new equation numbers for expository convenience.

$$\dot{\hat{x}} = y + k\tilde{x}, y(0) = -k\tilde{x}(0) \tag{4.1}$$

$$\dot{y} = \Lambda^{-1}\Big[F - \Psi(x, \dot{\hat{x}})\dot{\hat{x}} - p(x) + k_i\tilde{x}\Big] \tag{4.2}$$

where

$$\tilde{x} = x - \hat{x} \tag{4.3}$$

where $\dot{\hat{x}}$, y, k, and F have been defined in Chapter 3.

k_i is a positive scalar constant. Compared with Eq (3.4), the purpose of the term $k_i\tilde{x}$ in Eq (4.2) is: when $\tilde{x}$ is greater than zero, it indicates that the estimated robot end-effector position $\hat{x}$ is less than the actual position, the term $k_i\tilde{x}$ becomes positive and $\dot{y}$ will increase. Consequently, $\dot{\hat{x}}$ will increase, and $\tilde{x}$, the error between the estimated robot end-effector position $\hat{x}$ and the actual position x will be smaller.

4.1.2 Formulation of Robust Observer-Based Controller

Based on the controller introduced in Chapter 3, an additional term $-k_i\tilde{x}$ is added in (3.8) to formulate the the following new controller:

$$F = (k_s + k_{nd})\eta_p + w_e - k_i\tilde{x} \tag{4.4}$$

where k_{nd} is defined by (3.9).

Compared with Eq (3.8), the purpose of the term $-k_i\tilde{x}$ in Eq 4.4 is: when $\tilde{x}$ is greater than zero, it indicates that the estimated robot end-effector position $\hat{x}$ is less than the actual position, the term $-k_i\tilde{x}$ becomes negative and the driving force F will decrease. Consequently, the robot will move slowly, and the error between the estimated robot end-effector position $\hat{x}$ and the actual position x will be smaller.

The force command F will be used in the observer indicated by (4.2). And the torque commands for driving the robot can be obtained by:

$$\Gamma = J^T F$$

4.2 Overall System Stability Result and Analysis

Theorem 2 *Under the assumption that the exact model of a robot is known, if the observer-controller gains satisfy the following sufficient conditions :*

$$\begin{array}{rcl} k_s & > & 1/k_n \\ k_0 & > & \|err(0)\| \end{array} \tag{4.5}$$

the closed-loop tracking error system is stable and the errors $e(t)$, $\dot{e}(t)$, $\tilde{x}(t)$, and $\dot{\tilde{x}}(t)$ are bounded where

$$err = \begin{bmatrix} \eta_p^T & e^T & \dot{\tilde{x}}^T & \tilde{x}^T \end{bmatrix}^T \in \Re^{4n} \tag{4.6}$$

We will now present the proof using Lyapunov stability analysis. To determine the stability of the overall closed-loop control system, we use the following Lyapunov function.

$$V = V_0 + V_1 + V_2 \tag{4.7}$$

where the three sub-Lyapunov function V_0, V_1, and V_2 are defined as:

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T k_i \tilde{x} \tag{4.8}$$

$$V_1 = \frac{1}{2}e^T e \tag{4.9}$$

$$V_2 = \frac{1}{2}\eta_p^T \Lambda(x)\eta_p \tag{4.10}$$

$\dot{V}$ can be obtained by:

$$\dot{V} = \dot{V}_0 + \dot{V}_1 + \dot{V}_2 \tag{4.11}$$

4.2.1 Lyapunov Function for Observation Error $\tilde{x}$ and $\dot{\tilde{x}}$

To form the bound of $\dot{V}_0$, first, take the time derivative of (4.1) and then substitute (4.2) into the resulting expression to yield:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} + p(x) - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} = F \tag{4.12}$$

where the velocity observation error $\dot{\tilde{x}}$ is obtained by differentiating (4.3) with respect to time:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \tag{4.13}$$

Subtract (4.12) from (2.2), use (A.3) and (3.7) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x,\dot{x})\dot{\tilde{x}} + \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} + k_i\tilde{x} = 0 \tag{4.14}$$

Differentiate V_0 along (4.14) to get:

$$\begin{aligned} \dot{V}_0 &= \dot{\tilde{x}}^T\left[-\Psi(x,\dot{x})\dot{\tilde{x}} - \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x}\right] \\ &\quad + \frac{1}{2}\dot{\tilde{x}}^T\dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i\tilde{x} \end{aligned} \tag{4.15}$$

Utilize (A.2) to yield:

$$\dot{V}_0 = -\dot{\tilde{x}}\left[\Psi(x,\dot{\hat{x}}) + k\Lambda(x)\right]\dot{\tilde{x}} \tag{4.16}$$

Then utilize (A.1) and (A.4) to get the upper bound of $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\|\dot{\hat{x}}\| - km_1\right)\|\dot{\tilde{x}}\|^2 \tag{4.17}$$

Substitute for $\dot{\hat{x}}$ from (3.10) into (4.17), and utilize (3.2) to get the new upper bound for $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\zeta_d + \zeta_c\|\eta_p\| + \zeta_c k_s\|e\| - km_1\right)\|\dot{\tilde{x}}\|^2 \tag{4.18}$$

4.2.2 Lyapunov Function for Tracking Error e

From the form of (4.18), we are motivated to design a controller which ensures that the $\|e\|$ and $\|\eta_p\|$ terms in (4.18) are both driven to zero; hence, we are motivated to develop tracking error systems and the corresponding sub-Lyapunov functions to facilitate the goal.

The position tracking error system can be formed by differentiating (3.1) with respect to time to yield:

$$\dot{e} = \dot{x}_d - \dot{x}$$

Since $\dot{x}$ is not measurable, we use the estimated term $\dot{\hat{x}}$ to eliminate $\dot{x}$ and get the following equation:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} - \dot{\tilde{x}} \tag{4.19}$$

Add and subtract a fictitious controller [66] to the right-hand side of (4.19) to yield:

$$\dot{e} = \dot{x}_d - [\dot{x}_d + k_s e] + [\dot{x}_d + k_s e] - \dot{\hat{x}} - \dot{\tilde{x}} \tag{4.20}$$

where k_s is a positive controller gain.

Simplify (4.20) by utilizing (3.10) to get:

$$\dot{e} = -k_s e + \eta_p - \dot{\tilde{x}} \tag{4.21}$$

The upper bound for the time derivative of V_1 along (4.21) is given by:

$$\dot{V}_1 \leq -k_s \|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{4.22}$$

4.2.3 Lyapunov Function for η_p

From the form of (4.22) and the fact that the form of (4.18) indicates that $\|\dot{\tilde{x}}\|$ can be driven to zero, we are motivated to design a force input controller which ensures that η_p can be driven to zero.

The tracking error system for η_p can be formed by differentiating (3.10) with respect to time, multiplying both sides of the resulting expression by $\Lambda(x)$, and substituting the right-hand side of (4.12) for $\ddot{\tilde{x}}$ to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= \Lambda(x)\ddot{x}_d + k_s\Lambda(x)(\dot{x}_d - \dot{x}) - k\Lambda(x)\dot{\tilde{x}} \\ &\quad -k_i\tilde{x} + \Psi(x,\dot{\hat{x}})\dot{\hat{x}} + p(x) - F \end{aligned} \tag{4.23}$$

Substitute the force input given by (4.4) into (4.23), use the definitions of w_e and η_p to get:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -(k_s + k_{nd})\eta_p - (k + k_s)\Lambda(x)\dot{\tilde{x}} \\ &\quad -\Psi(x,\dot{\hat{x}})\eta_p \end{aligned} \tag{4.24}$$

Rewrite the term $\Psi(x,\dot{\hat{x}})\eta_p$ on the right-hand side of (4.24) in terms of $\dot{\tilde{x}}$, and utilize (A.3) and (3.7) to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -\Psi(x,\dot{x})\eta_p - (k_s + k_{nd})\eta_p \\ &\quad -(k_s + k_{nd})\Lambda(x)\dot{\tilde{x}} + \Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{4.25}$$

In section 4.2, V_2 was defined as:

$$V_2 = \frac{1}{2}\eta_p^T\Lambda(x)\eta_p$$

Differentiate V_2 along (4.25), and utilize (A.2) to get:

$$\begin{aligned}\dot{V}_2 &= -(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &\quad +\eta_p^T\Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{4.26}$$

From (4.26), utilize (A.1) and (A.4), we can obtain the following upper bound for $\dot{V}_2$:

$$\begin{aligned}\dot{V}_2 &\leq -(k_s + k_{nd})\|\eta_p\|^2 + (k + k_s)m_2\|\|\eta_p\|\dot{\tilde{x}}\| \\ &\quad +\zeta_c\|\eta_p\|^2\|\dot{\tilde{x}}\| \end{aligned} \tag{4.27}$$

4.2.4 Overall System Stability Analysis

Use the upper bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$, and utilize (3.6), (3.9), and (4.6), we can form the upper bound on $\dot{V}$:

$$\begin{aligned}\dot{V} \leq\ & -k_s\|e\|^2 - k_s\left\|\eta_p\right\|^2 - k_s\left\|\dot{\tilde{x}}\right\|^2 \\ & + \left[\left\|\eta_p\right\|\left(\|e\| - 2k_n\left\|\eta_p\right\|\right)\right] \\ & + \left[\left\|\dot{\tilde{x}}\right\|\left(\|e\| - 2k_n\left\|\dot{\tilde{x}}\right\|\right)\right] \\ & + \left[(k + k_s)\, m_2\left\|\eta_p\right\|\left(\left\|\dot{\tilde{x}}\right\| - (k + k_s)\, m_2 k_n\left\|\eta_p\right\|\right)\right] \\ & + \left(\zeta_c\zeta_d\left\|\dot{\tilde{x}}\right\|^2 - \zeta_c\zeta_d\left\|\dot{\tilde{x}}\right\|^2\right) \\ & - (k_o - \|err\|)\left(\zeta_c\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c\left\|\eta_p\right\|^2\right)\end{aligned} \tag{4.28}$$

where we have used the fact derived from (4.6) that $\|err\| \geq \|e\|, \|\eta\|$, and $\|\dot{\tilde{x}}\|$.

By applying the nonlinear damping tool [66] on the three bracketed terms in (4.28), a new upper bound on $\dot{V}$ can be formed as:

$$\begin{aligned}\dot{V} \leq\ & -(k_s - 1/k_n)\|e\|^2 - k_s\left\|\eta_p\right\|^2 - (k_s - 1/k_n)\left\|\dot{\tilde{x}}\right\|^2 \\ & - (k_o - \|err\|)\left(\zeta_c\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c\left\|\eta_p\right\|^2\right)\end{aligned} \tag{4.29}$$

From (4.29), it is easy to get:

$$\begin{aligned} \dot{V} \leq \quad & -(k_s - 1/k_n)(\|e\|^2 + \left\|\eta_p\right\|^2 + \left\|\dot{\tilde{x}}\right\|^2) \\ & \text{for} \quad \|err\| \leq k_0 \end{aligned} \tag{4.30}$$

where err was defined in (4.6).

From (4.30) we can see that, if $k_s > 1/k_n$ and $k_0 \geq ||err(0)||$, we can get:

$$\dot{V} \leq 0 \tag{4.31}$$

From (4.31), we can get the conclusion that, the position tracking errors e, position estimation errors $\tilde{x}$, velocity estimation errors $\dot{\tilde{x}}$, and the observed filtered tracking error signal η_p of the observer-controller are bounded. Furthermore, the end-effector velocity tracking error is also bounded. In fact, after adding and subtracting $\dot{x}$ to the right-hand side of (3.10) and rearranging the terms, we can formulate the following inequality:

$$\|\dot{e}\| = \|\dot{x}_d - \dot{x}\| \leq \left\|\eta_p\right\| + k_s \|e\| + \|\dot{x}\| \tag{4.32}$$

Since each of the terms on the right-hand side of the above equation is bounded, $\|\dot{e}\|$ is also bounded. This yields the result indicated by *Theorem 2*.

4.3 Experimental Results

The experiments were performed using PUMA 560 robot.

Our task is to move the end-effector in XYZ direction with the following desired position trajectory while maintaining the initial end-effector orientation all the time.

$$\begin{aligned} p_{x_d} &= p_{x_0} + 50.0 \sin(0.2\pi t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{y_d} &= p_{y_0} + 50.0 \cos(0.2\pi t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{z_d} &= p_{z_0} + 50.0 \cos(0.2\pi t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\ x_d &= \begin{bmatrix} p_{x_d} & p_{y_d} & p_{z_d} \end{bmatrix}^T \end{aligned} \tag{4.33}$$

where $(p_{x_0}, p_{y_0}, p_{z_0})$ is the initial position vector of the robot. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

In our experiments, we use the combination of (3.45) and (4.2) to form our velocity observer, and use (3.49) to form the position and orientation estimation error.

4.3.1 Experimental Results under Parametric Uncertainty

For our experiments, in order to make comparison with the controller proposed in Chapter 3, the same gain values for k_{nd}, k, and k_s were used, k_i was selected as:

$$k_i = \text{diag}\{2000, 2000, 2000, 3000, 3000, 3000\} \tag{4.34}$$

all these gains are in the unit of 1/s.

In order to make fair comparison, we also implemented the controller indicated in Section 4.1.2, but the observed velocities $\dot{\hat{x}}$ were replaced by that obtained from backwards difference plus a low pass filter with cutoff frequency of 100Hz. We called it the backwards difference controller. During the experiments, we found that, due to large inherent noise ripple of filtered velocity, the controller gains values indicated in (4.34) were too high for the backwards difference controller, they caused the robot to vibrate. In order to reduce vibration, the possibly highest gains were selected as listed in (4.35):

$$\begin{array}{rcl} k_{nd} & = & diag\{20, 20, 20, 12, 12, 12\} \\ k_s & = & diag\{90, 90, 90, 30, 30, 30\} \end{array} \tag{4.35}$$

The experimental results are shown in Fig. 4.1 and Table 4.1. Where ROC stands for the observer-controller, and BD for the backwards difference controller. The position tracking errors e_x, e_y, and e_z, orientation tracking errors e_{ϕ_x}, e_{ϕ_y}, and e_{ϕ_z} are defined in (3.52).

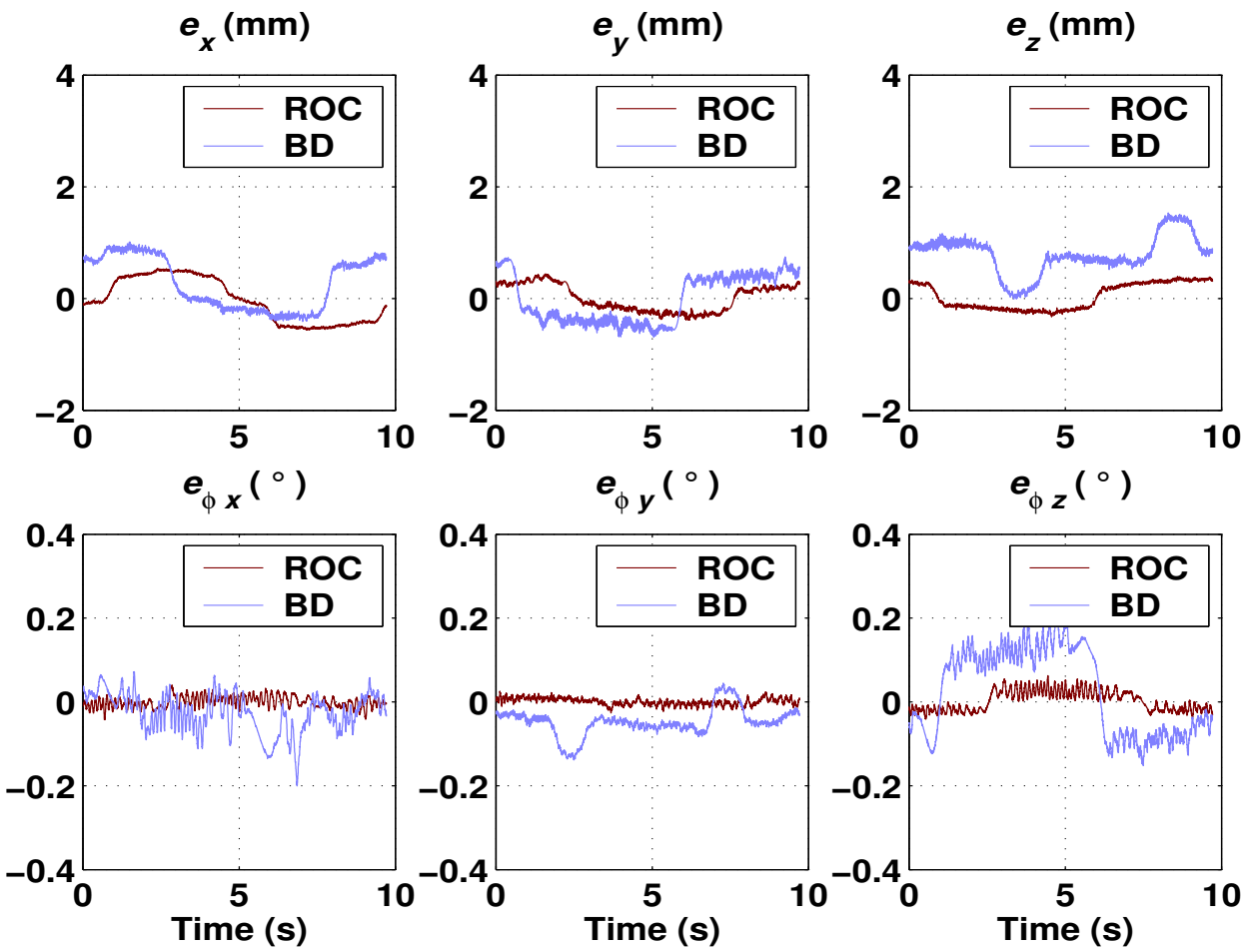

Figure 4.1: Robust observer controller - Tracking errors under parametric uncertainty

The results show that, using the proposed observer-controller, the maximum position and orientation tracking errors are about 1.7 to 4.7 times smaller than that of the backwards difference controller.

4.3.2 Experimental Results under Payload Variations

To examine the performance of the proposed observer-controller and the backwards difference controller under payload variations, a payload of 1.5kg, which is

Table 4.1: Robust observer controller - Maximum tracking errors under parametric uncertainty

	e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
ROC	0.57mm	0.44mm	0.40mm	0.04^o	0.03^o	0.06^o
BD	1.02mm	0.75mm	1.55mm	0.20^o	0.14^o	0.22^o

Table 4.2: Robust observer controller - Maximum tracking errors under parametric uncertainty

	e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
ROC	0.70mm	0.48mm	0.67mm	0.03^o	0.03^o	0.07^o
BD	1.96mm	1.58mm	3.12mm	0.30^o	0.38^o	0.33^o

about 60 percent of the maximum allowable payload of the robot, was attached to the end-effector of PUMA 560. Using the same gains as listed in (4.34) and (4.35) for the observer-controller and the backwards difference controller, respectively, the experimental results are shown in Fig. 4.2 and Table 4.2.

The results show that, using the observer-controller, the maximum position and orientation tracking errors are about 2.8 to 12.7 times smaller than that of the backwards difference controller.

Compared with the observer-controller introduced in Chapter 3, the controller presented in this Chapter can achieve higher tracking accuracy under parametric uncertainties and payload variations. This is due to the introduction of the two terms $k_i\tilde{x}$, and $-k_i\tilde{x}$ in (4.2), and (4.4), respectively, as explained in Section 4.1.1.

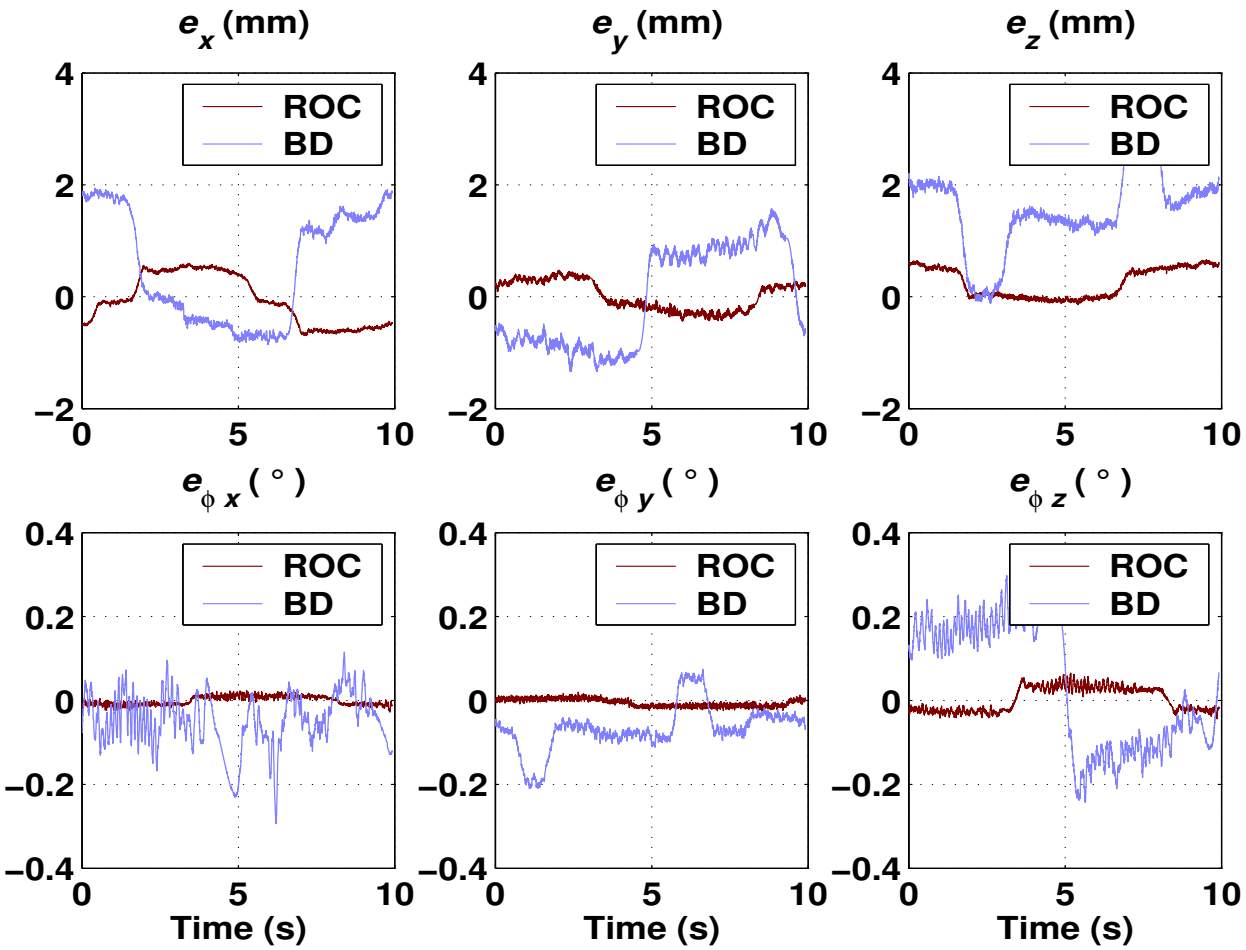

Figure 4.2: Robust observer controller - Tracking errors under payload variations

4.3.3 Quality of the Observed Velocities

To exam the velocity observation quality of the proposed velocity observer, we let the robot to follow the trajectory defined by (4.33), and at the same time, calculate the filtered joint velocities. Using the controllers gains given in (4.34), and select the desired trajectory as indicated by (4.33), under different frequencies of the desired position trajectories ($f = 0.1, 0.5$, and $1.0Hz$), the velocities in operational space obtained from the proposed velocity observer and the filtering method are shown in Fig. 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8, respectively.

It is clear that the velocities estimation obtained from the proposed velocity observer is smoother than that of the filtering approach, which is critical for high precision motion control.

Remark 9 *Based on the observer-controller presented in Chapter 3, we introduce a gain k_i in the observer-controller presented in this chapter. The gain k_i is used in both the velocity observer and the force controller to make correction to the observed velocity and the force command based on the position estimation error. As explained in Section 4.1.1, because of this gain, under the assumption that the dynamic model of a robot is exactly known, the position estimation error $\tilde{\boldsymbol{x}}(t)$ can also be driven to zero. Under system uncertainties, by adjusting k_i, $\tilde{\boldsymbol{x}}(t)$ can be confined within a narrow boundary so that the variation of the observed velocity can be much smaller, hence the velocity observer becomes more robust.*

Remark 10 *In the observer-controller presented in Chapter 3, we used fourth order Runge-Kutta algorithm to obtain robot velocity and position. First order Euler algorithm is less accurate than fourth order Runge-Kutta method. However, in order to*

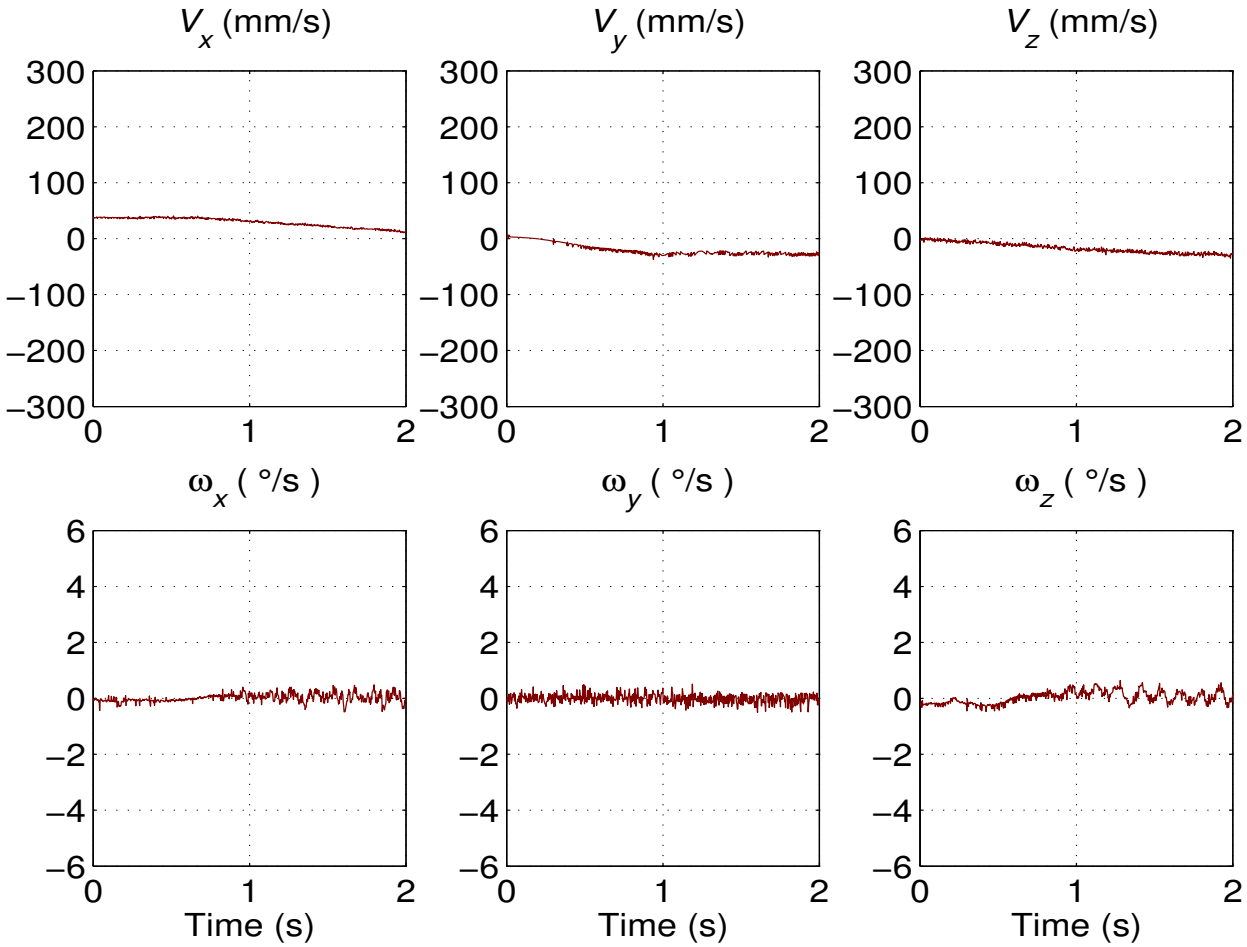

Figure 4.3: Robust observer controller - Velocities obtained from the velocity observer (f=0.1Hz)

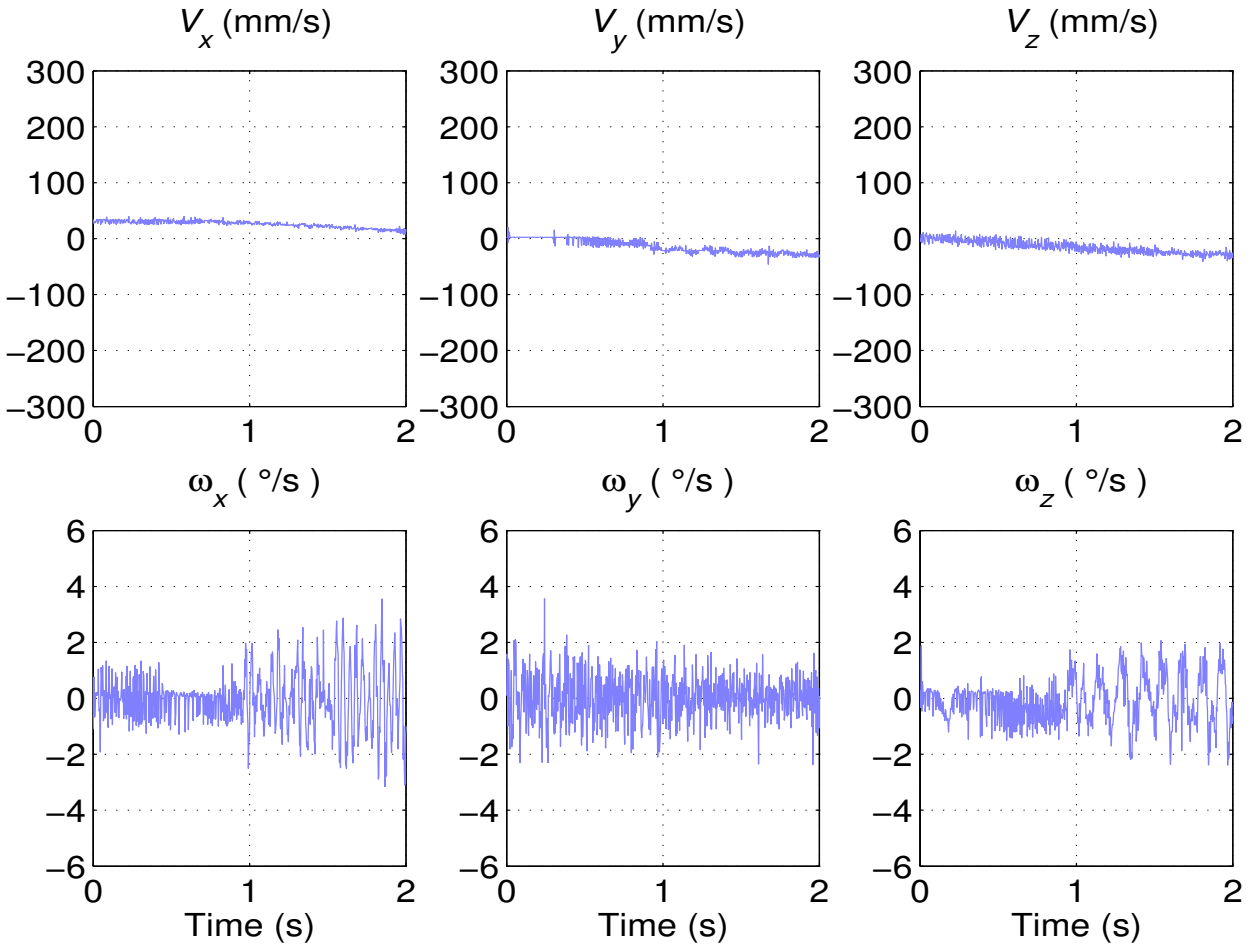

Figure 4.4: Robust observer controller - Velocities obtained from filtering method (f=0.1Hz)

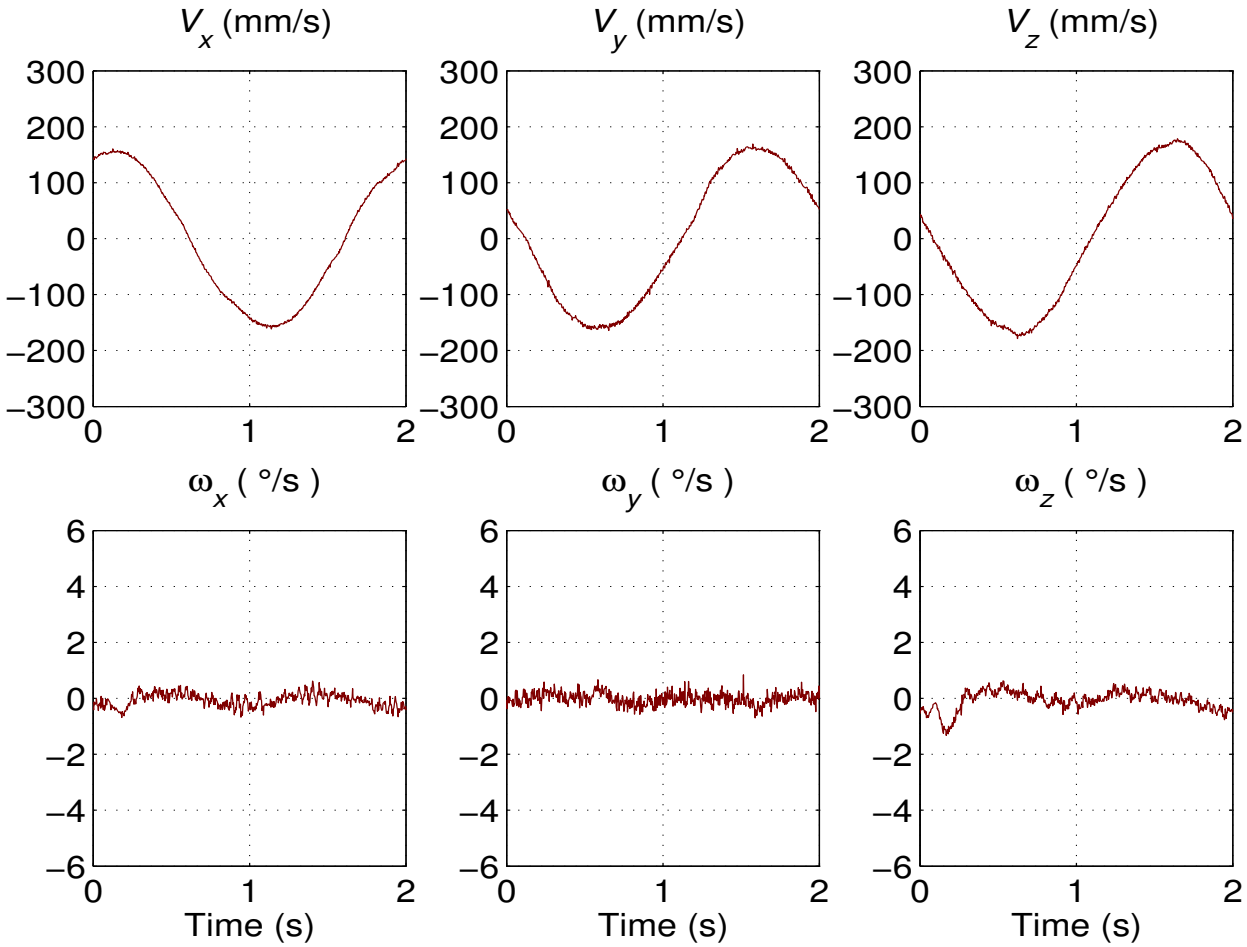

Figure 4.5: Robust observer controller - Velocities obtained from the velocity observer (f=0.5Hz)

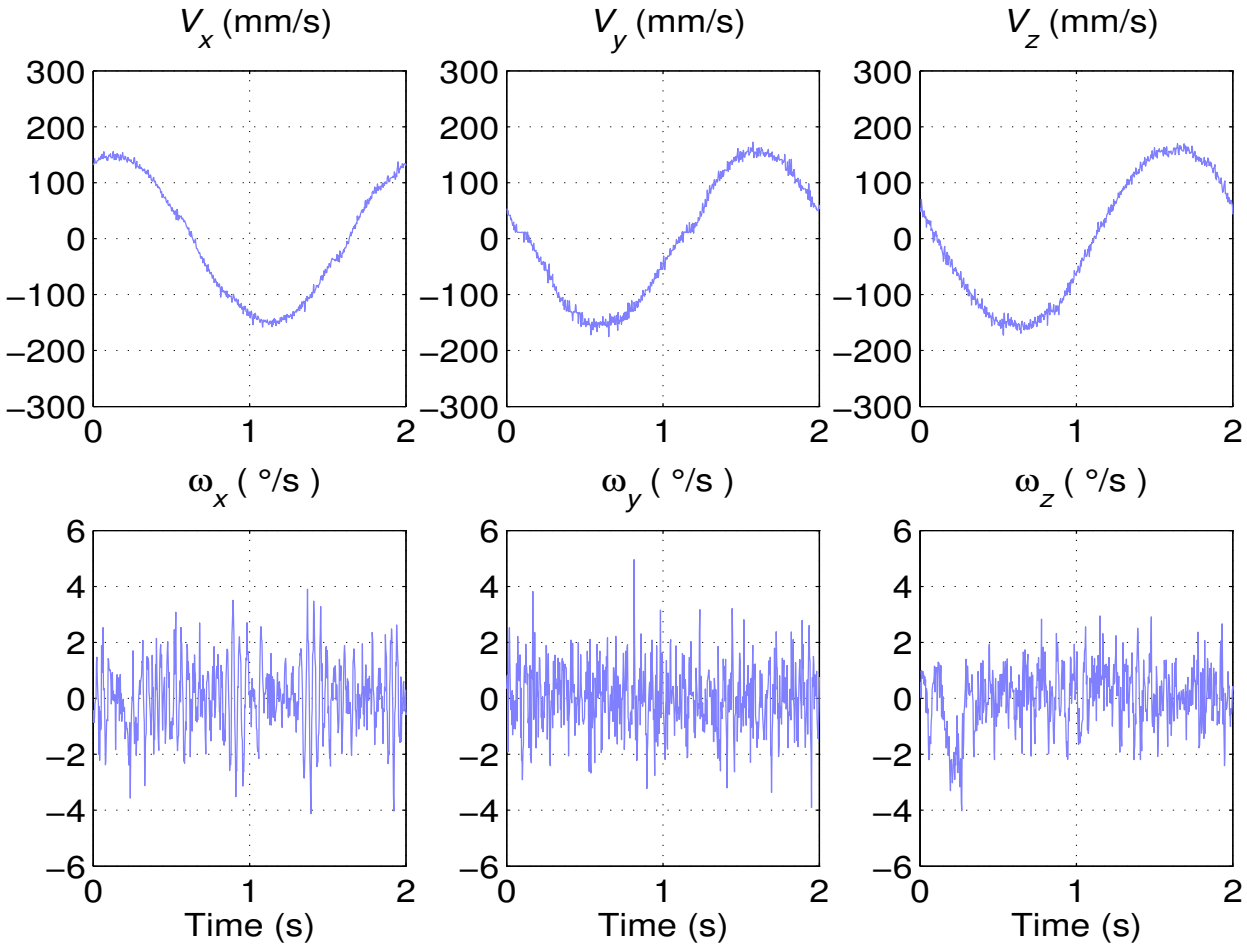

Figure 4.6: Robust observer controller - Velocities obtained from filtering method (f=0.5Hz)

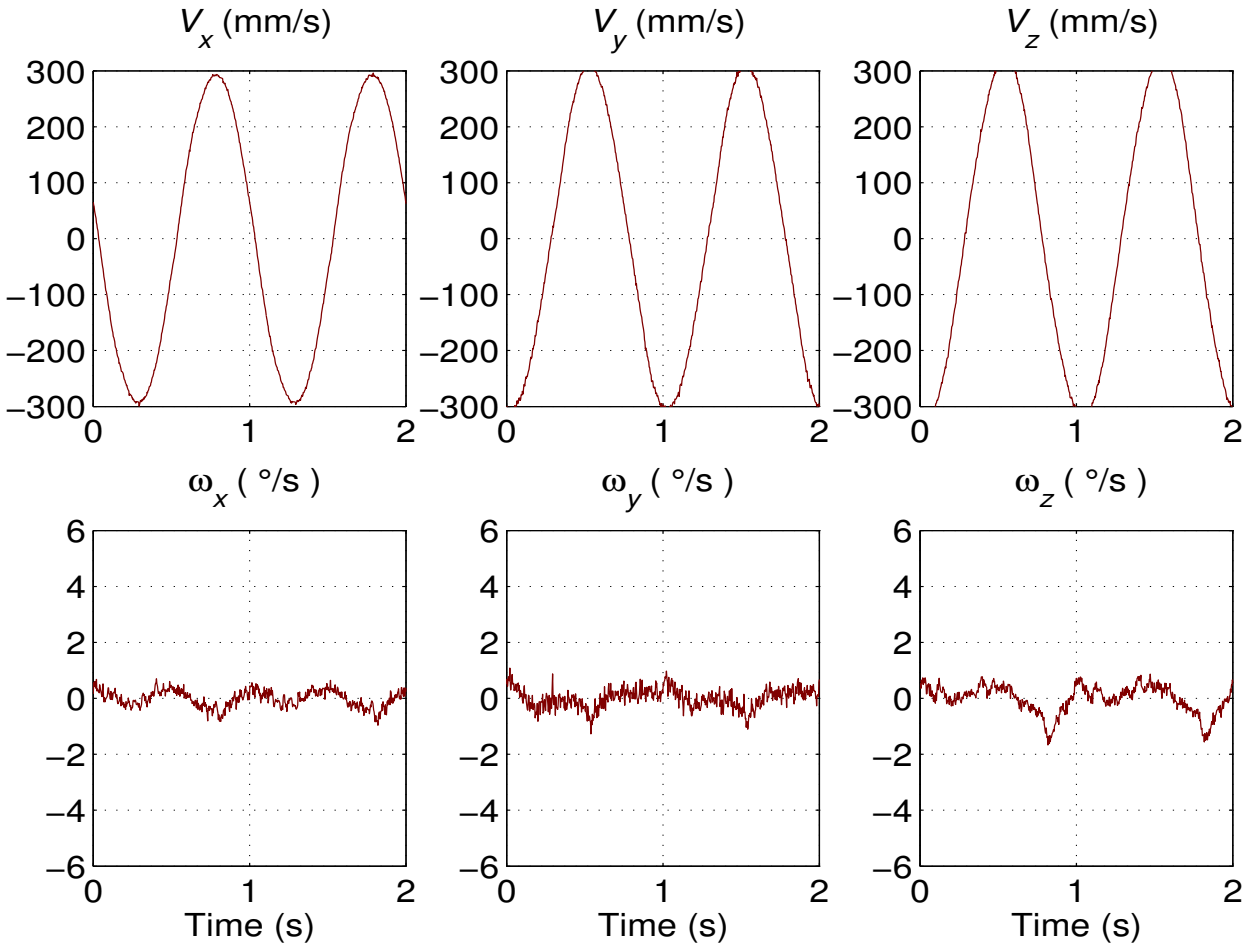

Figure 4.7: Robust observer controller - Velocities obtained from the velocity observer (f=1.0Hz)

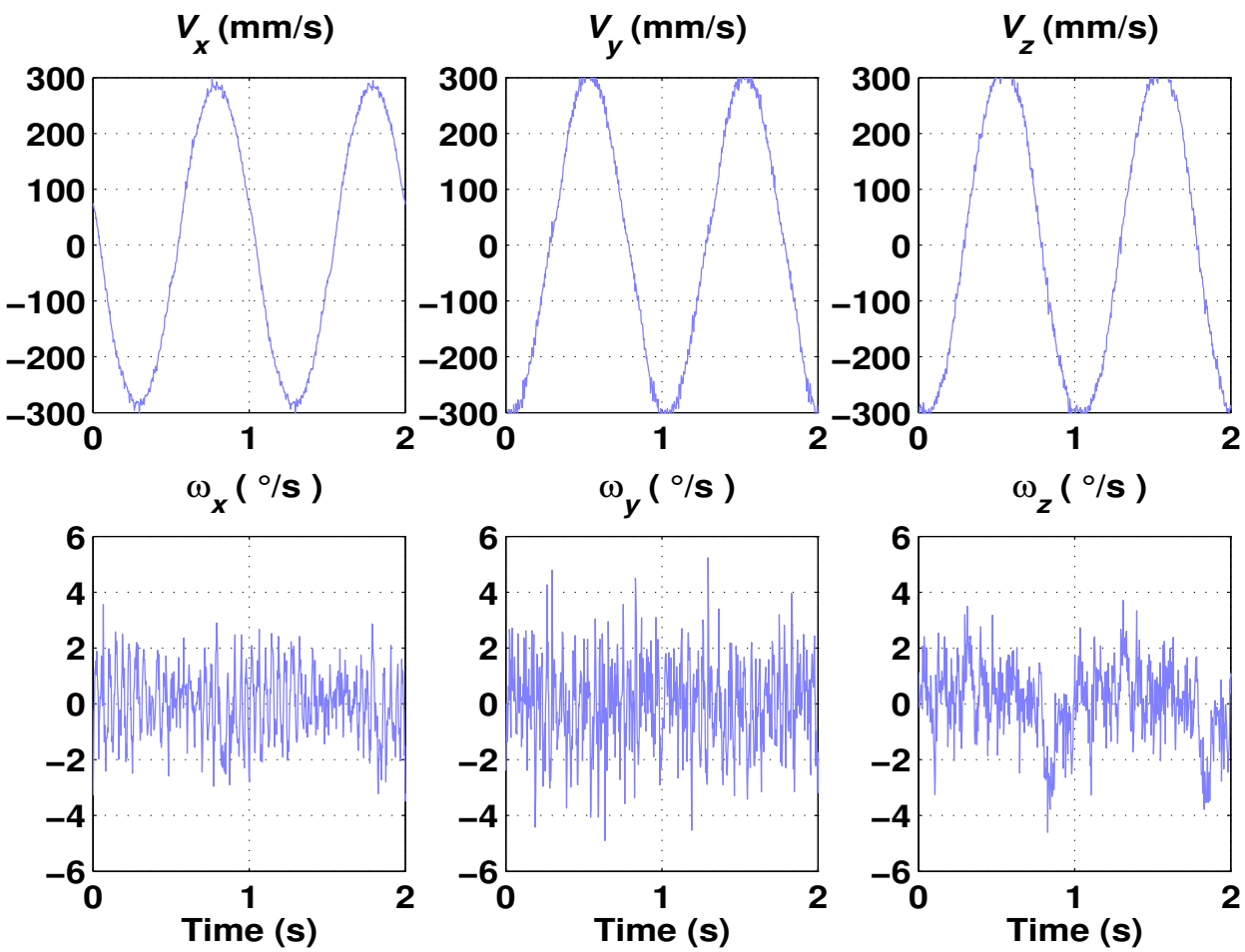

Figure 4.8: Robust observer controller - Velocities obtained from filtering method (f=1.0Hz)

reduce computational burden, in the experiments presented in this chapter, we used first order Euler method for velocity and position estimation and found that, the controller could still achieve higher tracking accuracy. This is due to the introduction of the two terms $k_i\tilde{x}$ in (4.2), and $-k_i\tilde{x}$ in (4.4), which produced more accurate joint velocity and position estimation.

4.4 Conclusions

Based on the observer-controller structure proposed in Chapter 3, we developed a robust operational space observer-controller for trajectory tracking. Experimental results indicate that the performance of the controller is better than the controller introduced in Chapter 3. In addition, compared with the estimated velocities obtained from backwards difference algorithm used in conjunction with a lowpass filter, the observed velocities using the proposed velocity observer are less noisy. Under parametric uncertainty and payload variations, the robust observer-controller can achieve higher position tracking accuracy than the controller employing filtered velocity. The result of the controller has been published in [70].

The controller developed in Chapter 3 has the control over three kinds of errors, i.e. position and velocity tracking errors, plus velocity observation errors, while the proposed controller in this chapter further improve the control performance through gaining control over four kinds of errors, i.e. position and velocity tracking errors, plus position and velocity observation errors.

The controller presented in this chapter can provide more accurate velocity information than the controller in Chapter 3. However, it is also semi-global stable, hence, it may not be suitable to use it under large degree of uncertainties.

CHAPTER 5

CONTROL ALGORITHM 3: ADAPTIVE FRICTION IDENTIFICATION AND COMPENSATION VIA ROBUST OBSERVER-CONTROLLER

5.1 Introduction

An operational space controller that employs a velocity observer and a friction adaptation law to achieve higher tracking accuracy is presented in this chapter. Without velocity measurements, the overall observer-controller system can achieve a semi-global asymptotic stability for the position and velocity tracking errors, and position and velocity estimation errors, with estimated friction coefficients converging asymptotically. Experimental results indicate that the proposed adaptive observer-controller is able to achieve much higher tracking accuracy than the observer-controller without friction compensation, which verify the effectiveness of the adaptive controller.

5.2 Adaptive Observer-Controller Formulation

Our proposed adaptive observer-controller consists of a model-based velocity observer, a controller that is formulated in operational space, plus friction adaptation law.

5.2.1 Formulation of Operational Space Velocity Observer

Based on the observer-controller presented in Chapter 4, we introduce the friction estimation term $\hat{f}$ in (4.2) to formulate the following new velocity observer:

$$\dot{\hat{x}} = y + k\tilde{x} \tag{5.1}$$

$$\dot{y} = \Lambda(x)^{-1}\left[F - \Psi(x,\dot{\hat{x}})\dot{\hat{x}} - p(x) - \hat{f} + k_i\tilde{x}\right] \tag{5.2}$$

where $\hat{f}$ is the estimated friction term given later in (5.20). $\dot{\hat{x}}$, y, and F have been defined in Chapter 3, and k_i defined in Chapter 4.

k is a positive scalar constant defined by:

$$k = \frac{1}{m_1}\left[\zeta_c\zeta_d + \zeta_c k_0 + \zeta_c k_s k_0 + k_s + 2k_n + \zeta_e\right] \tag{5.3}$$

where k_0, k_s and k_n, ζ_c, and ζ_d were defined in Chapter 3. ζ_e is defined in (5.9).

5.2.2 Formulation of Friction Adaptation Law

We assume that the friction term τ_f in (2.5) is uncoupled among the joints, so that

$$\tau_f = \text{vec}\{\tau_{f_i}(\dot{q}_i)\} \equiv \begin{bmatrix} \tau_{f_1}(\dot{q}_1) \\ . \\ . \\ . \\ \tau_{f_n}(\dot{q}_n) \end{bmatrix} \tag{5.4}$$

with $\tau_{f_i}(\cdot)$ known scalar functions that may be determined for any given arm. Here we have defined the vec$\{\cdot\}$ function for future use.

We assume that the viscous frictions have the form:

$$\tau_{vis}\dot{q} = \text{vec}\{\tau_{vis_i}\dot{q}_i\} \tag{5.5}$$

with τ_{vis_i} constant coefficients. Then τ_{vis} =diag$\{\tau_{vis_i}\}$, a diagonal matrix with entries τ_{vis_i}.

The viscous friction term has the following property [71]:

$$\|\tau_{vis}\dot{q}\| \leq \zeta_v \|\dot{q}\| \tag{5.6}$$

where ζ_v is a positive scalar.

The relationship between a robot end-effector velocity $\dot{x}$ and joint velocity $\dot{q}$ can be expressed as:

$$\dot{x} = J(q)\dot{q}$$

For a non-redundant robot, in the non-singular region, the joint velocity $\dot{q}$ can be obtained by:

$$\dot{q} = J^{-1}(q)\dot{x} \tag{5.7}$$

and from (5.7) we can get [72]:

$$\|\dot{q}\| \leq \|J^{-1}\|_{i2} \|\dot{x}\|$$

thus the following result can be obtained:

$$\|\tau_{vis}\dot{q}\| \leq \zeta_e \|\dot{x}\| \tag{5.8}$$

where ζ_e is defined as:

$$\zeta_e = \zeta_v \|J^{-1}\|_{i2} \tag{5.9}$$

We will use this property for our controller development.

Assume that the Coulomb friction has the form:

$$\tau_{cou}\text{sgn}(\dot{q}) = \text{vec}\{\tau_{cou_i}\text{sgn}(\dot{q}_i)\} \tag{5.10}$$

with τ_{cou_i} constant coefficients, and τ_{cou} =diag$\{\tau_{cou_i}\}$.

The friction term $\tau_{sti}\exp(-\tau_{dec}\dot{q}^2)\mathrm{sgn}(\dot{q})$ in (2.5) is the combination of static friction and Stribeck effect, and we assume that it can be written in the following form [60]:

$$\tau_{sti}\exp(-\tau_{dec}\dot{q}^2)\mathrm{sgn}(\dot{q}) = \mathrm{vec}\{\tau_{sti_i}\exp(-\tau_{dec_i}\dot{q}_i^2)\mathrm{sgn}(\dot{q}_i)\} \tag{5.11}$$

with τ_{sti_i} and τ_{dec_i} constant coefficients, and τ_{sti} =diag$\{\tau_{sti_i}\}$, τ_{dec} =diag$\{\tau_{dec_i}\}$.

The joint space robot friction model (2.5) can be written in the following linear-in-the-parameters form:

$$\tau_f = W_j(\dot{q})\theta \tag{5.12}$$

where $W_j(\dot{q})$ is the $n \times 3n$ regression vector given by:

$$\begin{array}{lcl} W_j(\dot{q}) & = & \left[\begin{array}{ccc} w_{j1}(\dot{q}) & w_{j2}(\dot{q}) & w_{j3}(\dot{q}) \end{array}\right] \\ w_{j1}(\dot{q}) & = & \mathrm{diag}(\dot{q}) \\ w_{j2}(\dot{q}) & = & \mathrm{diag}(\mathrm{sgn}(\dot{q})) \\ w_{j3}(\dot{q}) & = & \mathrm{diag}(\mathrm{sgn}(\dot{q})\exp(-\tau_{dec}\dot{q}^2)) \end{array} \tag{5.13}$$

and θ is the $3n \times 1$ vector of constant parameters defined as:

$$\theta = \left[\begin{array}{ccc} \mathrm{vec}^T\{\tau_{vis_i}\} & \mathrm{vec}^T\{\tau_{cou_i}\} & \mathrm{vec}^T\{\tau_{sti_i}\} \end{array}\right]^T \tag{5.14}$$

Here we assume that the coefficients τ_{vis}, τ_{cou} and τ_{sti} are unknown constants, but the Stribeck parameters τ_{dec} are assumed to be known.

The friction parameter estimate vector $\hat{\theta}$ is updated using the following adaptation algorithm:

$$\dot{\hat{\theta}} = -K_{ad}W_j^T(\dot{\hat{q}})\dot{\tilde{q}} \tag{5.15}$$

where K_{ad} is a $3n \times 3n$ diagonal, positive-definite, adaptation gain matrix; the joint velocity observation error $\dot{\tilde{q}}$ is defined as the difference between the actual joint velocity and the observed joint velocity:

$$\dot{\tilde{q}} = \dot{q} - \dot{\hat{q}} \tag{5.16}$$

In the non-singular region of a robot, the observed joint velocity $\dot{\hat{q}}$ can be obtained by:

$$\dot{\hat{q}} = J^{-1}(q)\dot{\hat{x}} \tag{5.17}$$

where the observed end-effector velocity $\dot{\hat{x}}$ is calculated by (5.1).

$W_j(\dot{\hat{q}})$ is the $n \times 3n$ regression vector given by:

$$\begin{array}{lcl} W_j(\dot{\hat{q}}) & = & \left[\begin{array}{ccc} \hat{w}_{j1}(\dot{\hat{q}}) & \hat{w}_{j2}(\dot{\hat{q}}) & \hat{w}_{j3}(\dot{\hat{q}}) \end{array} \right] \\ \hat{w}_{j1}(\dot{\hat{q}}) & = & \mathrm{diag}(\dot{\hat{q}}) \\ \hat{w}_{j2}(\dot{\hat{q}}) & = & \mathrm{diag}(\mathrm{sgn}(\dot{\hat{q}})) \\ \hat{w}_{j3}(\dot{\hat{q}}) & = & \mathrm{diag}(\mathrm{sgn}(\dot{\hat{q}}) \exp(-\tau_{dec}\dot{\hat{q}}^2)) \end{array} \tag{5.18}$$

and the estimated joint frictions are obtained by:

$$\hat{\tau}_f = W_j(\dot{\hat{q}})\hat{\theta} \tag{5.19}$$

From (5.19), the estimated frictions in operational space can be obtained by:

$$\hat{f} = J^{-T}\hat{\tau}_f \tag{5.20}$$

5.2.3 Formulation of Operational Space Controller

By using the estimated velocity $\dot{\hat{x}}$ proposed in section 5.2.1, the following model-based controller is formulated to generate the required driving force:

$$F = (k_s + k_{nd})\eta_p + w_e - k_i\tilde{x} \tag{5.21}$$

where k_{nd} is a positive controller gain defined as:

$$k_{nd} = 2k_n + \zeta_c k_0 + (k_s m_2 + k m_2)^2 k_n \tag{5.22}$$

The $n \times 1$ observed filtered tracking error signal η_p is defined as:

$$\eta_p = \dot{x}_d + k_s e - \dot{\hat{x}} \tag{5.23}$$

and the $n \times 1$ auxiliary vector w_e is defined as:

$$w_e = \Lambda(x)[\ddot{x}_d + k_s(\dot{x}_d - \dot{\hat{x}})] + \Psi(x, \dot{\hat{x}})(\dot{x}_d + k_s e) + p(x) + \hat{f} \tag{5.24}$$

where the $n \times 1$ end-effector position and orientation tracking error e is defined as:

$$e = x_d - x \tag{5.25}$$

The force command F will be used in the observer indicated by (5.2). And the torque commands for driving the robot can be obtained by:

$$\Gamma = J^T(q)F$$

5.3 Overall System Stability Result and Analysis

Theorem 3 *Under the assumption that the exact model of a robot except friction is known, if the observer-controller gains satisfy the following sufficient conditions :*

$$\begin{array}{rcl} k_s & > & 1/k_n + \eta_e \\ k_0 & > & \|err(0)\| \end{array} \tag{5.26}$$

the closed-loop tracking error system is stable and the errors $e(t)$, $\dot{e}(t)$, $\tilde{x}(t)$, $\dot{\tilde{x}}(t)$, and $(\hat{\theta}(t) - \theta)$ are bounded

where

$$err = \begin{bmatrix} \eta_p^T & e^T & \dot{\tilde{x}}^T & \tilde{x}^T \end{bmatrix}^T \in \Re^{4n} \tag{5.27}$$

We will now present the stability proof using Lyapunov stability analysis. To determine the stability of the overall closed-loop control system, we define the following Lyapunov function:

$$V = V_0 + V_1 + V_2 \tag{5.28}$$

where the three sub-Lyapunov functions V_0, V_1, and V_2 are defined as:

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T k_i \tilde{x} + \frac{1}{2}\tilde{\theta}^T K_{ad}^{-1}\tilde{\theta} \tag{5.29}$$

where $\tilde{\theta}$ is the difference between the actual θ and the estimated $\hat{\theta}$, and the velocity observation error $\dot{\tilde{x}}$ is defined as the difference between the actual end-effector velocity $\dot{x}$ and the observed end-effector velocity $\dot{\hat{x}}$, it is obtained by differentiating (3.5) with respect to time:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \tag{5.30}$$

$$V_1 = \frac{1}{2}e^T e \tag{5.31}$$

$$V_2 = \frac{1}{2}\eta_p^T \Lambda(x)\eta_p \tag{5.32}$$

Hence, $\dot{V}$ can be obtained by:

$$\dot{V} = \dot{V}_0 + \dot{V}_1 + \dot{V}_2 \tag{5.33}$$

We will formulate the bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$ separately, and then combine them together to get the bound of $\dot{V}$.

5.3.1 Lyapunov Function for Observation Error $\tilde{x}$, $\dot{\tilde{x}}$ and $\tilde{\theta}$

V_0 is defined in (5.29). To form the bound of $\dot{V}_0$, first, take the time derivative of (5.1) and then substitute (5.2) into the resulting expression to yield:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x,\dot{\hat{x}})\dot{\hat{x}} + p(x) + \hat{f} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} = F \tag{5.34}$$

Subtract (5.34) from (2.7), use (A.3) and (5.30) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x,\dot{x})\dot{\tilde{x}} + \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} + k_i\tilde{x} + f - \hat{f} = 0 \tag{5.35}$$

Differentiate V_0 along (5.35) to get:

$$\begin{aligned}\dot{V}_0 = \quad & \dot{\tilde{x}}^T \left[-\Psi(x,\dot{x})\dot{\tilde{x}} - \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x}\right] \\ & + \frac{1}{2}\dot{\tilde{x}}^T \dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i\tilde{x} - \dot{\tilde{x}}^T(f - \hat{f})\end{aligned} \tag{5.36}$$

Utilizing (A.2) to get:

$$\dot{V}_0 = -\dot{\tilde{x}}\left[\Psi(x,\dot{\hat{x}}) + k\Lambda(x)\right]\dot{\tilde{x}} - \dot{\tilde{x}}^T(f - \hat{f}) \tag{5.37}$$

Then utilize (A.1) and (A.4) to get the upper bound of $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\|\dot{\hat{x}}\| - km_1\right)\|\dot{\tilde{x}}\|^2 + \zeta_e\|\dot{\tilde{x}}\|^2 \tag{5.38}$$

Substitute for $\dot{\hat{x}}$ from (5.23) into (5.38), and utilize (3.2) to get the new upper bound for $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\zeta_d + \zeta_c\|\eta_p\| + \zeta_c k_s\|e\| - km_1\right)\|\dot{\tilde{x}}\|^2 + \zeta_e\|\dot{\tilde{x}}\|^2 \tag{5.39}$$

5.3.2 Lyapunov Function for Tracking Error e

The evolution of the position tracking error with time can be derived by differentiating (5.25) with respect to time to yield:

$$\dot{e} = \dot{x}_d - \dot{x}$$

Since $\dot{x}$ is not measurable, we use the estimated term $\dot{\hat{x}}$ to eliminate $\dot{x}$ and get the following equation:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} - \dot{\tilde{x}} \tag{5.40}$$

Simplify (5.40) by utilizing (5.23) to get:

$$\dot{e} = -k_s e + \eta_p - \dot{\tilde{x}} \tag{5.41}$$

V_1 is defined in (5.31), and the upper bound for the time derivative of V_1 along (5.41) is given by:

$$\dot{V}_1 \leq -k_s\|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{5.42}$$

5.3.3 Lyapunov Function for η_p

The evolution of η_p with time can be derived by differentiating (5.23) with respect to time, multiplying both sides of the resulting expression by $\Lambda(x)$, and substituting the expression from (5.34) for $\ddot{\tilde{x}}$ to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= \Lambda(x)\ddot{x}_d + k_s\Lambda(x)(\dot{x}_d - \dot{x}) - k\Lambda(x)\dot{\tilde{x}} \\ &\quad -k_i\tilde{x} + \Psi(x,\dot{\hat{x}})\dot{\hat{x}} + p(x) + \hat{f} - F \end{aligned} \tag{5.43}$$

Substitute the force input given by (5.21) into (5.43), use the definitions of w_e and η_p to get:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -(k_s + k_{nd})\eta_p - (k + k_s)\Lambda(x)\dot{\tilde{x}} \\ &\quad -\Psi(x,\dot{\hat{x}})\eta_p \end{aligned} \tag{5.44}$$

Rewrite the term $\Psi(x,\dot{\hat{x}})\eta_p$ on the right-hand side of (5.44) in terms of $\dot{\tilde{x}}$, and utilize (A.3) and (5.30) to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -\Psi(x,\dot{x})\eta_p - (k_s + k_{nd})\eta_p \\ &\quad -(k + k_s)\Lambda(x)\dot{\tilde{x}} + \Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{5.45}$$

V_2 is defined in (5.32). Differentiating V_2 along (5.45) and utilizing (A.2) yields:

$$\begin{aligned} \dot{V}_2 &= -(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &\quad +\eta_p^T\Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{5.46}$$

From (5.46), and using (A.1) and (A.4), we can obtain the following upper bound for $\dot{V}_2$:

$$\begin{aligned} \dot{V}_2 &\leq -(k_s + k_{nd})\|\eta_p\|^2 + (k + k_s)m_2\|\eta_p\|\|\dot{\tilde{x}}\| \\ &\quad +\zeta_c\|\eta_p\|^2\|\dot{\tilde{x}}\| \end{aligned} \tag{5.47}$$

5.3.4 Overall System Stability Analysis

Use the upper bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$, and utilize (5.3), (5.22), and (5.27), we can form the upper bound on $\dot{V}$:

$$\begin{aligned}\dot{V} \leq\ & -k_s \|e\|^2 - k_s \left\|\eta_p\right\|^2 - k_s \left\|\dot{\tilde{x}}\right\|^2 \\ & + \left\|\eta_p\right\| \left(\|e\| - 2k_n \left\|\eta_p\right\|\right) + \left\|\dot{\tilde{x}}\right\| \left(\|e\| - 2k_n \left\|\dot{\tilde{x}}\right\|\right) \\ & + (k+k_s)\, m_2 \left\|\eta_p\right\| \left(\left\|\dot{\tilde{x}}\right\| - (k+k_s)\, m_2 k_n \left\|\eta_p\right\|\right) \\ & + \left(\zeta_c\zeta_d \left\|\dot{\tilde{x}}\right\|^2 - \zeta_c\zeta_d \left\|\dot{\tilde{x}}\right\|^2\right) + \zeta_e \|\dot{\tilde{x}}\|^2 \\ & - (k_o - \|err\|) \left(\zeta_c \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c \left\|\eta_p\right\|^2\right)\end{aligned} \tag{5.48}$$

where we have used the fact derived from (5.27) that $\|err\| \geq \|e\|, \|\eta\|$, and $\|\dot{\tilde{x}}\|$.

By applying the nonlinear damping tool [66] on the terms in the second and third lines on the right hand side of (5.48), a new upper bound on $\dot{V}$ can be formed as:

$$\begin{aligned}\dot{V} \leq\ & -(k_s - \tfrac{1}{k_n}) \|e\|^2 - k_s \left\|\eta_p\right\|^2 - (k_s - \tfrac{1}{k_n} - \zeta_e) \left\|\dot{\tilde{x}}\right\|^2 \\ & - (k_o - \|err\|) \left(\zeta_c \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c \left\|\eta_p\right\|^2\right)\end{aligned} \tag{5.49}$$

From (5.49) we can see that, if $k_s > 1/k_n + \zeta_e$ and $k_0 \geq ||err||$, we can get:

$$\dot{V} \leq 0 \tag{5.50}$$

From (5.50), we can get the conclusion that, friction coefficients estimation error $\tilde{\theta}$, position tracking errors e, position estimation errors $\widetilde{x}$, velocity estimation errors $\dot{\widetilde{x}}$, observed filtered tracking error signal η_p of the observer-controller are all bounded. Furthermore, the end-effector velocity tracking error is also bounded. In fact, after adding and subtracting $\dot{x}$ to the right-hand side of (5.23) and rearranging the terms,

we can formulate the following inequality:

$$\|\dot{e}\| = \|\dot{x}_d - \dot{x}\| \leq \left\|\eta_p\right\| + k_s \|e\| + \|\dot{x}\| \tag{5.51}$$

Since each of the terms on the right-hand side of the above equation is bounded, $\|\dot{e}\|$ is also bounded. This yields the result indicated by *Theorem 3*.

5.4 Implementation of Friction Adaptation Law

The friction adaptation law indicated by (5.15) consists of the actual joint velocity and the observed joint velocity. Since we assume that the actual joint velocity is unknown, in order to implement this friction adaptation law, both sides of (5.15) are integrated, giving:

$$\hat{\theta}(t) = \hat{\theta}(t - \Delta t) - K_{ad} \int_{t-\Delta t}^{t} W_j^T(\dot{\hat{q}})(\frac{dq}{dt} - \dot{\hat{q}})dt \tag{5.52}$$

where Δt represents the sampling time of the system.

In our experiment, the sampling time Δt is selected to be a reasonably small value of 1ms. Based on this condition, from (5.52) we can get the following form:

$$\begin{array}{rcl} \hat{\theta}(t) & = & \hat{\theta}(t - \Delta t) - K_{ad} W_j^T(\dot{\hat{q}}(t - \Delta t)) D_q \\ D_q & = & q(t) - q(t - \Delta t) - \dot{\hat{q}}(t - \Delta t)\Delta t \end{array} \tag{5.53}$$

where $\hat{q}$ is obtained from (5.7).

We will use (5.53) as our friction adaptation algorithm.

5.5 Experimental Results

The experiments were performed using PUMA 560 robot, and the sampling time is selected to be 1ms.

The defined trajectory is to move the end-effector in XYZ direction with the desired position trajectory indicated by (5.54), while maintaining the initial end-effector orientation constant all the time.

$$\begin{aligned} p_{x_d} &= p_{x_0} + 50.0\sin(0.4\pi t)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{y_d} &= p_{y_0} + 50.0\cos(0.4\pi ft)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{z_d} &= p_{z_0} + 50.0\cos(0.4\pi ft)\left(1 - e^{-0.05t^3}\right) \text{mm} \end{aligned} \tag{5.54}$$

where p_{x_0}, p_{y_0} and p_{z_0} are the initial positions of the robot. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

The controller gains were selected as diagonal gains matrices as following:

$$\begin{aligned} k_{nd} &= \text{diag}\{120, 120, 120, 35, 35, 35\} \\ k &= \text{diag}\{108, 108, 108, 32, 32, 32\} \\ k_s &= \text{diag}\{97, 97, 97, 30, 30, 30\} \\ k_i &= \text{diag}\{2000, 2000, 2000, 3000, 3000, 3000\} \end{aligned} \tag{5.55}$$

All the diagonal terms of the 18×18 friction adaptation gains K_{ad} were selected to be 500, all the diagonal terms of τ_{dec} were selected to be 1, and all the initial estimated friction coefficients are set to zeros.

5.5.1 Friction Identification and Compensation Performance

Using the trajectory defined by (5.54), the experimental result is shown in Figs. 5.1, 5.3, 5.2, and 5.4. Figs. 5.1 and 5.2 show the initial tracking errors and the identified friction coefficients when the robot just start to move, and Figs. 5.3 and 5.4 shows the parameters when friction adaptation algorithm is activated for about two minutes, respectively. Where J_i stands for Joint i, and e_x, e_y, and e_z are the position tracking errors along X, Y, and Z axis, and e_{ϕ_x}, e_{ϕ_y}, and $e_{\phi_z}$are the orientation tracking errors about X, Y, and Z axis, respectively.

Tables 5.1 and 5.2 show the tracking errors and the identified friction coefficients after the robot ran for about two minutes.

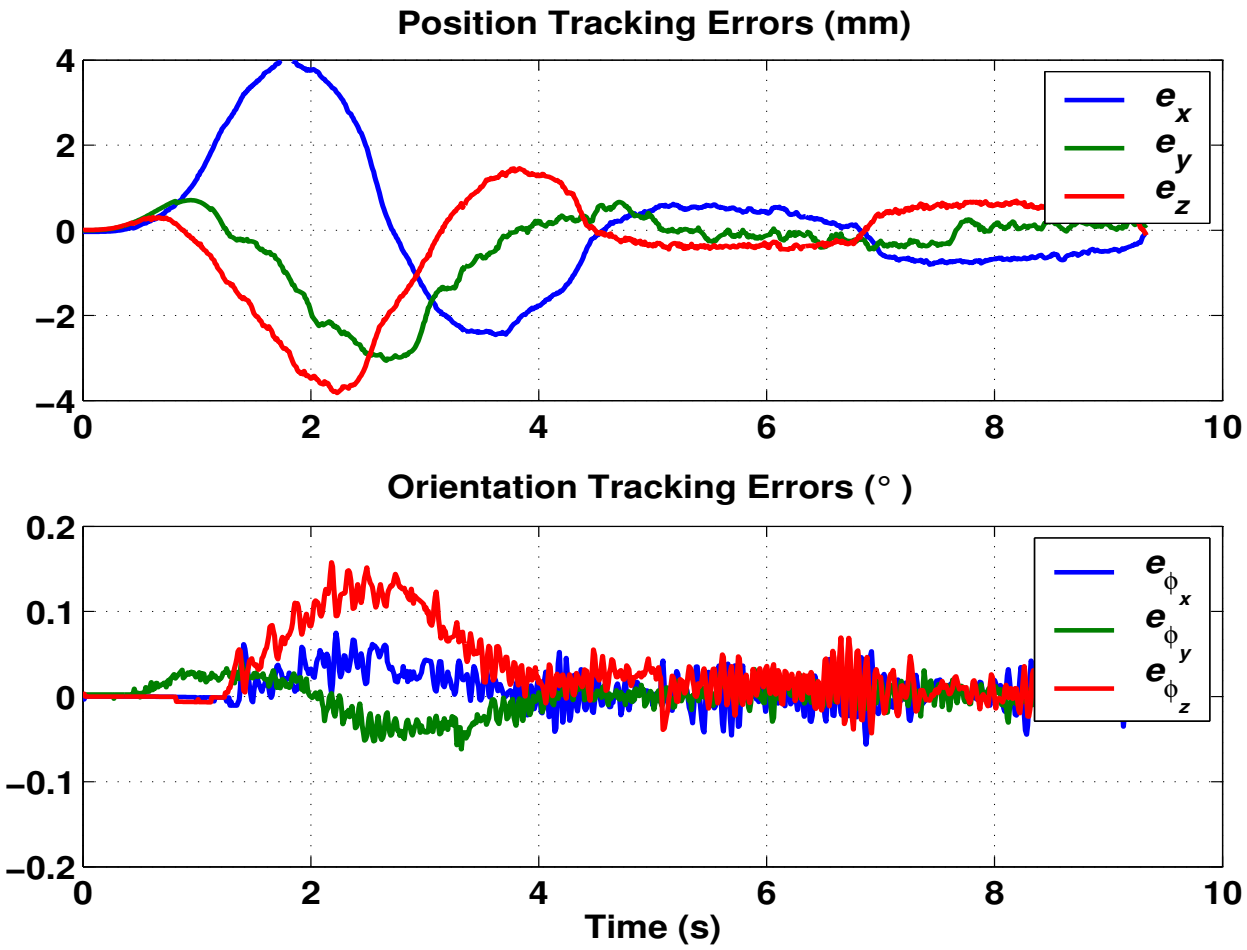

Figure 5.1: Adaptive friction identification and compensation - Initial tracking errors with adaptive friction compensation

Table 5.1: Adaptive friction identification and compensation - Maximum tracking errors with adaptive friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
0.37mm	0.44mm	0.27mm	0.06^o	0.04^o	0.06^o

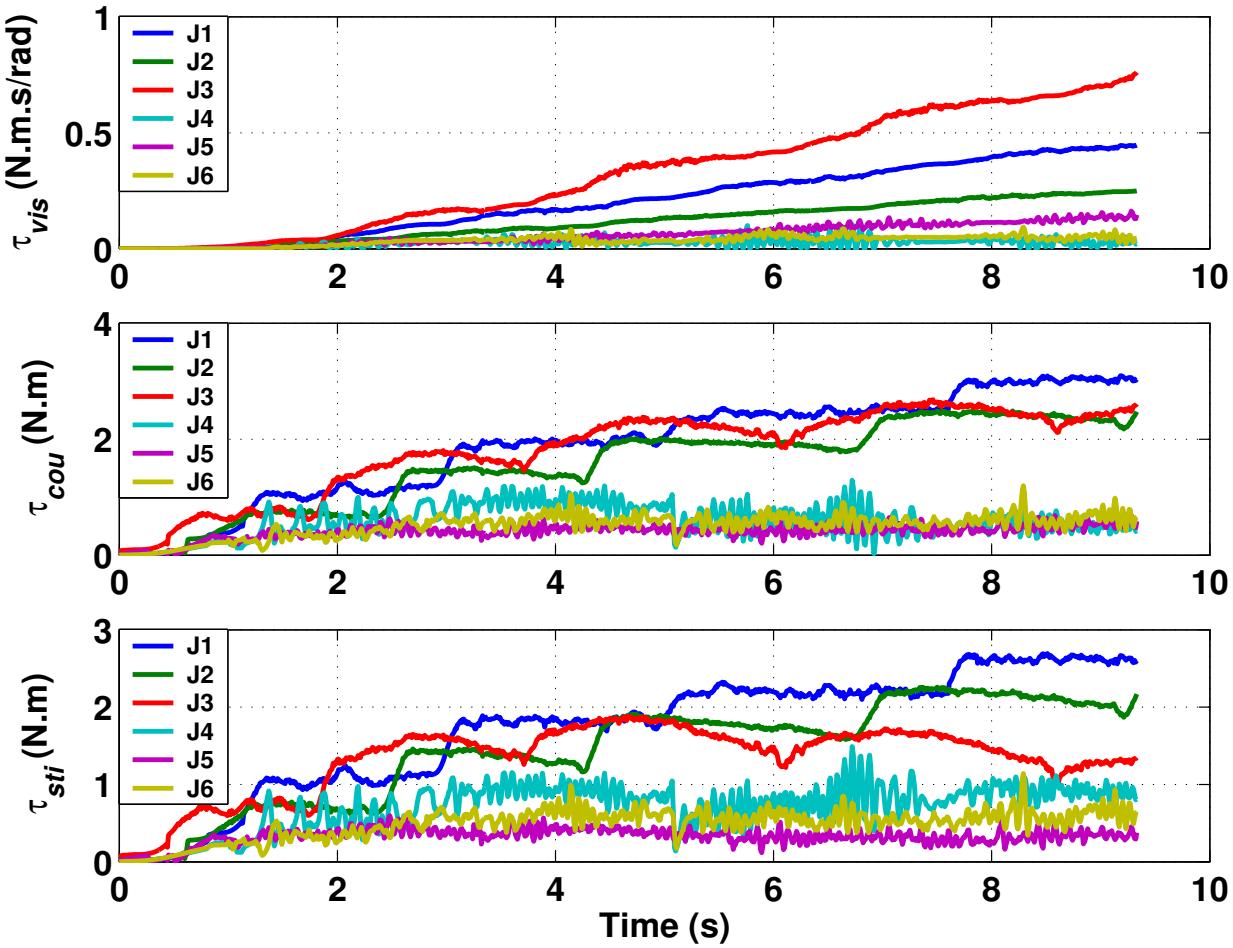

Figure 5.2: Adaptive friction identification and compensation - Initial identified joints friction coefficients

Table 5.2: Adaptive friction identification and compensation - Identified friction coefficients of each joint

	J_1	J_2	J_3	J_4	J_5	J_6
τ_{vis_i} (N.m.s/rad)	5.8	3.7	6.0	0.1	1.9	0.7
τ_{cou_i} (N.m)	4.9	4.8	3.0	1.0	0.6	1.2
τ_{sti_i} (N.m)	4.1	4.3	0.6	1.5	0.4	1.0

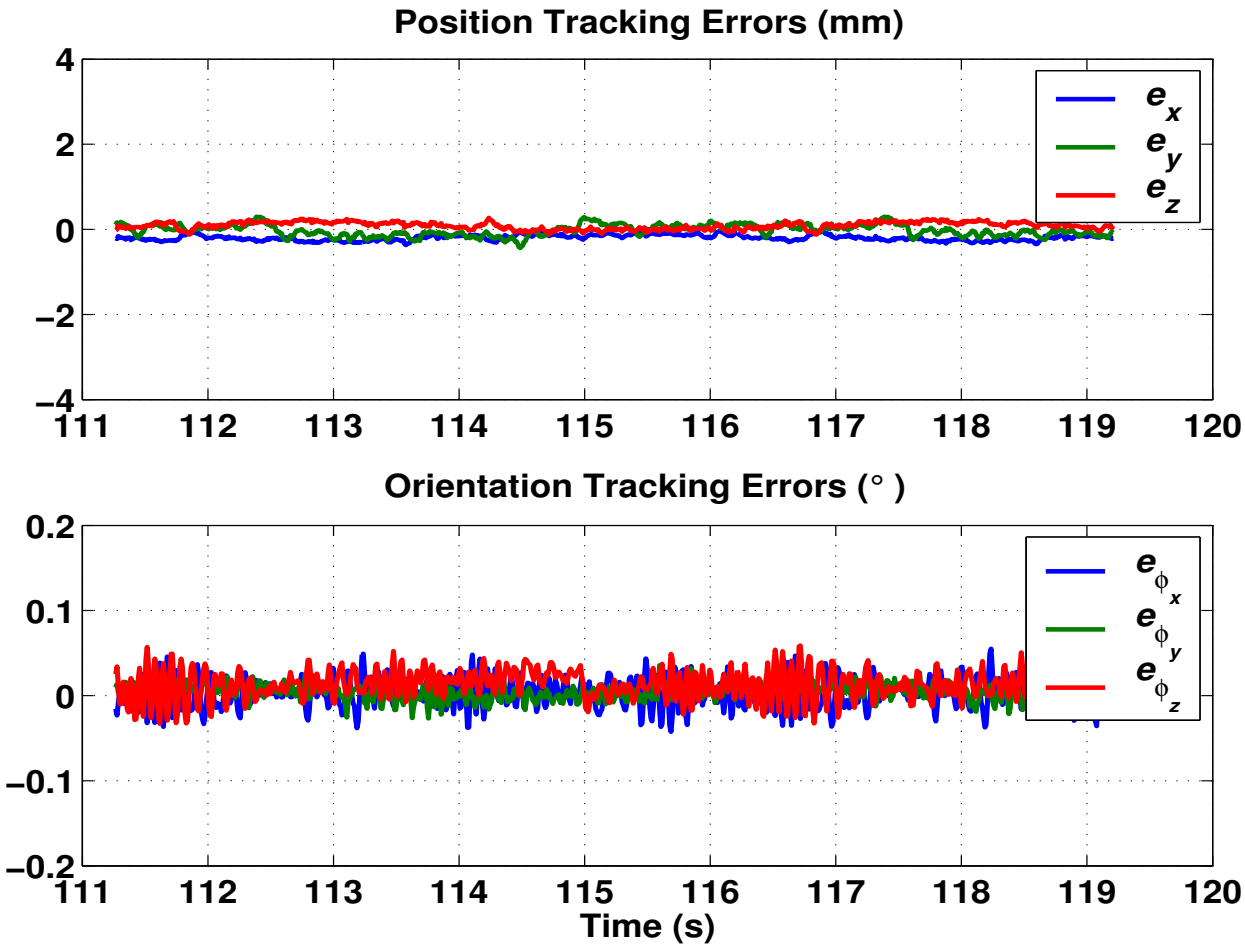

Figure 5.3: Adaptive friction identification and compensation - Tracking errors with adaptive friction compensation

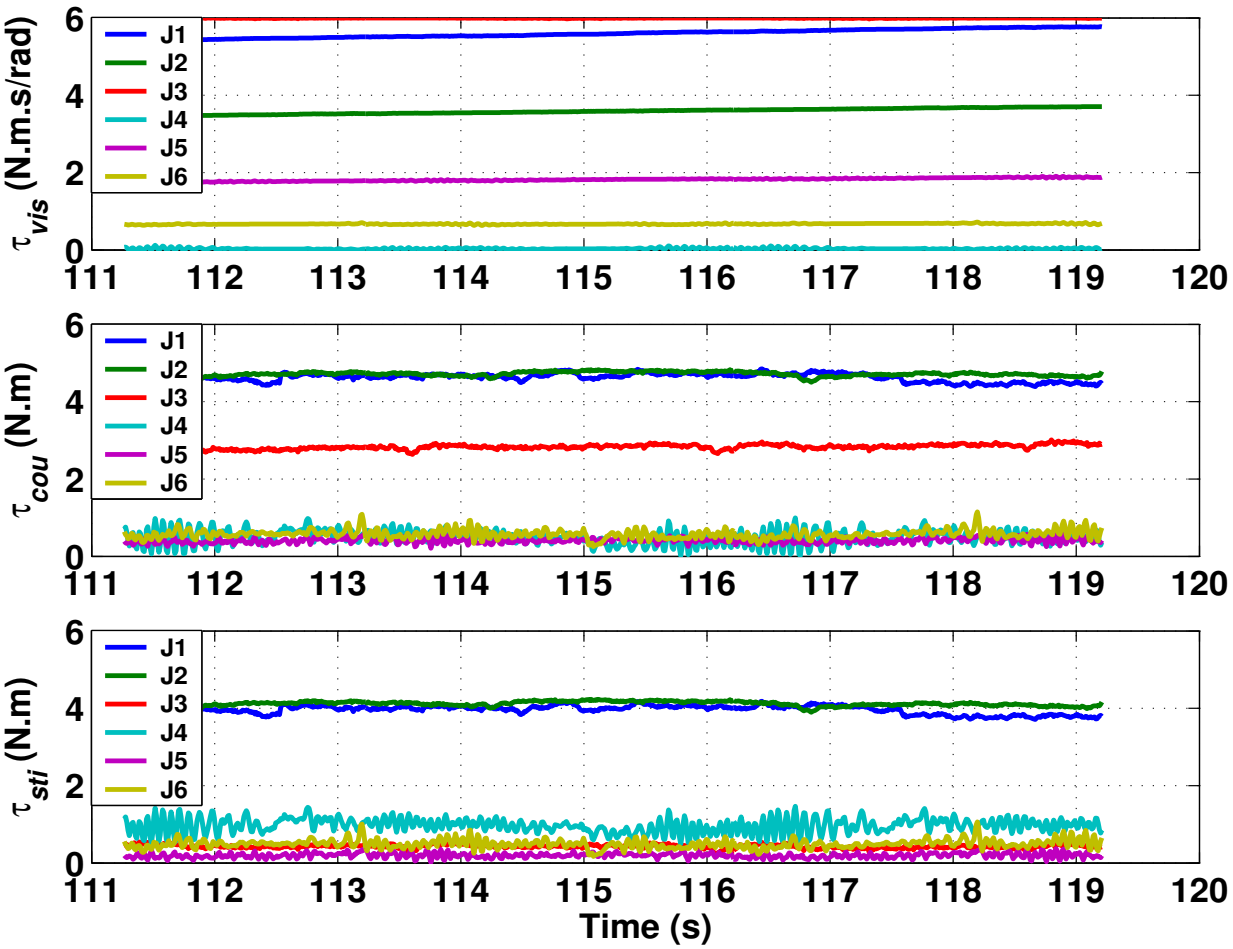

Figure 5.4: Adaptive friction identification and compensation - Final identified joints friction coefficients

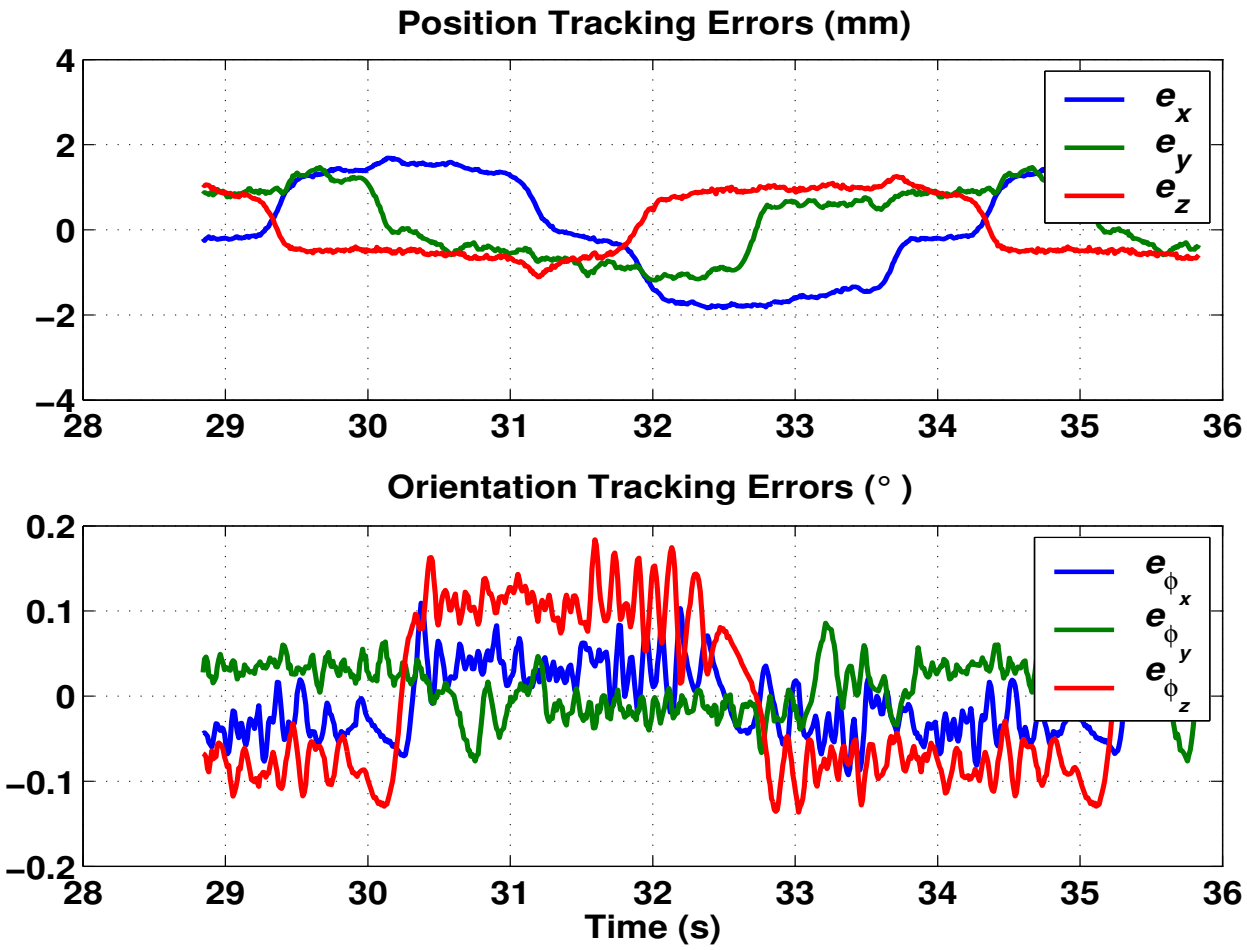

Figure 5.5: Adaptive friction identification and compensation - Tracking errors without friction compensation

Under the same conditions, using the same controllers gains listed in (5.55) but without friction compensation, the result is shown in Fig. 5.5 and Table 5.3.

The results indicate that the tracking errors of the controllers with adaptive friction compensation is about 2 to 5 times smaller than the controller without friction compensation, which verify the effectiveness of the proposed adaptive controller.

Table 5.3: Adaptive friction identification and compensation - Maximum tracking errors without friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
1.84mm	1.58mm	1.30mm	0.11^o	0.09^o	0.19^o

5.6 Conclusions

Based on the observer-controller presented in Chapter 4, in this chapter, we propose an operational space observer-controller with adaptive friction compensation capability. The friction adaptation algorithm is designed to make use of the merits of the "cleaner" observed velocity to achieve better performance. Experimental results using PUMA 560 indicate that the proposed adaptive controller is able to achieve higher tracking accuracy than the observer-controller without friction compensation, which verifies the effectiveness of the control algorithm. The result of the controller has been published in [73].

The main contributions of this controller is that, we have successfully incorporated adaptive friction compensation law into the observer-controller presented in Chapter 4 to achieve friction identification and compensation.

When developing this adaptive controllers, we assumed that we possessed exact knowledge of a robot dynamic model except friction. However, when a robot model is not accurate enough, the accuracy of friction identification will be affected.

CHAPTER 6

CONTROL ALGORITHM 4: ADAPTIVE FRICTION IDENTIFICATION AND COMPENSATION VIA FILTERED VELOCITY

6.1 Introduction

There are many ways of compensating joint frictions. In this chapter, we introduce an adaptive controller that employs a friction adaptation law. Different from the adaptive controller presented in Chapter 5, which used observed velocity information, the controller in this chapter makes use of filtered velocity. The overall adaptive control system can achieve a global asymptotical stability for the position and velocity tracking errors in the presence of friction coefficients uncertainties. Experimental results indicate that the proposed adaptive controller is able to achieve much smaller tracking errors than the controller without friction compensation.

6.2 Adaptive Controller Formulation

The control objective is to achieve position/velocity tracking despite the uncertainty associated with the parameter vector θ defined in (5.14).

The $n \times 1$ end-effector position and orientation tracking error e is defined as the difference between the desired end-effector position and orientation x_d, and the actual end-effector position and orientation x:

$$e = x_d - x \tag{6.1}$$

Let the ∞-norm (denoted by $\| \cdot \|_\infty$) of $f(t)$ be defined as:

$$\|f(t)\|_\infty = \sup_t \mid f(t) \mid \tag{6.2}$$

If $\|f(t)\|_\infty < \infty$, then we say that the function $f(t)$ belongs to the subspace $\pounds_\infty$ of the space of all possible functions (i.e., $f(t) \in \pounds_\infty$).

The desired end-effector position and orientation x_d must be constructed to ensure that $x_d(t)$, $\dot{x}_d(t)$, and $\ddot{x}_d(t) \in \pounds_\infty$.

We define $\dot{e}$ as the difference between the desired end-effector velocity $\dot{x}_d$ and the filtered end-effector velocity $\dot{\hat{x}}$:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} \tag{6.3}$$

The filtered end-effector velocity $\dot{\hat{x}}$ is computed from the filtered joint velocity $\dot{\hat{q}}$ by forward kinematics of the robot as follows:

$$\dot{\hat{x}} = J(q)\dot{\hat{q}}$$

where $\dot{\hat{q}}$ is the estimated joint velocity obtained from backwards difference algorithm used in conjunction with a low pass filter.

The filtered tracking error r is defined as:

$$r = \dot{e} + K_f e \tag{6.4}$$

where K_f is a $n \times n$ diagonal, positive-definite, and constant control gain matrix.

6.2.1 Formulation of Friction Adaptation Law

The friction parameter estimate vector $\hat{\theta}$ is updated using the following adaptation algorithm:

$$\dot{\hat{\theta}} = K_{ad} W_j^T(\dot{q}) J^{-1}(q) r \tag{6.5}$$

where r is defined by (6.4), and $J^{-1}(q)$ is the inverse of the basic Jacobian of a robot. We assume that a robot works in a non-singular region so that $J^{-1}(q)$ exists.

K_{ad} is a $3n \times 3n$ diagonal, positive-definite adaptation gain matrix, and $W_j(\dot{\hat{q}})$ is the $n \times 3n$ regression vector given by:

$$\begin{array}{lcl} W_j(\dot{\hat{q}}) & = & \left[\begin{array}{ccc} \hat{w}_{j1}(\dot{\hat{q}}) & \hat{w}_{j2}(\dot{\hat{q}}) & \hat{w}_{j3}(\dot{\hat{q}}) \end{array} \right] \\ \hat{w}_{j1}(\dot{\hat{q}}) & = & \mathrm{diag}(\dot{\hat{q}}) \\ \hat{w}_{j2}(\dot{\hat{q}}) & = & \mathrm{diag}(\mathrm{sgn}(\dot{\hat{q}})) \\ \hat{w}_{j3}(\dot{\hat{q}}) & = & \mathrm{diag}(\mathrm{sgn}(\dot{\hat{q}}) \exp(-\tau_{dec} \dot{\hat{q}}^2)) \end{array} \tag{6.6}$$

The estimated joint frictions can be obtained by:

$$\hat{\tau}_f = W_j(\dot{\hat{q}}) \hat{\theta} \tag{6.7}$$

From (6.7), the estimated frictions in operational space can be obtained by:

$$\hat{f}_f = J^{-T} \hat{\tau}_f \tag{6.8}$$

6.2.2 Formulation of Operational Space Controller

By using the estimated velocity $\dot{\hat{x}}$, the model-based controller is formulated to generate the required driving force:

$$\begin{array}{rl} F = & \Lambda(x)[\ddot{x}_d + K_f \dot{e}] + k\Lambda(x) r \\ & + \Psi(x, \dot{\hat{x}})(\dot{\hat{x}} + r) + p(x) + \hat{f}_f \end{array} \tag{6.9}$$

where k is a positive constant gain.

The torque commands for driving the robot can be obtained by:

$$\Gamma = J^T F$$

6.3 Overall System Stability Analysis

We will now present the proof using Lyapunov stability analysis. To determine the stability of the overall closed-loop control system, we define the following Lyapunov function.

$$V = \frac{1}{2} r^T \Lambda(x) r + \frac{1}{2} \tilde{\theta}^T K_{ad}^{-1} \tilde{\theta} \tag{6.10}$$

where $\tilde{\theta}$ is defined as the difference between the actual θ and the estimated $\hat{\theta}$:

$$\tilde{\theta} = \theta - \hat{\theta} \tag{6.11}$$

In order to facilitate the stability analysis, we assume that θ remains unchanged over a certain period of time [1]. Differentiating (6.11) with respect to time to get:

$$\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$$

Differentiating (6.10) with respect to time:

$$\dot{V} = r^T \Lambda(x) \dot{r} + \frac{1}{2} r^T \dot{\Lambda}(x) r + \tilde{\theta}^T K_{ad}^{-1} \dot{\tilde{\theta}} \tag{6.12}$$

From (6.4) we can get:

$$\begin{aligned} \Lambda(x)\dot{r} \quad = \quad & \Lambda(x)[\ddot{x}_d + K_f \dot{e}] + \Psi(x, \dot{\hat{x}})\dot{\hat{x}} \\ & + p(x) + W_j^T(\dot{\hat{q}}) J^{-1} \theta - F \end{aligned} \tag{6.13}$$

Substitute (2.3) into (6.13) to yield:

$$\Lambda(x)\dot{r} = -\Psi(x, \dot{\hat{x}}) r - k\Lambda(x) r + W_j^T(\dot{\hat{q}}) J^{-1} \tilde{\theta} \tag{6.14}$$

Substitute (6.14) into (6.12) and cancel out some terms gives:

$$\dot{V} = -k r^T \Lambda(x) r + \tilde{\theta}^T (W_j^T(\dot{\hat{q}}) J^{-1} r + K_{ad}^{-1} \dot{\tilde{\theta}}) \tag{6.15}$$

[1] This assumption is realistic since friction characteristics are in general constant in longer periods of time compared to the time the adaptation is achieved. Changes in friction characteristics happen over much longer periods of time

Selecting the adaptive update rule as indicated by (6.5), finally we can get:

$$\dot{V} = -kr^T \Lambda(x) r \tag{6.16}$$

From (6.16) it is clear that $\dot{V}(t)$ is a non-positive function; hence, we know that $V(t)$ is either decreasing or constant. Since $V(t)$ is a nonnegative function, we can conclude that $V(t) \in \pounds_\infty$.

Using the fact that $V(t) \in \pounds_\infty$ and that $\Lambda(x)$ is a positive-definite matrix, we can state that $r(t) \in \pounds_\infty$ and $\tilde{\theta}(t) \in \pounds_\infty$.

From the definition of r given in (6.4), we can use the arguments from standard linear control theory to state that e and $\dot{e}$ are bounded; hence, owing to the boundedness of $x_d(t)$ and $\dot{x}_d(t)$, we can conclude that $x(t)$, $\dot{x}(t) \in \pounds_\infty$.

Since $\tilde{\theta}(t) \in \pounds_\infty$ and θ is assumed to be a constant vector over a certain period of time, it is clear that $\hat{\theta}(t) \in \pounds_\infty$.

From the above boundness statements and the fact that $\ddot{x}_d$ is assumed bounded, we can see that $\dot{r}$ is bounded, hence $\ddot{V}$ obtained by differentiating (6.16) is bounded.

Since V is lower bounded, $\dot{V}$ is negative semi-definite, and $\ddot{V}$ is bounded, we can use Barbalat's lemma [71] to state that

$$\lim_{t\to\infty} \dot{V}(t) = 0$$

which means that by the Rayleigh-Ritz Throrem [71]

$$\lim_{t\to\infty} r(t) = 0 \tag{6.17}$$

Note that (6.4) is a stable first-order differential equation driven by the input $r(t)$, therefore, using linear control theory and the result indicated by (6.17), we can conclude that

$$\lim_{t\to\infty} e(t) = 0, \text{ and } \lim_{t\to\infty} \dot{e}(t) = 0$$

This results indicates that the tracking errors $e(t)$ and $\dot{e}(t)$ are global asymptotically stable.

6.4 Experimental Results

The experiments were conducted on PUMA 560 robot, and the sampling time is selected to be 1ms. The defined trajectory is to move the end-effector in XYZ direction with the desired position trajectory indicated by (6.18), while maintaining the initial end-effector orientation constant all the time.

$$\begin{array}{rcl} p_{x_d} & = & p_{x_0} + 50.0\sin(2\pi ft)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{y_d} & = & p_{y_0} + 50.0\cos(2\pi ft)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{z_d} & = & p_{z_0} + 50.0\cos(2\pi ft)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ f & = & 0.2\text{Hz} \end{array} \tag{6.18}$$

where p_{x_0}, p_{y_0} and p_{z_0} are the initial positions of the robot. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

6.4.1 Experimental Result without Friction Adaptation

First, we show the experimental result without friction compensation. The diagonal controller gain K_f in (6.9) was selected as follows:

$$K_f = \text{diag}\{32, 37, 43, 53, 77, 50\} \tag{6.19}$$

k was selected to be 50, and $\hat{f}_f$ in (6.8) was set to zero so that no friction compensation is involved.

Using the trajectory defined by (6.18), the experimental result is shown in Fig. 6.1 and Table 6.1. Where e_x, e_y, and e_z stand for the position tracking errors along X, Y, and Z axis, and e_{ϕ_x}, e_{ϕ_y}, and $e_{\phi_z}$for the orientation tracking errors about X, Y, and Z axis, respectively.

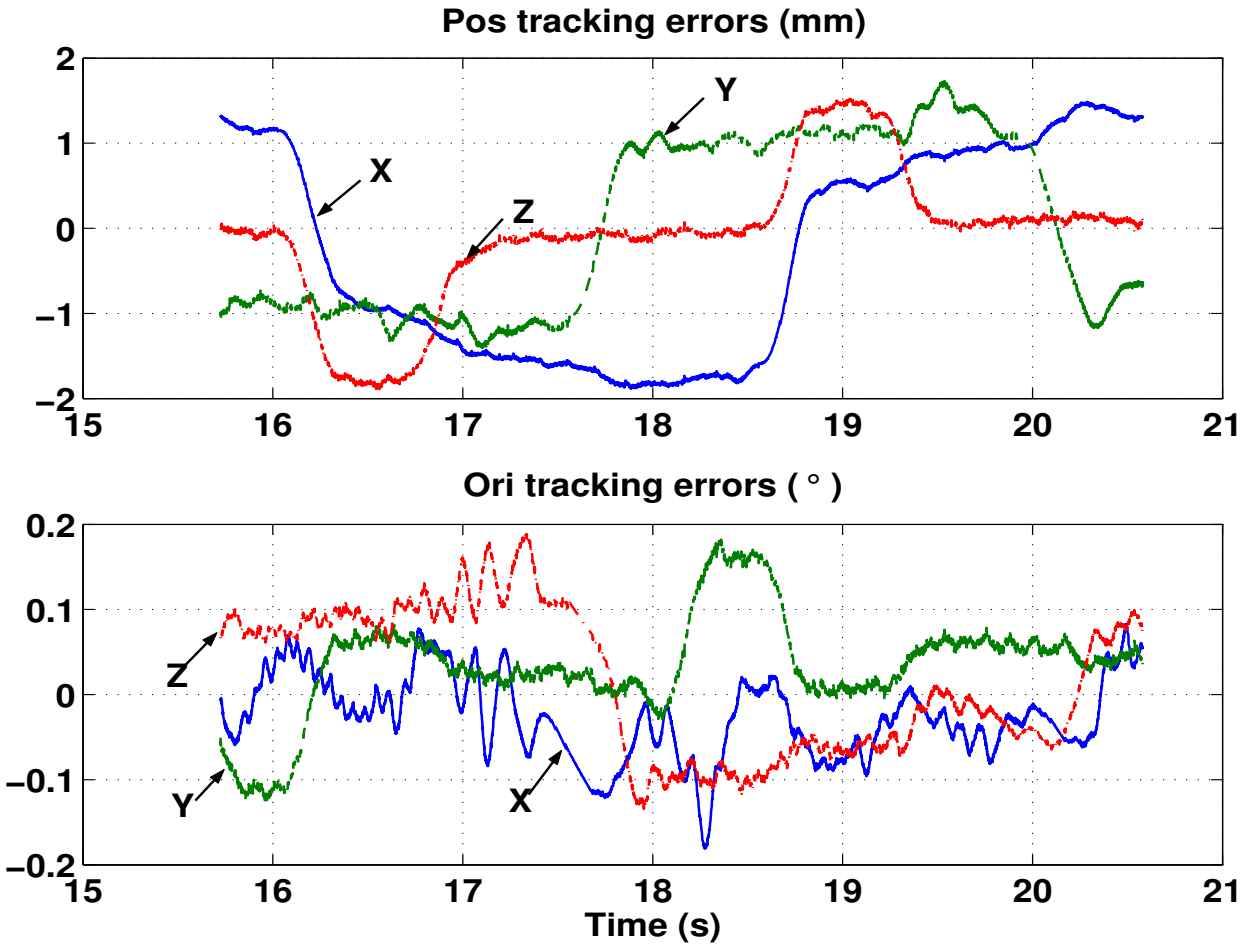

Figure 6.1: Adaptive friction identification and compensation via filtered velocity - Tracking errors without friction compensation

Table 6.1: Adaptive friction identification and compensation via filtered velocity - Tracking errors without friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
1.88mm	1.73mm	1.90mm	0.18^o	0.18^o	0.19^o

Table 6.2: Adaptive friction identification and compensation via filtered velocity - Tracking errors with adaptive friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
0.65mm	0.47mm	0.40mm	0.09^o	0.04^o	0.07^o

Table 6.3: Adaptive friction identification and compensation via filtered velocity - Identified friction coefficients of each joint (J_i)

	J_1	J_2	J_3	J_4	J_5	J_6
τ_{vis_i} (N.m.s/rad)	0.2	0.4	0.6	0.01	0.01	0.01
τ_{cou_i} (N.m)	4.0	5.3	3.0	0.8	0.4	0.5
τ_{sti_i} (N.m)	4.0	5.0	2.0	0.8	0.5	0.6

6.4.2 Experimental Result with Friction Adaptation

The diagonal controller gain K_f was selected to be the same as indicated in (6.19), k was also selected to be 50, all the diagonal terms of τ_{dec} in (6.6) were selected to be 1, and all the initial estimated friction coefficients are set to zeros.

Using the same trajectory defined by (6.18), the experimental result is shown in Figs. 6.2 and 6.3, where tracking errors are shown for the first 10 seconds and after 88 seconds, respectively. It is clear that the position and orientation tracking errors have been greatly reduced because of the introduction of the adaptive friction compensation.

Tables 6.2 and 6.3 show the tracking errors and the identified friction coefficients after the robot ran for about 90 seconds.

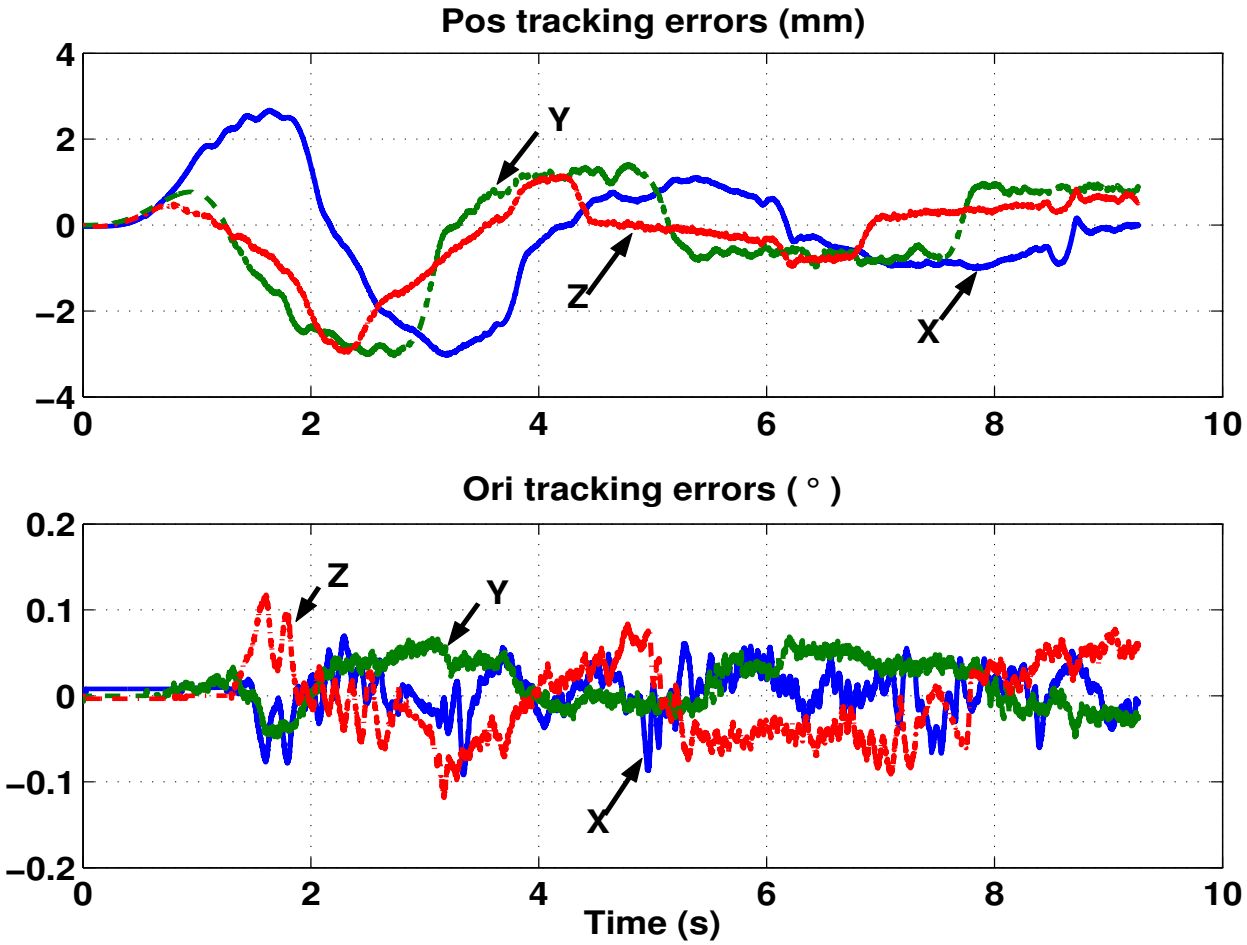

Figure 6.2: Adaptive friction identification and compensation via filtered velocity - Initial tracking errors of the system

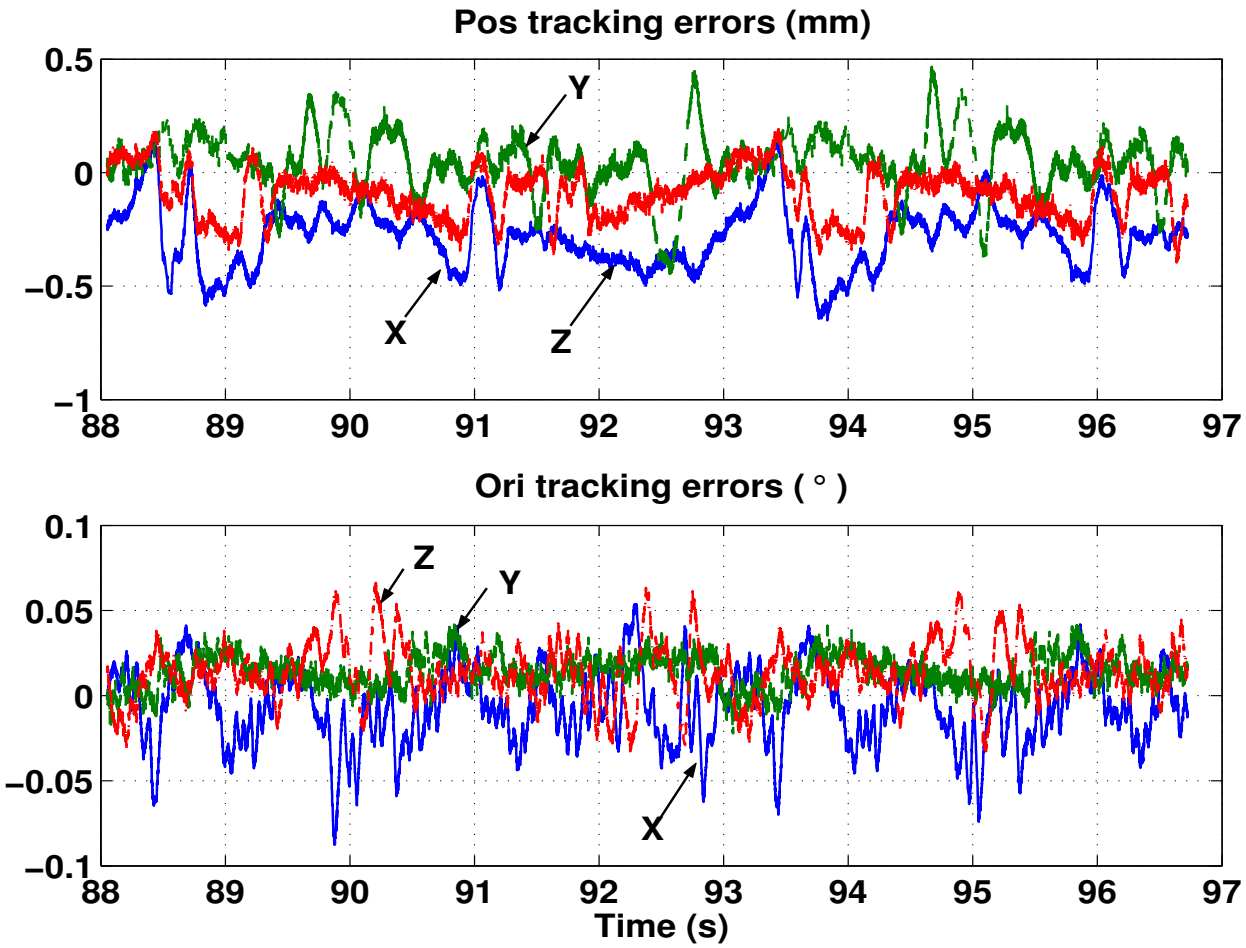

Figure 6.3: Adaptive friction identification and compensation via filtered velocity - Tracking errors with friction compensation

The results indicate that the tracking errors of the controllers with adaptive friction compensation is about 3 to 5 times smaller than the controller without friction compensation.

Remark 11 *It is noted that we do not include stiction term in the friction model, this is because the estimated joint velocities are obtained from backwards difference algorithm, and the chance of obtaining zero velocity is very little. Hence, even if we include the stiction term in the friction adaptation algorithm, it will get little chance to be updated, and the friction identification process will take a long time.*

6.5 Conclusions

In this chapter, we propose an operational space controller with adaptive friction compensation capability to achieve better tracking performance. The errors between actual joint frictions and estimated joint frictions are bounded. Experimental results using PUMA 560 indicate that, the proposed adaptive controller is able to achieve much higher tracking accuracy that the one without friction compensation. The result of the controller has been published in [74].

The improvement of this controller over the one used for SISO system presented in [60] are: first, the new controller is designed to be used for n degree-of-freedom robot manipulators; second, the controller has been redeveloped for operational space instead of joint space control; and third, two additional terms have been added to take into consideration of the centrifugal and Coriolis force, and the gravitational effect of a robot manipulator.

Compared with the adaptive friction compensation controller presented in Chapter 5, the tracking errors of the controller proposed in this chapter is slightly larger. The

reason being that the filtered velocity is less accurate than the observed velocity, as indicated in Chapters 3 and 4. However, due to its global stability nature, the gains selection of the adaptive controller using filtered velocity is easier. In addition, its controller is simpler than the one introduced in Chapter 5.

CHAPTER 7

CONTROL ALGORITHM 5: ADAPTIVE FRICTION IDENTIFICATION AND COMPENSATION USING BOTH OBSERVED AND DESIRED VELOCITY

7.1 Introduction

Based on the adaptive observer-controller presented in Chapter 6, we developed a new adaptive controller in this chapter. Since desired velocity is not contaminated by noise, the friction adaptation will also become smoother if we use desired instead of filtered velocity information in our friction adaptation law. Based on this motivation, our proposed friction adaptation law is formulated by utilizing both observed and desired velocity information. Without velocity measurements, the overall observer-controller system can achieve a semi-global asymptotic stability for the position and velocity tracking errors, and position and velocity estimation errors, with estimated friction coefficients converging asymptotically. Experimental results indicate that the proposed adaptive observer-controller is able to achieve much higher tracking accuracy than the observer-controller without friction compensation. It can also achieve higher tracking accuracy than the adaptive controller using filtered velocity, which verify the effectiveness of the controller proposed in this chapter.

7.2 Adaptive Observer-Controller Formulation

Similar to the adaptive controller structure in Chapter 5, our proposed adaptive observer-controller in this chapter also consists of a model-based velocity observer, a controller that is formulated in operational space, plus friction adaptation law.

7.2.1 Formulation of Robust Velocity Observer

The velocity observer used in this chapter has the same structure as the one in Chapter 5. The equations are repeated and assigned new equation numbers for expository convenience:

$$\dot{\hat{x}} = y + k\tilde{x} \tag{7.1}$$

$$\dot{y} = \Lambda(x)^{-1}\Big[F - \Psi(x, \dot{\hat{x}})\dot{\hat{x}} - p(x) - \hat{f} + k_i\tilde{x}\Big] \tag{7.2}$$

where

$$\tilde{x} = x - \hat{x} \tag{7.3}$$

where $\hat{f}$ has been defined in (5.20). $\dot{\hat{x}}$, y, and F defined in Chapter 3, and k_i defined in Chapter 4.

k is a positive scalar constant defined by:

$$k = \frac{1}{m_1}\left[\zeta_c\zeta_d + \zeta_c k_0 + \zeta_c k_s k_0 + k_s + 2k_n + k_s^2\zeta_f^2 + \zeta_f^2 + \zeta_f\right] \tag{7.4}$$

where k_0, k_s and k_n, ζ_c, and ζ_d were defined in Chapter 3. ζ_e is defined in (5.9), ζ_f defined in (7.7).

7.2.2 Formulation of Friction Adaptation Law

We still use the same formula for joint friction as defined by (5.4).

We assume that the viscous frictions have the form:

$$\tau_{vis}\dot{q} = \text{vec}\{\tau_{vis_i}\dot{q}_i\} \tag{7.5}$$

with τ_{vis_i} constant coefficients. Then τ_{vis} =diag$\{\tau_{vis_i}\}$, a diagonal matrix with entries τ_{vis_i}.

The corresponding operational space viscous friction f_{vis} can be obtained by:

$$f_{vis} = J^{-T}\tau_{vis} \tag{7.6}$$

The operational space viscous friction term f_{vis} has the following property [71]:

$$\|f_{vis}\dot{x}\| \leq \zeta_f \|\dot{x}\| \tag{7.7}$$

We will use this property for our controller development.

θ is the $3n \times 1$ vector of constant parameters defined in Chapter 5 and repeated here:

$$\theta = \begin{bmatrix} \text{vec}^T\{\tau_{vis_i}\} & \text{vec}^T\{\tau_{cou_i}\} & \text{vec}^T\{\tau_{sti_i}\} \end{bmatrix}^T \tag{7.8}$$

The friction parameter estimate vector $\hat{\theta}$ is updated using the following adaptation algorithm:

$$\dot{\hat{\theta}} = -K_{ad}W_j(\dot{\hat{q}}, \dot{\hat{q}}_d)\tilde{\dot{q}} \tag{7.9}$$

where K_{ad} and $\tilde{\dot{q}}$ were defined in Chapter 5, and $W_j(\dot{\hat{q}}, \dot{\hat{q}}_d)$ is the $n \times 3n$ regression vector given by:

$$\begin{array}{lcl} W_j(\dot{\hat{q}}, \dot{\hat{q}}_d) & = & \begin{bmatrix} \hat{w}_{j1}(\dot{\hat{q}}_d) & \hat{w}_{j2}(\dot{\hat{q}}) & \hat{w}_{j3}(\dot{\hat{q}}) \end{bmatrix} \\ \hat{w}_{j1}(\dot{\hat{q}}_d) & = & \text{diag}(\dot{\hat{q}}_d) \\ \hat{w}_{j2}(\dot{\hat{q}}) & = & \text{diag}(\text{sgn}(\dot{\hat{q}})) \\ \hat{w}_{j3}(\dot{\hat{q}}) & = & \text{diag}(\text{sgn}(\dot{\hat{q}})\exp(-\tau_{dec}\dot{\hat{q}}^2)) \end{array} \tag{7.10}$$

From (5.15) in Chapter 5, we know that $\dot{\hat{\theta}}$ is the function of $\dot{\hat{q}}$. If $\dot{\hat{q}}$ is contaminated with lesser noise, $\dot{\hat{\theta}}$ will become smoother. Since we can define a motion trajectory

that is smooth, there will be no noise component in desired velocity. To make use of the merit of no-noise desired velocity, we develop a friction adaptation law as shown by Equ 7.9, with the regression vector the function of both observed and desired velocity.

The estimated joint frictions are then obtained by:

$$\hat{\tau}_f = W_j(\dot{\hat{q}}, \dot{\hat{q}}_d)\hat{\theta} \tag{7.11}$$

From (7.11), the estimated frictions in operational space can be obtained by:

$$\hat{f} = J^{-T}\hat{\tau}_f \tag{7.12}$$

7.2.3 Formulation of Operational Space Controller

By using the estimated velocity $\dot{\hat{x}}$ proposed in Section 7.2.1, the following model-based controller is formulated to generate the required driving force:

$$F = (k_s + k_{nd})\eta_p + w_e - k_i\tilde{x} \tag{7.13}$$

where k_{nd} is a positive controller gain defined as:

$$k_{nd} = 2k_n + \zeta_c k_0 + (k_s m_2 + k m_2)^2 k_n \tag{7.14}$$

The $n \times 1$ observed filtered tracking error signal η_p is defined as:

$$\eta_p = \dot{x}_d + k_s e - \dot{\hat{x}} \tag{7.15}$$

and the $n \times 1$ auxiliary vector w_e is defined as:

$$w_e = \Lambda(x)[\ddot{x}_d + k_s(\dot{x}_d - \dot{\hat{x}})] + \Psi(x, \dot{\hat{x}})(\dot{x}_d + k_s e) + p(x) + \hat{f} \tag{7.16}$$

where the $n \times 1$ end-effector position and orientation tracking error e is defined as:

$$e = x_d - x \tag{7.17}$$

The force command F will be used in the observer indicated by (7.2). And the torque commands for driving the robot can be obtained by:

$$\Gamma = J^T(q)F$$

7.3 Overall System Stability Result and Analysis

Theorem 4 *Under the assumption that the exact model of a robot except friction is known, if the observer-controller gains satisfy the following sufficient conditions :*

$$\begin{array}{rcl} k_s & > & 1/k_n + \eta_e \\ k_0 & > & \|err(0)\| \end{array} \tag{7.18}$$

the closed-loop tracking error system is stable and the errors $e(t)$, $\dot{e}(t)$, $\tilde{x}(t)$, $\dot{\tilde{x}}(t)$, *and* $(\hat{\theta}(t) - \theta)$ *are bounded where*

$$err = \begin{bmatrix} \eta_p^T & e^T & \dot{\tilde{x}}^T & \tilde{x}^T \end{bmatrix}^T \in \Re^{4n} \tag{7.19}$$

We will now present the stability proof using Lyapunov stability analysis. To determine the stability of the overall closed-loop control system, we define the following Lyapunov function:

$$V = V_0 + V_1 + V_2 \tag{7.20}$$

where the three sub-Lyapunov functions V_0, V_1, and V_2 are defined as:

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T k_i \tilde{x} + \frac{1}{2}\tilde{\theta}^T K_{ad}^{-1}\tilde{\theta} \tag{7.21}$$

where $\tilde{\theta}$ is the difference between the actual θ and the estimated $\hat{\theta}$, and the velocity observation error $\dot{\tilde{x}}$ is defined as the difference between the actual end-effector velocity $\dot{x}$ and the observed end-effector velocity $\dot{\hat{x}}$, it is obtained by differentiating (7.3) with respect to time:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \tag{7.22}$$

$$V_1 = \frac{1}{2} e^T e \tag{7.23}$$

$$V_2 = \frac{1}{2} \eta_p^T \Lambda(x) \eta_p \tag{7.24}$$

Hence, $\dot{V}$ can be obtained by:

$$\dot{V} = \dot{V}_0 + \dot{V}_1 + \dot{V}_2 \tag{7.25}$$

We will formulate the bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$ separately, and then combine them together to get the bound of $\dot{V}$.

7.3.1 Lyapunov Function for Observation Error $\tilde{x}$, $\dot{\tilde{x}}$ and $\tilde{\theta}$

V_0 is defined in (7.21). To form the bound of $\dot{V}_0$, first, take the time derivative of (7.1) and then substitute (7.2) into the resulting expression to yield:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} + p(x) + \hat{f} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} = F \tag{7.26}$$

Subtract (7.26) from (2.7), use (A.3) and (7.22) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{x})\dot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} + k_i\tilde{x} + f - \hat{f} = 0 \tag{7.27}$$

Differentiate V_0 along (7.27) to get:

$$\begin{aligned} \dot{V}_0 = \; & \dot{\tilde{x}}^T \left[-\Psi(x, \dot{x})\dot{\tilde{x}} - \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} \right] \\ & + \frac{1}{2}\dot{\tilde{x}}^T \dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i \tilde{x} - \dot{\tilde{x}}^T (f - \hat{f}) \end{aligned} \tag{7.28}$$

Utilizing (A.2) to get:

$$\dot{V}_0 = -\dot{\tilde{x}} \left[\Psi(x, \dot{\hat{x}}) + k\Lambda(x) \right] \dot{\tilde{x}} - \dot{\tilde{x}}^T (f - \hat{f}) \tag{7.29}$$

Then utilize (A.1), (A.4), and (3.2) to get the upper bound of $\dot{V}_0$:

$$\begin{aligned} \dot{V}_0 \leq \; & \left(\zeta_c \zeta_d + \zeta_c \|\eta_p\| + \zeta_c k_s \|e\| - k m_1 \right) \|\dot{\tilde{x}}\|^2 \\ & + \zeta_f k_s \|\dot{\tilde{x}}\| \|e\| + \zeta_f \|\dot{\tilde{x}}\| \|\eta_p\| + \zeta_f \|\dot{\tilde{x}}\|^2 \end{aligned} \tag{7.30}$$

7.3.2 Lyapunov Function for Tracking Error e

The evolution of the position tracking error with time can be derived by differentiating (7.17) with respect to time to yield:

$$\dot{e} = \dot{x}_d - \dot{x}$$

Since $\dot{x}$ is not measurable, we use the estimated term $\dot{\hat{x}}$ to eliminate $\dot{x}$ and get the following equation:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} - \dot{\tilde{x}} \tag{7.31}$$

Simplify (7.31) by utilizing (7.15) to get:

$$\dot{e} = -k_s e + \eta_p - \dot{\tilde{x}} \tag{7.32}$$

V_1 is defined in (7.23), and the upper bound for the time derivative of V_1 along (7.32) is given by:

$$\dot{V}_1 \leq -k_s \|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{7.33}$$

7.3.3 Lyapunov Function for η_p

The evolution of η_p with time can be derived by differentiating (7.15) with respect to time, multiplying both sides of the resulting expression by $\Lambda(x)$, and substituting the expression from (7.26) for $\ddot{\hat{x}}$ to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= \Lambda(x)\ddot{x}_d + k_s\Lambda(x)(\dot{x}_d - \dot{x}) - k\Lambda(x)\dot{\tilde{x}} \\ &\quad -k_i\tilde{x} + \Psi(x, \dot{\hat{x}})\dot{\hat{x}} + p(x) + \hat{f} - F \end{aligned} \tag{7.34}$$

Substitute the force input given by (7.13) into (7.34), use the definitions of w_e and η_p to get:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -(k_s + k_{nd})\eta_p - (k + k_s)\Lambda(x)\dot{\tilde{x}} \\ &\quad -\Psi(x, \dot{\hat{x}})\eta_p \end{aligned} \tag{7.35}$$

Rewrite the term $\Psi(x, \dot{\hat{x}})\eta_p$ on the right-hand side of (7.35) in terms of $\dot{\tilde{x}}$, and utilize (A.3) and (7.22) to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -\Psi(x, \dot{x})\eta_p - (k_s + k_{nd})\eta_p \\ &\quad -(k + k_s)\Lambda(x)\dot{\tilde{x}} + \Psi(x, \dot{\tilde{x}})\eta_p \end{aligned} \tag{7.36}$$

V_2 is defined in (7.24). Differentiating V_2 along (7.36) and utilizing (A.2) yields:

$$\begin{aligned} \dot{V}_2 &= -(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &\quad +\eta_p^T\Psi(x, \dot{\tilde{x}})\eta_p \end{aligned} \tag{7.37}$$

From (7.37), and using (A.1) and (A.4), we can obtain the following upper bound for $\dot{V}_2$:

$$\begin{aligned} \dot{V}_2 &\leq -(k_s + k_{nd})\|\eta_p\|^2 + (k + k_s)m_2\|\eta_p\|\|\dot{\tilde{x}}\| \\ &\quad +\zeta_c\|\eta_p\|^2\|\dot{\tilde{x}}\| \end{aligned} \tag{7.38}$$

7.3.4 Overall System Stability Analysis

Use the upper bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$, the upper bound of $\dot{V}$ can be formed as:

$$\begin{aligned} \dot{V} \leq\ & -(k_s - \tfrac{1}{k_n} - 1)\|e\|^2 - (k_s - 1)\left\|\eta_p\right\|^2 - (k_s - \tfrac{1}{k_n})\left\|\dot{\tilde{x}}\right\|^2 \\ & -(k_o - \|err\|)\left(\zeta_c\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s\left\|\dot{\tilde{x}}\right\|^2 + \zeta_c\left\|\eta_p\right\|^2\right) \end{aligned} \tag{7.39}$$

From (7.39) we can see that, if $k_s > 1 + 1/k_n$ and $k_0 \geq ||err||$, we can get:

$$\dot{V} \leq 0 \tag{7.40}$$

From (7.40), we can get the conclusion that, friction coefficients estimation error $\tilde{\theta}$, position tracking errors e, position estimation errors $\widetilde{x}$, velocity estimation errors $\dot{\widetilde{x}}$, observed filtered tracking error signal η_p of the observer-controller are all bounded. Furthermore, the end-effector velocity tracking error is also bounded. In fact, after adding and subtracting $\dot{x}$ from the right-hand side of (7.15) and rearranging the terms, we can formulate the following inequality:

$$\|\dot{e}\| = \|\dot{x}_d - \dot{x}\| \leq \left\|\eta_p\right\| + k_s \|e\| + \|\dot{x}\| \tag{7.41}$$

Since each of the terms on the right-hand side of the above equation is bounded, $\|\dot{e}\|$ is also bounded. This yields the result indicated by *Theorem 4*.

7.4 Implementation of Friction Adaptation Law

Like what we have done in Chapter 5, we follow the same procedure to obtain our new friction adaptation algorithm. Integrating both sides of (7.9) giving:

$$\hat{\theta}(t) = \hat{\theta}(t - \Delta t) - K_{ad} \int_{t-\Delta t}^{t} W_j^T(\dot{\hat{q}}, \dot{\hat{q}}_d)(\frac{dq}{dt} - \dot{\hat{q}})dt \tag{7.42}$$

where Δt represents the sampling time of the system.

Assuming that Δt is small enough, from (7.42) we can get the following form:

$$\begin{array}{rcl} \hat{\theta}(t) & = & \hat{\theta}(t - \Delta t) - K_{ad} W_j^T(\dot{\hat{q}}(t - \Delta t), \dot{\hat{q}}_d(t - \Delta t)) D_q \\ D_q & = & q(t) - q(t - \Delta t) - \dot{\hat{q}}(t - \Delta t)\Delta t \end{array} \tag{7.43}$$

We will use (7.43) as our friction adaptation algorithm.

7.5 Experimental Results

The experiments were performed using PUMA 560 robot, and the sampling time is selected to be 1ms. The defined trajectory is to move the end-effector in XYZ direction with the desired position trajectory indicated by (7.44), while maintaining the initial end-effector orientation constant all the time.

$$\begin{aligned} p_{x_d} &= p_{x_0} + 50.0\sin(0.4\pi t)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{y_d} &= p_{y_0} + 50.0\cos(0.4\pi t)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{z_d} &= p_{z_0} + 50.0\cos(0.4\pi t)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ x_d &= \begin{bmatrix} p_{x_d} & p_{y_d} & p_{z_d} \end{bmatrix}^T \end{aligned} \tag{7.44}$$

where $(p_{x_0}, p_{y_0}, p_{z_0})$ is the initial position vector of the robot. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

The controller gains were selected as diagonal gains matrices as following:

$$\begin{aligned} k_{nd} &= \text{diag}\{120, 120, 120, 35, 35, 35\} \\ k &= \text{diag}\{108, 108, 108, 32, 32, 32\} \\ k_s &= \text{diag}\{97, 97, 97, 30, 30, 30\} \\ k_i &= \text{diag}\{2000, 2000, 2000, 3000, 3000, 3000\} \end{aligned} \tag{7.45}$$

All the diagonal terms of the 18×18 friction adaptation gains K_{ad} were selected to be 100, all the diagonal terms of τ_{dec} were selected to be 1, and all the initial estimated friction coefficients are set to zeros.

7.5.1 Friction Identification and Compensation Performance

Using the trajectory defined by (7.44), the experimental result is shown in Figs. 7.1, 7.3, 7.2, and 7.4. Figs. 7.1 and 7.2 show the initial tracking errors and the identified friction coefficients when the robot just starts to move, and Figs. 7.3 and 7.4 shows the parameters when friction adaptation algorithm is activated for about two minutes, respectively. Where J_i stands for Joint i, and e_x, e_y, and e_z

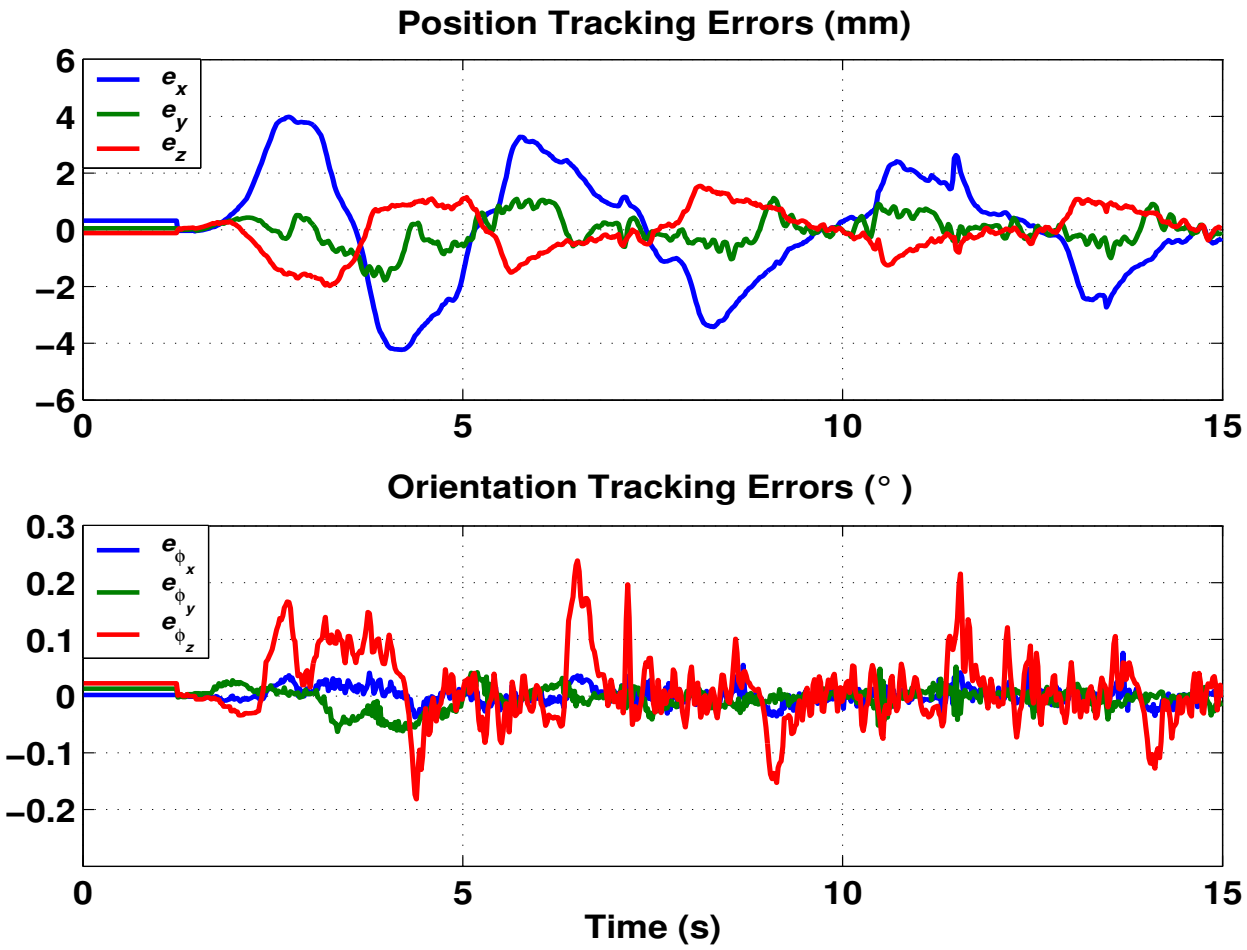

Figure 7.1: Adaptive friction identification and compensation using both observed and desired velocity - Initial tracking errors with adaptive friction compensation

are the position tracking errors along X, Y, and Z axis, and e_{ϕ_x}, e_{ϕ_y}, and $e_{\phi_z}$are the orientation tracking errors about X, Y, and Z axis, respectively.

Tables 7.1 and 7.2 show the tracking errors and the identified friction coefficients after the robot ran for about two minutes.

Under the same conditions, using the same controllers gains listed in (7.45) but without friction compensation, the result is shown in Fig. 7.5 and Table 7.3.

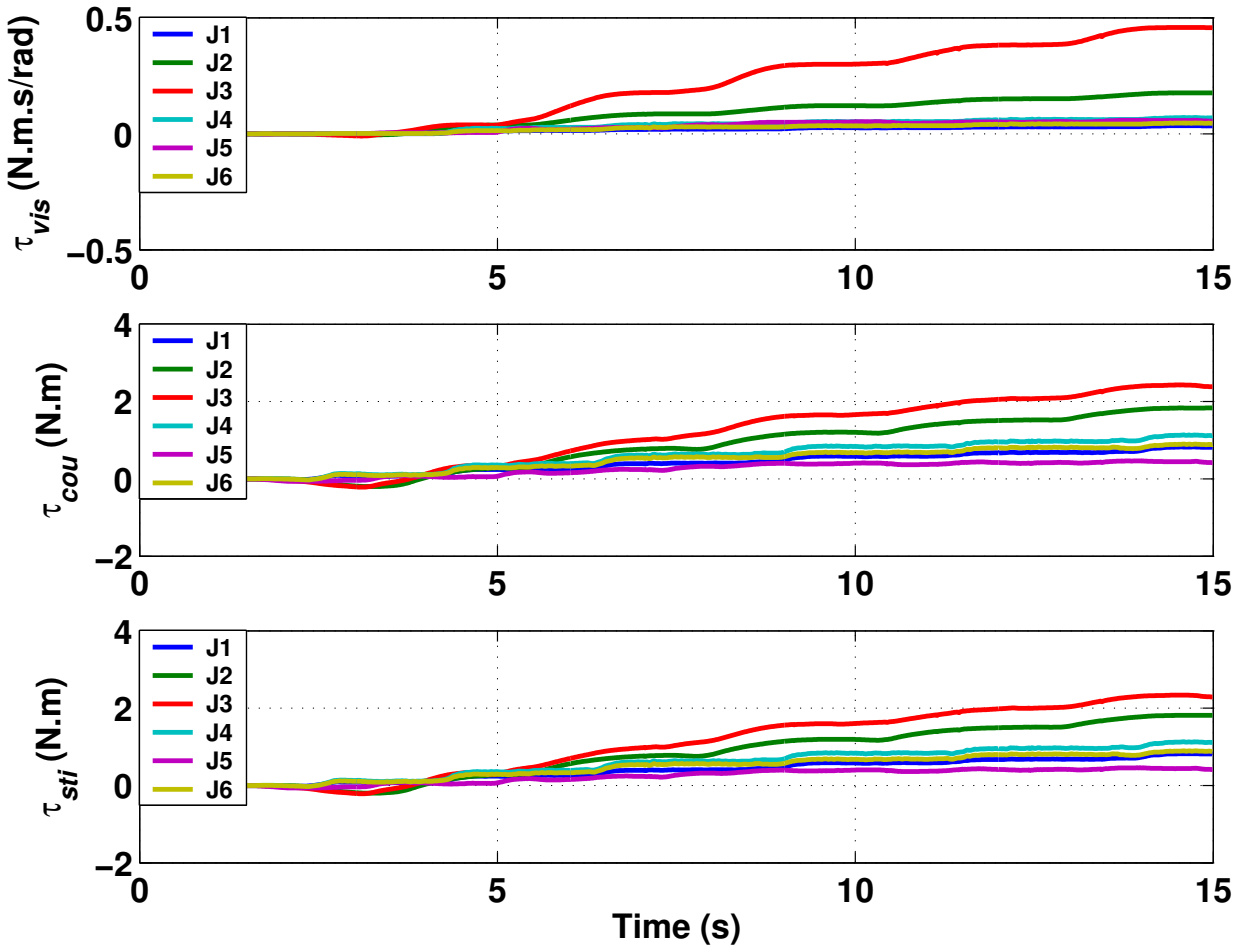

Figure 7.2: Adaptive friction identification and compensation using both observed and desired velocity - Initial identified joints friction coefficients

Table 7.1: Adaptive friction identification and compensation using both observed and desired velocity - Maximum tracking errors with adaptive friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
0.45mm	0.62mm	0.37mm	0.04^o	0.04^o	0.11^o

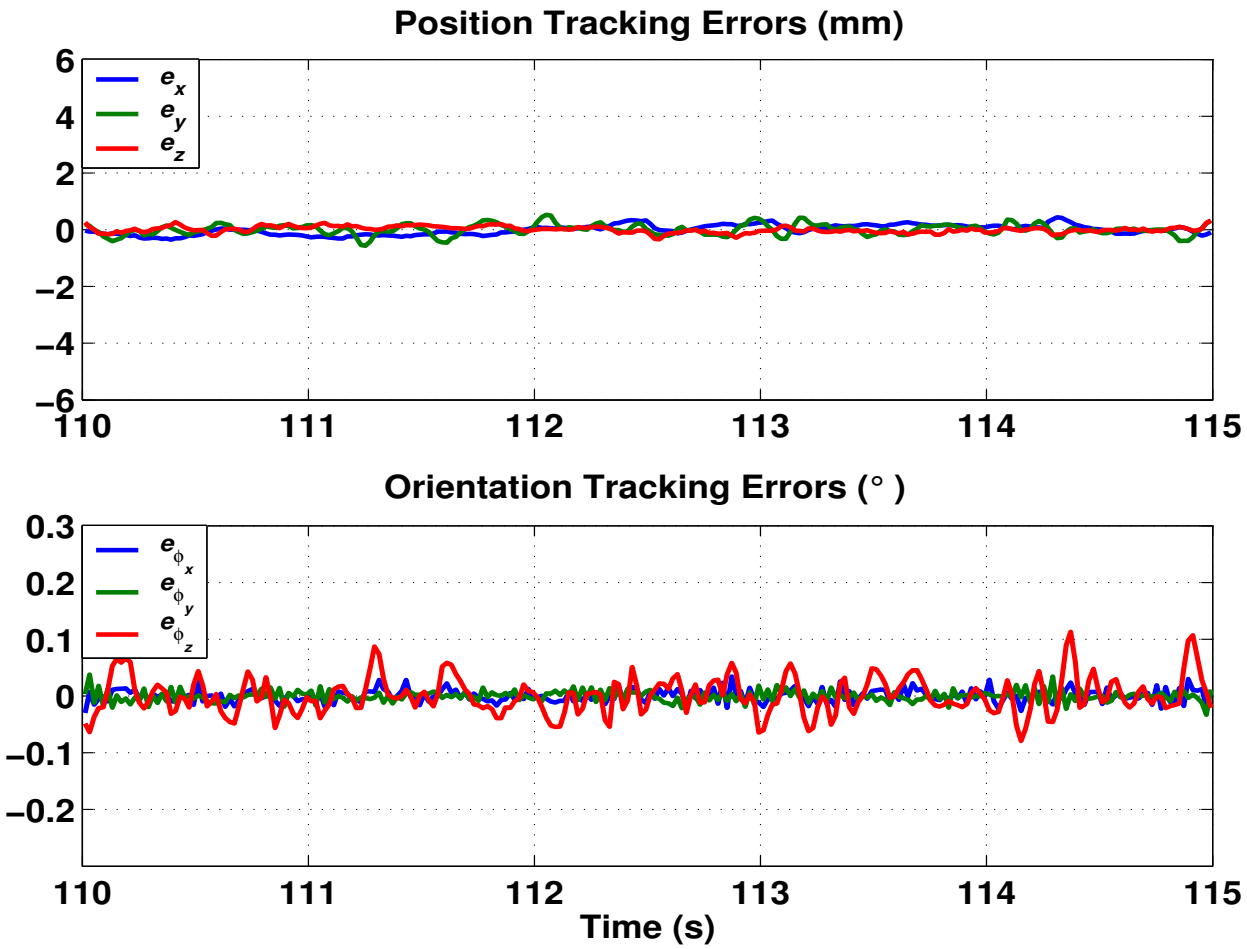

Figure 7.3: Adaptive friction identification and compensation using both observed and desired velocity - Tracking errors with adaptive friction compensation

Table 7.2: Adaptive friction identification and compensation using both observed and desired velocity - Identified friction coefficients of each joint

	J_1	J_2	J_3	J_4	J_5	J_6
τ_{vis_i} (N.m.s/rad)	0.2	0.5	1.3	0.3	0.1	0.1
τ_{cou_i} (N.m)	3.5	4.8	2.3	0.8	0.2	0.5
τ_{sti_i} (N.m)	3.5	4.7	2.0	0.7	0.2	0.5

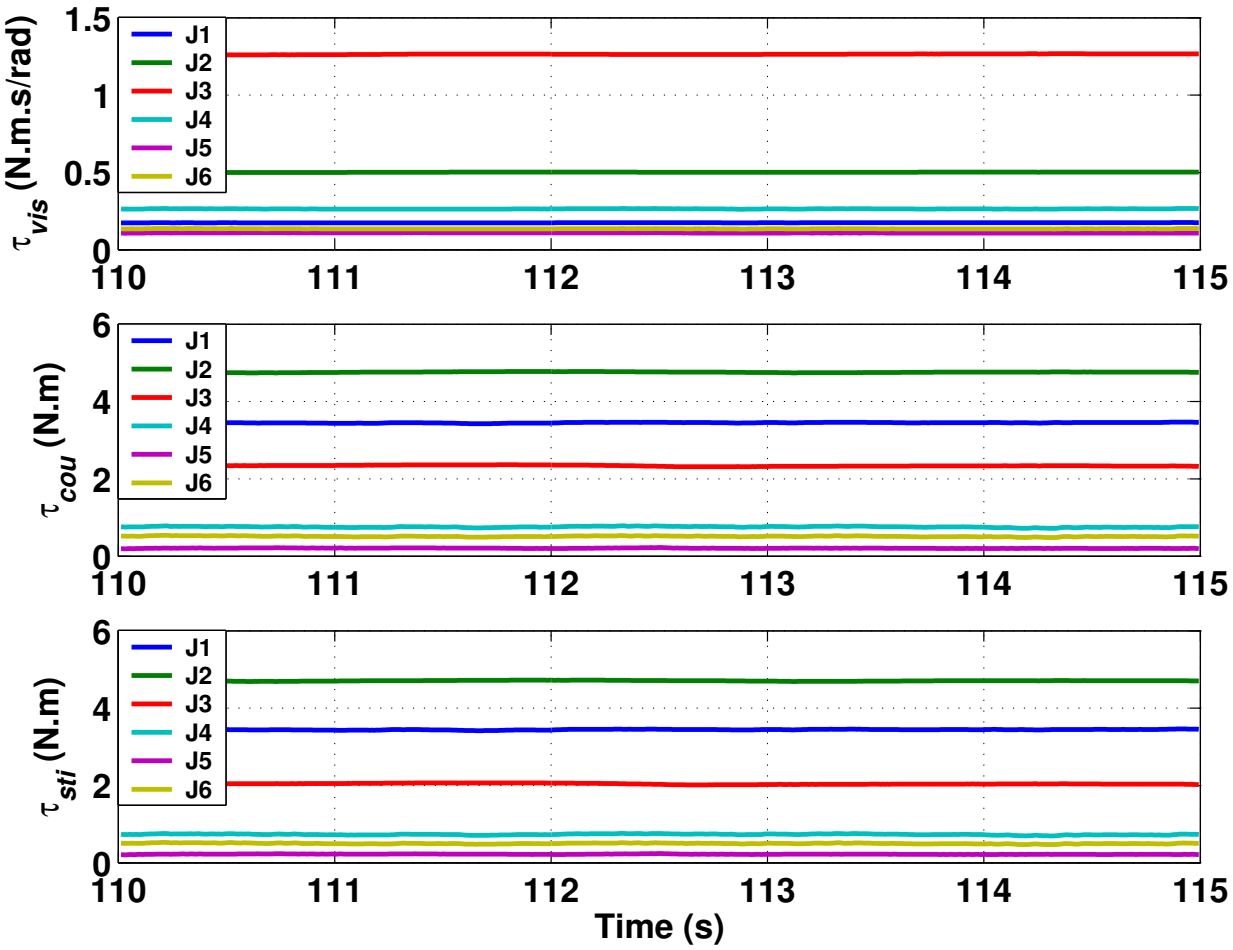

Figure 7.4: Adaptive friction identification and compensation using both observed and desired velocity - Final identified joints friction coefficients

Table 7.3: Adaptive friction identification and compensation using both observed and desired velocity - Maximum tracking errors without friction compensation

e_x	e_y	e_z	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
2.14mm	1.59mm	1.51mm	0.08^o	0.07^o	0.14^o

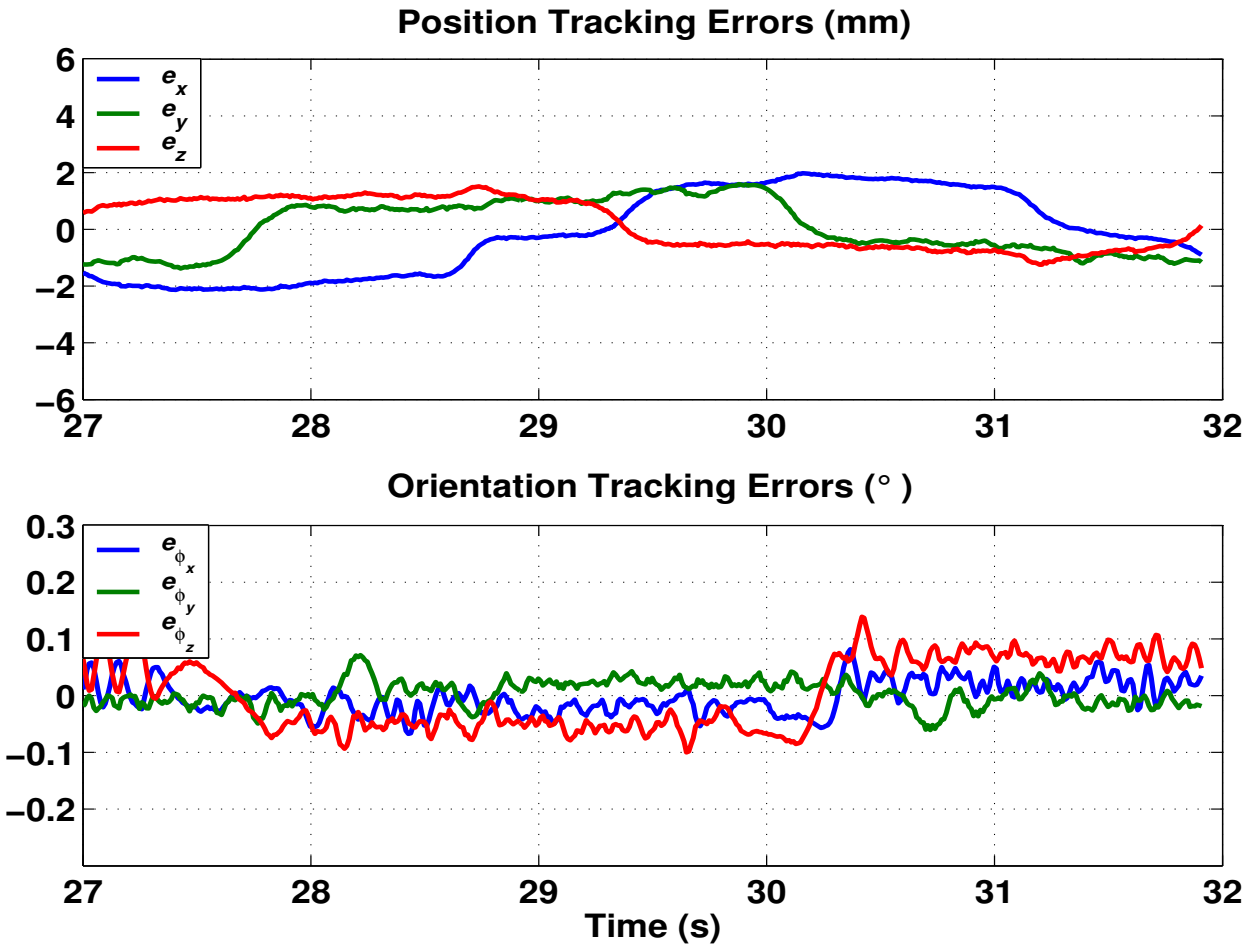

Figure 7.5: Adaptive friction identification and compensation using both observed and desired velocity - Tracking errors without friction compensation

The results indicate that the tracking errors of the controllers with adaptive friction compensation is about 1.3 to 4.7 times smaller than the controller without friction compensation, which verify the effectiveness of the proposed adaptive controller.

Remark 12 *The joint friction coefficients identification rate is depended on trajectories, but eventually the identified coefficients are more or less the same.*

Remark 13 *When a robot comes to a standstill, ideally,* $q(t)-q(t-\Delta t)$ *should equal zero. However, due to sensor noise, it seldom equals zero. From (7.43) we know that* $\hat{\theta}(t)$ *will be updated by* $q(t)-q(t-\Delta t)$ *even if none of the robot joints is moving, which will cause wrong identification. In order to eliminate the problem, we can temporarily stop friction adaptation and identification when a robot is close to a standstill.*

Remark 14 *From Tables 5.2 and 7.2, we can see that the identified friction coefficients are different. The reasons being that the errors between the actual joint frictions and the estimated joint frictions are bounded, and the controller can not guarantee that the estimated joint frictions will eventually approach the actual joint frictions.*

7.6 Conclusions

In this chapter, we proposed an operational space observer-controller with adaptive friction compensation capability. The friction adaptation algorithm is designed to make use of the merits of the smooth desired velocity and "cleaner" observed velocity information to achieve better performance. For friction compensation, some researchers use desired velocity to replace the noisy filtered velocity information in their friction adaptation algorithms, but are unable to provide stability proof. In this chapter, we have incorporated desired velocity information into our adaptive friction

compensation law, the overall stability result is also given to prove the feasibility of the proposed algorithm. The result of the controller has been published in [75].

Compared with the adaptive friction compensation controller presented in Chapter 6, the tracking errors of the controller proposed in this chapter is smaller. The reason being that the filtered velocity is less accurate than the observed velocity, as explained in Chapters 3 and 4. However, the controller in this chapter is a semi-global asymptotic stable controller, while the one in Chapter 6 is a global stable controller. Besides, the controller structure in Chapter 6 is simpler than the one introduced in this chapter, and gains selection is also easier.

Compared with the results shown in Figs. 5.1 and 7.1, it is clear that, by adopting both desired and observed velocity information in the friction adaptation law, the friction adaptation process is smoother than that using observed velocity information only.

The main contributions of this controller is that, instead of using observed velocity information only (the controller in Chapter 5), the proposed controller here used both desired and observed velocity information for friction adaptation.

Both the controllers in Chapter 5 and this chapter can achieve similar tracking accuracy.

Similar to the adaptive controllers in Chapters 5 and 6, when a robot model is not accurate enough, the accuracy of friction identification will be affected.

CHAPTER 8

CONTROL ALGORITHM 6: PARALLEL FORCE AND MOTION CONTROL USING OBSERVED VELOCITY

8.1 Introduction

In Chapter 4, we presented a robust observer controller that was designed to make use of the merits of "cleaner" observed velocity. For force control, the operational space formulation provides a framework for the analysis and control of robotic systems with respect to interactions with their environments. Combining our proposed robust observer-controller and force control algorithm, we present in this chapter a parallel force and motion controller to achieve force and position tracking. Impact control algorithm has also been implemented to remove oscillation when the end-effector comes into contact with stiff environment.

8.2 Parallel Force and Motion Control

Our proposed parallel force and motion controller consists of a motion controller and a force controller. The motion controller is designed to use observed velocity, while the force controller is a PD controller. To begin with, task must be defined: which degrees of freedom are assigned to force control and which to motion control. Then appropriate control algorithms are applied respectively.

The resulting force and motion control is done by selecting the desired force or motion response of the robot and adding them together to get the effective robot response. This is expressed as,

$$f = f_{motion} + f_{force} \tag{8.1}$$

where

$$f_{motion} = \Omega f^*_{motion} \tag{8.2}$$

where f^*_{motion} is the force control input to the proposed observer-controller.

and

$$f_{force} = \hat{\Lambda}(x)\bar{\Omega} f^*_{force} + \bar{\Omega} f_{sensor} \tag{8.3}$$

Ω and $\bar{\Omega}$ are the selection matrices to switch the application between force or motion whichever is desired and to specify the direction of application. f^*_{motion} takes into account robot dynamics and computes the required force for motion control (details are in Section 8.2.2). f^*_{force} represent the desired end-effector accelarations in force control subspace. The Coriolis and gravity vectors $\hat{\Psi}(x, \dot{x})\dot{x}$ and $\hat{p}(x)$ only need to be compensated once. Since the two terms are included in f^*_{motion} command, there is no need to put them in f_{force} command one more time.

The force control algorithm can be selected as:

$$f^*_{force} \;=\; K_{pf} e_{force} + K_{if} \int e_{force} dt \tag{8.4}$$

where e_{force} is defined as

$$e_{force} = f_d - f_{contact} \tag{8.5}$$

The formulation of unified force and motion control has been explained in Chapter 2. Fig 2.2 illustrated the tool frame assignment.

The generalized task specification matrices Ω can be defined as:

$$\Omega = \begin{pmatrix} {}^{T}R_{P}^{T} \sum_{F} {}^{T}R_{P} & 0 \\ 0 & {}^{T}R_{P}^{T} \sum_{M} {}^{T}R_{P} \end{pmatrix} \tag{8.6}$$

where

$$\sum_{F} = \begin{pmatrix} \sigma_{FX} & 0 & 0 \\ 0 & \sigma_{FY} & 0 \\ 0 & 0 & \sigma_{FZ} \end{pmatrix} \tag{8.7}$$

$$\sum_{M} = \begin{pmatrix} \sigma_{MX} & 0 & 0 \\ 0 & \sigma_{MY} & 0 \\ 0 & 0 & \sigma_{MZ} \end{pmatrix} \tag{8.8}$$

and

σ_{FX}, σ_{FY}, σ_{FZ}, σ_{MX}, σ_{MY}, σ_{MZ} are positive values between 0 and 1, and "1" signifies application of purely motion control along the corresponding axis, and "0" purely force control.

Based on Khatib's unified force and motion control, we modified the task specification matrix so that it can be used for our proposed parallel force and motion controller. For this kind of controller, the value of "σ" along a force control axis is very small, this is to guarantee that force control has a stronger control over motion control along the axis. The weak motion control along force control subspace is applied to improve system stability and force control performance.

For force control, since the desired force and sensor readings are expressed in Frame{T}, they do not need to perform transformation before the application of $\bar{\sum}_{F}$ and $\bar{\sum}_{M}$, which are the complements of $\sum_{F}$ and $\sum_{M}$. Hence $\bar{\Omega}$ is obtained by:

$$\bar{\Omega} = \begin{pmatrix} {}^{T}R_{P}^{T}\bar{\sum}_{F} & 0 \\ 0 & {}^{T}R_{P}^{T}\bar{\sum}_{M} \end{pmatrix} \tag{8.9}$$

8.2.1 Formulation of Robust Velocity Observer

As with before, to estimate the end-effector velocity, we utilize the following second order velocity observer:

$$\dot{\hat{x}} = y + k\tilde{x}, y(0) = -k\tilde{x}(0) \tag{8.10}$$

$$\dot{y} = \Lambda^{-1}\Big[f^*_{motion} - \Psi(x, \dot{\hat{x}})\dot{\hat{x}} - p(x) + k_i\tilde{x}\Big] \tag{8.11}$$

where y, $\hat{x}$, k, and $\dot{\hat{x}}$ have been defined in Chapter 3, k_i defined in Chapter 4.

f^*_{motion} is the force control input to the observer, it is the force generated by the controller (indicated in Eq (9.4) later).

8.2.2 Formulation of Robust Observer-Based Controller

Based on the structure of the above observer, we propose the following controller to generate the required driving force:

$$f^*_{motion} = (k_s + k_{nd})\eta_p + w_e - k_i\tilde{x} \tag{8.12}$$

where k_{nd} is a positive controller gain defined as:

$$k_{nd} = 2k_n + \zeta_c k_0 + (k_s m_2 + k m_2)^2 k_n \tag{8.13}$$

and the $n \times 1$ observed filtered tracking error signal η_p and w_e are defined as:

$$\eta_p = \dot{x}_d + k_s e - \dot{\hat{x}} \tag{8.14}$$

$$w_e = \Lambda(x)[\ddot{x}_d + k_s(\dot{x}_d - \dot{\hat{x}})] + \Psi(x, \dot{\hat{x}})(\dot{x}_d + k_s e) + p(x) \tag{8.15}$$

The force command f^*_{motion} will be used in the observer indicated by (8.11).

Combining the force commands in motion and force subspace, the torques that are send to a robot joints can be obtained by:

$$\Gamma = J^T[\Omega f^*_{motion} + \hat{\Lambda}(x)\bar{\Omega} f^*_{force} + \bar{\Omega} f_{sensor}]$$

8.3 Overall System Stability Result and Analysis

Define the following Lyapunov function.

$$V = V_0 + V_1 + V_2 \tag{8.16}$$

where the three sub-Lyapunov function V_0, V_1, and V_2 are defined

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T k_i \tilde{x} \tag{8.17}$$

$$V_1 = \frac{1}{2}e^T e \tag{8.18}$$

$$V_2 = \frac{1}{2}\eta_p^T \Lambda(x)\eta_p \tag{8.19}$$

$\dot{V}$ can be obtained by

$$\dot{V} = \dot{V_0} + \dot{V_1} + \dot{V_2} \tag{8.20}$$

8.3.1 Lyapunov Function for Observation Error $\tilde{x}$ and $\dot{\tilde{x}}$

To form the bound of $\dot{V_0}$, first, take the time derivative of (9.1) and then substitute (8.11) into the resulting expression to yield:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} + p(x) - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} = f_{motion} \tag{8.21}$$

where the velocity observation error $\dot{\tilde{x}}$ is obtained by differentiating (3.5) with respect to time:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \tag{8.22}$$

Subtract (8.21) from (2.2), use (A.3) and (3.7) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x,\dot{x})\dot{\tilde{x}} + \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} + k_i\tilde{x} = 0 \tag{8.23}$$

Differentiate V_0 along (8.23) to get:

$$\begin{aligned} \dot{V}_0 &= \dot{\tilde{x}}^T \left[-\Psi(x,\dot{x})\dot{\tilde{x}} - \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} \right] \\ &\quad + \frac{1}{2}\dot{\tilde{x}}^T \dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i \tilde{x} \end{aligned} \tag{8.24}$$

Utilize (A.2) to yield:

$$\dot{V}_0 = -\dot{\tilde{x}} \left[\Psi(x,\dot{\hat{x}}) + k\Lambda(x) \right] \dot{\tilde{x}} \tag{8.25}$$

Then utilize (A.1) and (A.4) to get the upper bound of $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c \|\dot{\hat{x}}\| - km_1 \right) \|\dot{\tilde{x}}\|^2 \tag{8.26}$$

Substitute for $\dot{\hat{x}}$ from (8.14) into (8.26), and utilize (3.2) to get the new upper bound for $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\zeta_d + \zeta_c\|\eta_p\| + \zeta_c k_s \|e\| - km_1 \right) \|\dot{\tilde{x}}\|^2 \tag{8.27}$$

8.3.2 Lyapunov Function for Tracking Error e and η_p

From the form of (8.27), we are motivated to design a controller which ensures that the $\|e\|$ and $\|\eta_p\|$ terms in (8.27) are both driven to zero; hence, we are motivated to develop tracking error systems and the corresponding sub-Lyapunov functions to facilitate the goal.

The position tracking error system can be formed by differentiating (3.1) with respect to time to yield:

$$\dot{e} = \dot{x}_d - \dot{x}$$

Since $\dot{x}$ is not measurable, we use the estimated term $\dot{\hat{x}}$ to eliminate $\dot{x}$ and get the following equation:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} - \dot{\tilde{x}} \tag{8.28}$$

Add and subtract a fictitious controller [66] to the right-hand side of (8.28) to yield:

$$\dot{e} = \dot{x}_d - [\dot{x}_d + k_s e] + [\dot{x}_d + k_s e] - \dot{\hat{x}} - \dot{\tilde{x}} \tag{8.29}$$

where k_s is a positive controller gain.

Simplify (8.29) by utilizing (8.14) to get:

$$\dot{e} = -k_s e + \eta_p - \dot{\tilde{x}} \tag{8.30}$$

The upper bound for the time derivative of V_1 along (8.30) is given by:

$$\dot{V}_1 \leq -k_s \|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{8.31}$$

From the form of (8.31) and the fact that the form of (8.27) indicates that $\|\dot{\tilde{x}}\|$ can be driven to zero, we are motivated to design a force input controller which ensures that η_p can be driven to zero.

The tracking error system for η_p can be formed by differentiating (8.14) with respect to time, multiplying both sides of the resulting expression by $\Lambda(x)$, and substituting the right-hand side of (8.21) for $\ddot{\tilde{x}}$ to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= \Lambda(x)\ddot{x}_d + k_s\Lambda(x)(\dot{x}_d - \dot{x}) - k\Lambda(x)\dot{\tilde{x}} \\ &\quad -k_i\tilde{x} + \Psi(x,\dot{\hat{x}})\dot{\hat{x}} + p(x) - f_{motion} \end{aligned} \tag{8.32}$$

Substitute the force input given by (9.4) into (8.32), use the definitions of w_e and η_p to get:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -(k_s + k_{nd})\eta_p - (k + k_s)\Lambda(x)\dot{\tilde{x}} \\ &\quad -\Psi(x, \dot{\hat{x}})\eta_p \end{aligned} \tag{8.33}$$

Rewrite the term $\Psi(x, \dot{\hat{x}})\eta_p$ on the right-hand side of (8.33) in terms of $\dot{\tilde{x}}$, and utilize (A.3) and (3.7) to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -\Psi(x, \dot{x})\eta_p - (k_s + k_{nd})\eta_p \\ &\quad -(k_s + k_{nd})\Lambda(x)\dot{\tilde{x}} + \Psi(x, \dot{\tilde{x}})\eta_p \end{aligned} \tag{8.34}$$

In section 8.3, V_2 was defined as:

$$V_2 = \frac{1}{2}\eta_p^T \Lambda(x)\eta_p$$

Differentiate V_2 along (8.34), and utilize (A.2) to get:

$$\begin{aligned} \dot{V}_2 &= -(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &\quad +\eta_p^T\Psi(x, \dot{\tilde{x}})\eta_p \end{aligned} \tag{8.35}$$

From (8.35), utilize (A.1) and (A.4), we can obtain the following upper bound for $\dot{V}_2$:

$$\begin{aligned} \dot{V}_2 &\leq -(k_s + k_{nd})\|\eta_p\|^2 + (k + k_s)m_2\|\|\eta_p\|\dot{\tilde{x}}\| \\ &\quad +\zeta_c\|\eta_p\|^2\|\dot{\tilde{x}}\| \end{aligned} \tag{8.36}$$

8.3.3 Overall System Stability

Use the upper bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$, and utilize (3.6), (8.13), and (4.6), we can form the upper bound on $\dot{V}$:

$$\begin{aligned}
\dot{V} \leq \; & -k_s \|e\|^2 - k_s \left\|\eta_p\right\|^2 - k_s \left\|\dot{\tilde{x}}\right\|^2 \\
& + \left[\left\|\eta_p\right\| \left(\|e\| - 2k_n \left\|\eta_p\right\|\right)\right] \\
& + \left[\left\|\dot{\tilde{x}}\right\| \left(\|e\| - 2k_n \left\|\dot{\tilde{x}}\right\|\right)\right] \\
& + \left[(k+k_s)\, m_2 \left\|\eta_p\right\| \left(\left\|\dot{\tilde{x}}\right\| - (k+k_s)\, m_2 k_n \left\|\eta_p\right\|\right)\right] \\
& + \left(\zeta_c \zeta_d \left\|\dot{\tilde{x}}\right\|^2 - \zeta_c \zeta_d \left\|\dot{\tilde{x}}\right\|^2\right) \\
& - (k_o - \|err\|) \left(\zeta_c \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c \left\|\eta_p\right\|^2\right)
\end{aligned} \tag{8.37}$$

where we have used the fact derived from (4.6) that $\|err\| \geq \|e\|, \|\eta\|$, and $\|\dot{\tilde{x}}\|$.

By applying the nonlinear damping tool [66] on the three bracketed terms in (8.37), a new upper bound on $\dot{V}$ can be formed as:

$$\begin{aligned}
\dot{V} \leq \; & -(k_s - 1/k_n) \|e\|^2 - k_s \left\|\eta_p\right\|^2 - (k_s - 1/k_n) \left\|\dot{\tilde{x}}\right\|^2 \\
& - (k_o - \|err\|) \left(\zeta_c \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c k_s \left\|\dot{\tilde{x}}\right\|^2 + \zeta_c \left\|\eta_p\right\|^2\right)
\end{aligned} \tag{8.38}$$

From (8.38), it is easy to get:

$$\begin{aligned}
\dot{V} \leq \; & -(k_s - 1/k_n)(\|e\|^2 + \left\|\eta_p\right\|^2 + \left\|\dot{\tilde{x}}\right\|^2) \\
& \text{for} \;\; \|err\| \leq k_0
\end{aligned} \tag{8.39}$$

where err was defined in (4.6).

From (8.39) we can see that, if $k_s > 1/k_n$ and $k_0 \geq ||err(0)||$, we can get:

$$\dot{V} \leq 0 \tag{8.40}$$

From (8.40), we can get the conclusion that, the position tracking errors e, position estimation errors $\tilde{x}$, velocity estimation errors $\dot{\tilde{x}}$, and the observed filtered tracking

error signal η_p of the observer-controller are all bounded. Furthermore, the end-effector velocity tracking error is also bounded. In fact, after adding and subtracting $\dot{x}$ to the right-hand side of (8.14) and rearranging the terms, we can formulate the following inequality:

$$\|\dot{e}\| = \|\dot{x}_d - \dot{x}\| \leq \left\|\eta_p\right\| + k_s \|e\| + \|\dot{x}\| \tag{8.41}$$

Since each of the terms on the right-hand side of the above equation is bounded, $\|\dot{e}\|$ is also bounded.

8.4 Experimental Setup and Results

For our experiment, a JR3 force/torque sensor was installed on the end-effector of the PUMA robot.

Our task is to perform force control along Z axis, and at the same time, move the end-effector in XY plane with the following desired position trajectory while maintaining the initial end-effector orientation all the time.

$$\begin{aligned} p_{x_d} &= p_{x_0} + 50.0 \sin(0.4\pi t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{y_d} &= p_{y_0} + 50.0 \cos(0.4\pi t) \left(1 - e^{-0.05t^3}\right) \text{mm} \\ x_d &= \left[\begin{array}{cc} p_{x_d} & p_{y_d} \end{array}\right]^T \end{aligned} \tag{8.42}$$

where p_{x_0} and p_{y_0} are the initial positions in X and Y direction, respectively. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

The selection matrixes were selected to be:

$$\sum\nolimits_F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \tag{8.43}$$

$$\sum_M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8.44}$$

The controller gains were selected as diagonal gains matrices as following:

$$\begin{array}{lcl} k_{nd} & = & \text{diag}\{120, 120, 120, 35, 35, 35\} \\ k & = & \text{diag}\{108, 108, 108, 32, 32, 32\} \\ k_s & = & \text{diag}\{97, 97, 97, 30, 30, 30\} \\ k_i & = & \text{diag}\{2000, 2000, 2000, 3000, 3000, 3000\} \end{array} \tag{8.45}$$

8.4.1 Damping Control Algorithm

In order to reduce impact effect, the force/motion control will be switched to damping control when the contact force exceeds a predefined value. The damping control algorithm is shown below:

$$f_{motion} = \hat{\mathbf{\Lambda}}(x)\Omega f^*_{damp} + \Psi(x, \dot{x})\dot{x} + \hat{p}(x) \tag{8.46}$$

where,

$$f^*_{damp} = -K_d\dot{x} \tag{8.47}$$

and K_d is a damping gain.

When switching back to force/moton control, the following algorithm is introduced to insure smooth transition of control command:

$$f_{force} = \hat{\mathbf{\Lambda}}(x)\bar{\Omega} f^*_{force} + \bar{\Omega} f_{sensor} \tag{8.48}$$

where

$$f^*_{force} = f_{sigmoid}(K_{pf} e_{force} + K_{if} \int e_{force} dt) \tag{8.49}$$

where e_{force} is defined as

$$e_{force} = f_d - f_{contact} \tag{8.50}$$

Table 8.1: Tracking errors during impact moment with hard surface - Using observed velocity

e_x	e_y	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
4.69mm	3.09mm	0.20^o	0.97^o	0.48^o

The sigmoid function $f_{sigmoid}$ is defined as:

$$f_{sigmoid} = K(1 - e^{-at})/(1 + e^{-at}) \tag{8.51}$$

where K and a are positive scalars.

8.4.2 Experimental Results

The desired force is set to be -10N along Z-axis. With damping control, the force measurements are shown in Fig. 8.1. The results show that the contact force was maintained around -8.6N, fluctuating between 5N and -56N. In our experiment, the force sensor is mounted between a fixture and robot end-effector. When the robot end-effector bounced backwards from the contact surface, the inertia force of the fixture generated a positive sensor reading, in this case, a 5N reading.

The tracking errors of the end-effector are shown in Fig. 8.2. Table 8.1 shows the tracking errors at the moment the end-effector comes into contact with the hard surface.

If there is no damping control along the direction of force control, the robot will become unstable. The reason is that, during impact, the force reading is very high, and high force reading will drive the robot end-effector to retract from contact surface very fast. Due to high acceleration of the end-effector, the force/torque sensor's inertial force and moment will generate additional reading on the sensor, and the robot

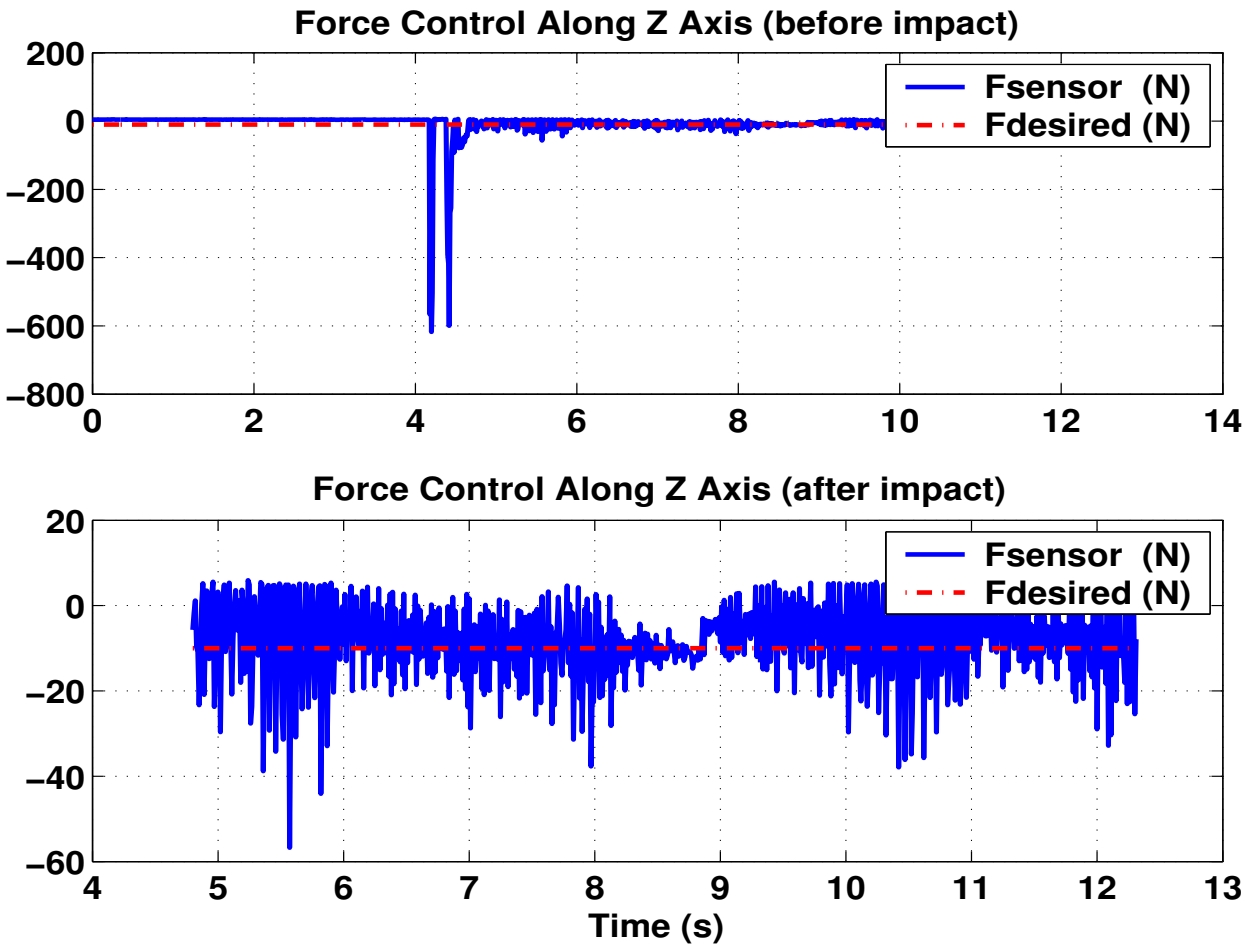

Figure 8.1: Force control response - Using observed velocity (lower graph shows response immediately after impact)

will react in high speed in order to offset the reading. Again, the high speed motion will generate high force/torque reading in the opposite direction, and this cyclic interaction will eventually cause stability problem of the robot. Fig. 8.3 indicates that the end-effector is unable to maintain its desired position immediately after the impact.

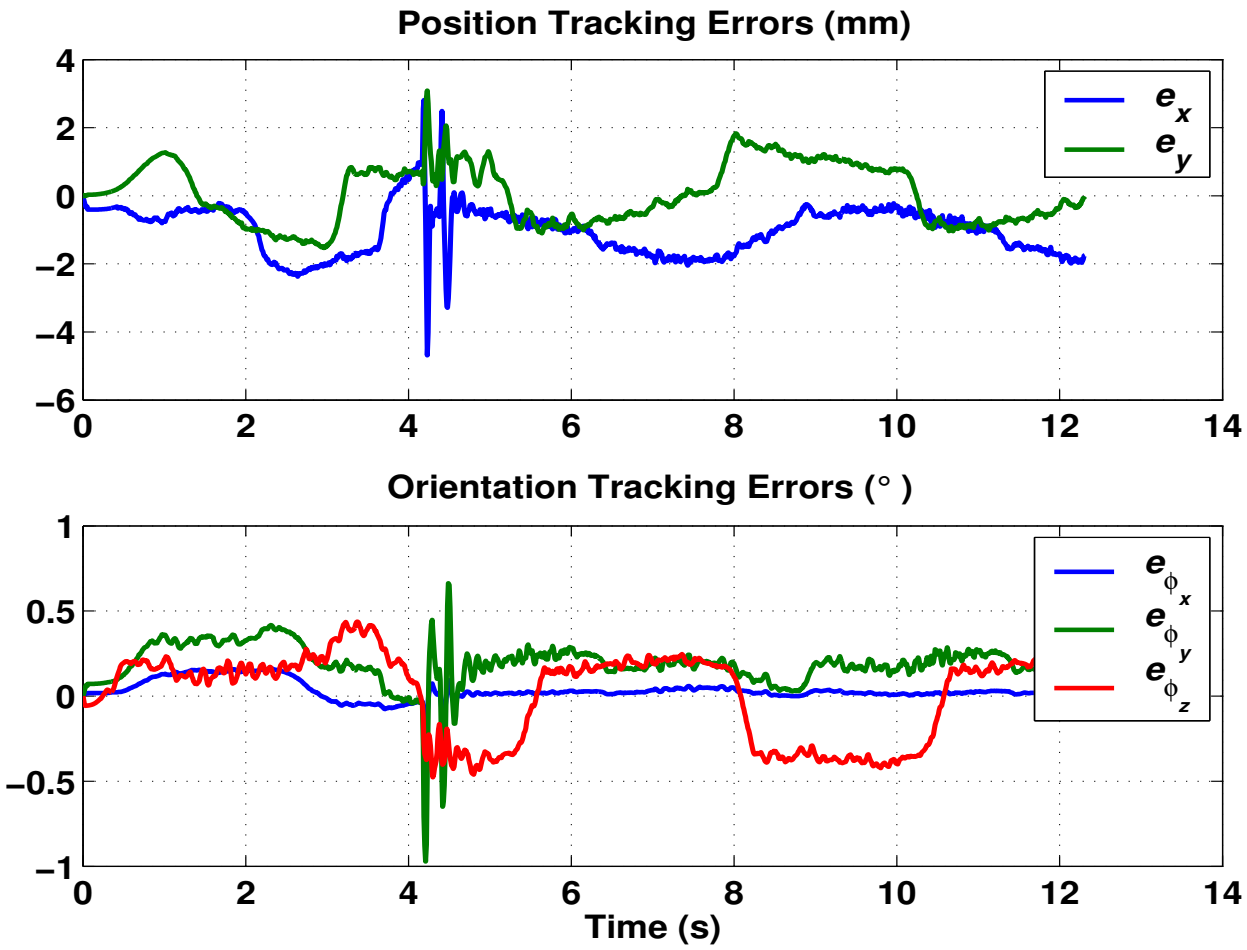

Figure 8.2: End-effector tracking errors - Using observed velocity

In order to make comparisons, we also implemented the controller indicated in Section 8.2.2, but the observed velocities $\hat{\dot{x}}$ were replaced by that obtained from backwards difference plus a low pass filter with cutoff frequency of 100Hz. We called it backwards difference controller.

During the experiments, we found that, due to large inherent noise ripple of filtered velocity, the controller gains values indicated in (8.45) were too high for the backwards difference controller, they caused the robot to vibrate. In order to reduce vibration,

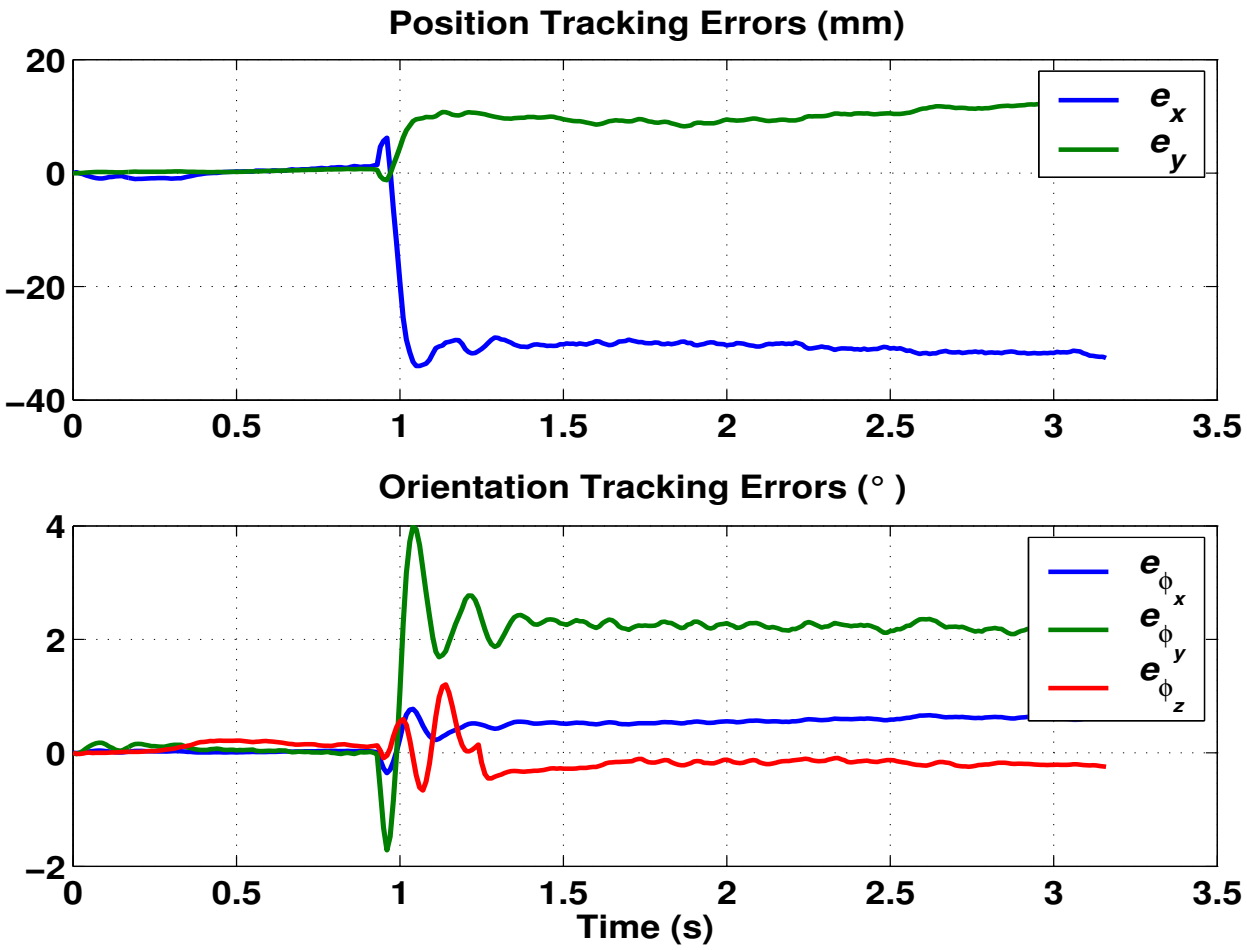

Figure 8.3: End-effector tracking errors - No damping control

the possibly highest gains were selected as listed in (8.52):

$$\begin{aligned} k_{nd} &= \text{diag}\{20, 20, 20, 12, 12, 12\} \\ k_s &= \text{diag}\{90, 90, 90, 30, 30, 30\} \\ k_i &= \text{diag}\{100, 100, 100, 120, 120, 120\} \end{aligned} \tag{8.52}$$

The force measurements are shown in Fig. 8.4. The results show that the contact force maintain around -8.2N, fluctuating between 16N and -69N.

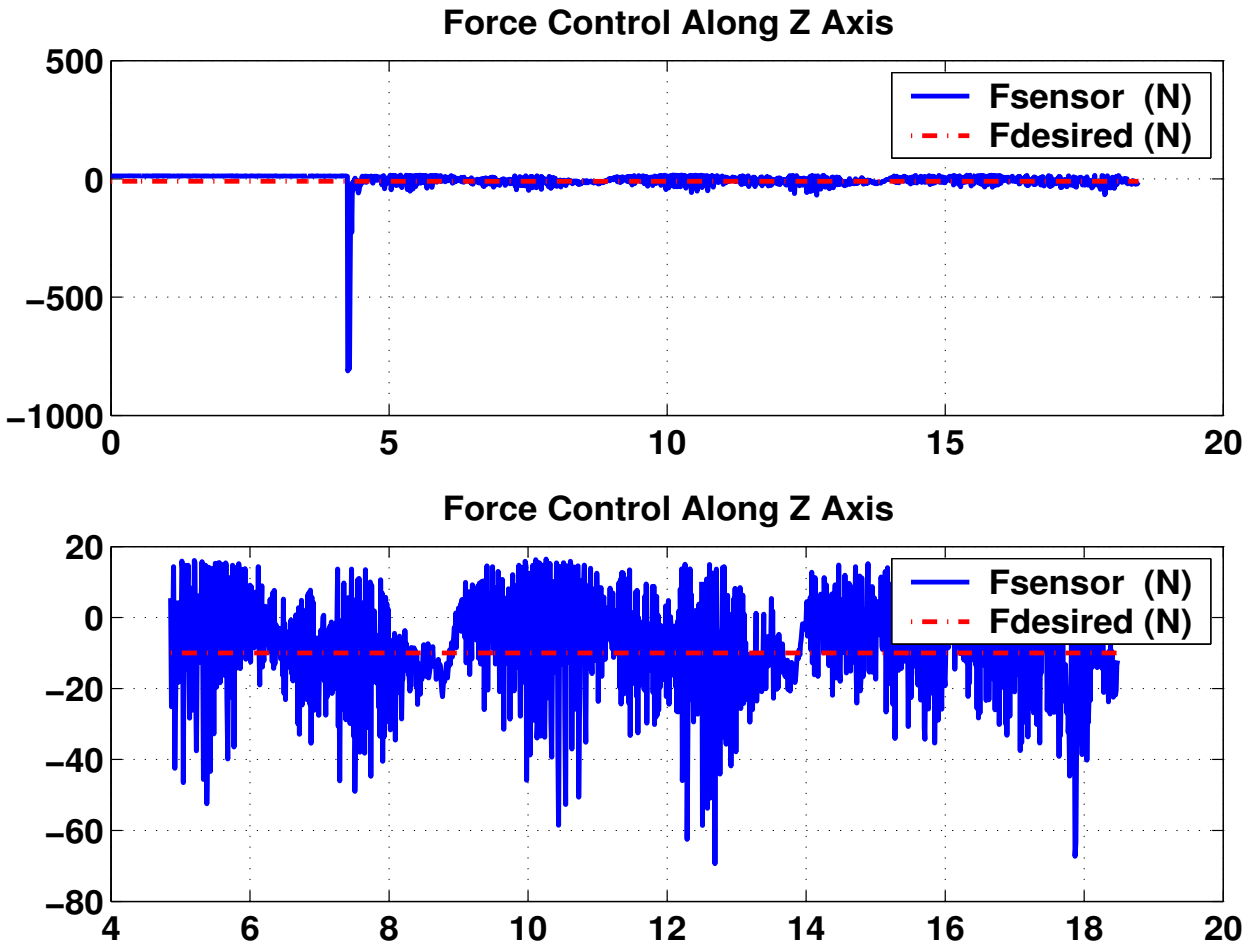

Figure 8.4: Force control response - Using filtered velocity (lower graph shows response immediately after impact)

The tracking errors of the end-effector are shown in Fig. 8.5. Table 8.2 shows the tracking errors at the moment when the end-effector comes into contact with hard surface.

From the results we can see that, in force control subspace, the force variation range using our proposed observer-controller is smaller than that using the backwards difference controller; in motion control subspace, the tracking errors using the

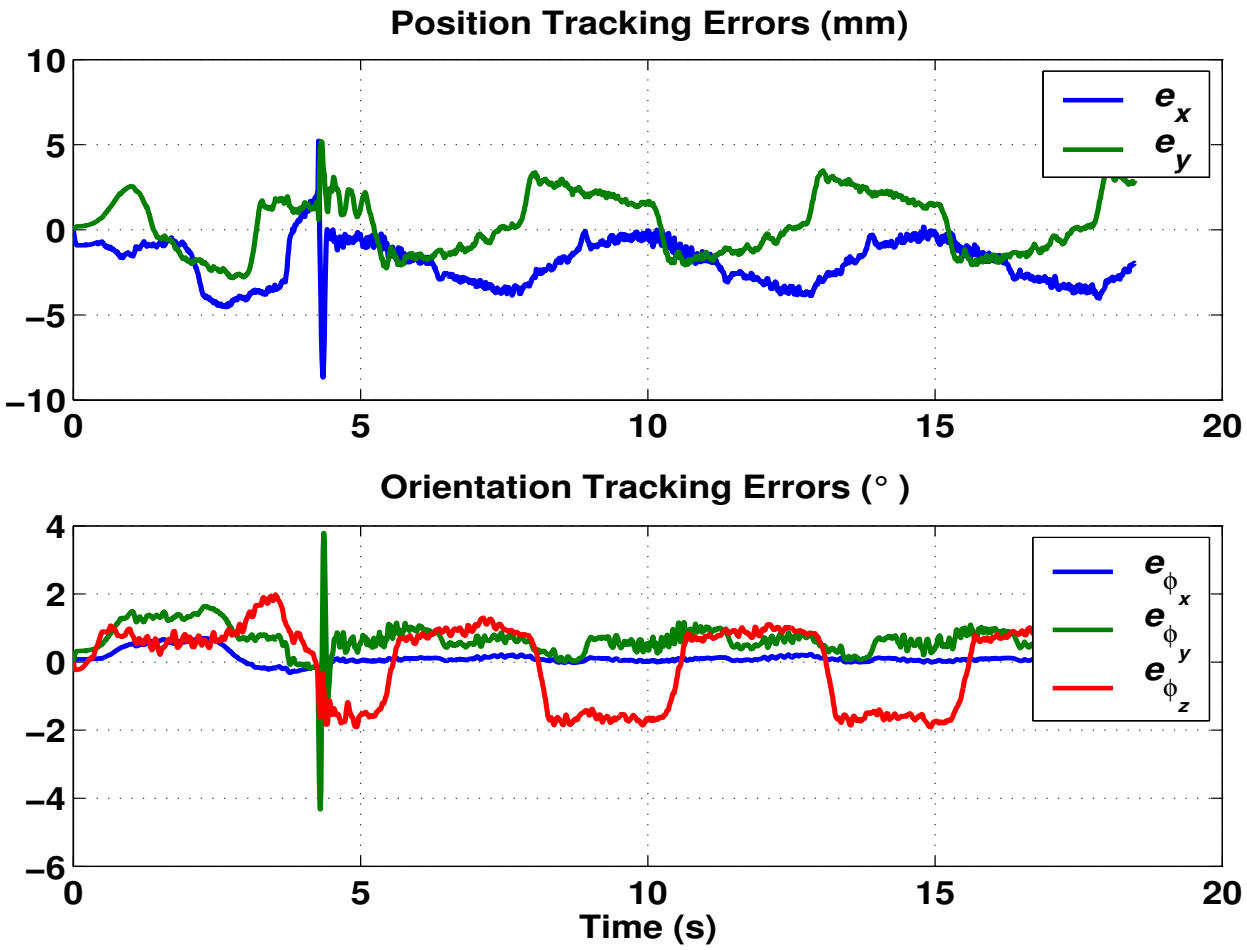

Figure 8.5: End-effector tracking errors - Using filtered velocity

observer-controller is much smaller than that using the backwards difference controller.

8.5 Conclusions

Both parallel force/motion control, and unified force and motion control have their advantages. By modifying Khatib's task specification matrix, we are able to select force and motion control subspaces in a systematic way. Meanwhile, by injecting weak

Table 8.2: Tracking errors during impact moment with hard surface - Using filtered velocity

e_x	e_y	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
8.68mm	5.20mm	0.80^o	4.33^o	1.98^o

motion control element along force control direction, the system becomes more stable and less sensitive to noisy force readings. Experimental results also indicate that the proposed observer-controller is able to achieve better motion and force control performance than the filtered backwards difference controller.

We would like to highlight here that Raibert's hybrid position/force control divide the whole space into motion and force subspaces, but does not use selection matrix to do this work [76]; Chiaverini's parallel force/position control approach does not use selection matrix either [48]; Khatib's unified force/motion control algorithm adopts a selection matrix to choose completely decoupled motion and force subspace in a systematic way [23]; while in our proposed algorithm, we modify Khatib's selection matrix in order to achieve parallel force/motion control along force control direction.

Again, this observer-based parallel force and motion controller is a semi-global controller, when the contact force is large and the controllers gains are not big enough, the system may face stability problems.

CHAPTER 9

CONTROL ALGORITHM 7: PARALLEL FORCE AND MOTION CONTROL USING ADAPTIVE OBSERVER-CONTROLLER

9.1 Introduction

In Chapter 5, we introduced an adaptive robust observer controller. In Chapter 8, we presented a parallel force and motion controller using observed velocity. In this chapter, we introduce a new controller that combines the adaptive controller in Chapter 5, and the force control algorithm in Chapter 8 to achieve parallel force and motion control.

9.2 Parallel Force and Motion Control

The proposed parallel force and motion controller consists of an adaptive motion controller and a force controller. The adaptive motion controller is designed to estimate joint velocity and compensate joint frictions, while the force controller is a PD controller.

9.2.1 Formulation of Operational Space Velocity Observer

The structure of the velocity observer is shown below. The difference between this observer and the one in Chapter 8 is that, there is an additional term $\hat{f}$, the estimated

friction expressed in operational space (given in Eq (5.20)), in the new controller.

$$\dot{\hat{x}} = y + k\tilde{x}, y(0) = -k\tilde{x}(0) \tag{9.1}$$

$$\dot{y} = \Lambda^{-1}\Big[f^*_{motion} - \Psi(x, \dot{\hat{x}})\dot{\hat{x}} - p(x) - \hat{f} + k_i\tilde{x}\Big] \tag{9.2}$$

$\dot{\hat{x}}$, y, and F have been defined in Chapter 3, k_i defined in Chapter 4, f^*_{motion} defined in Eq (9.4), and k defined by Eq (5.3).

The friction adaptation law is the same as that used in Chapter 5, and we repeat here for easy reference:

$$\begin{array}{rcl} \hat{\theta}(t) & = & \hat{\theta}(t - \Delta t) - K_{ad} W_j^T(\dot{\hat{q}}(t - \Delta t)) D_q \\ D_q & = & q(t) - q(t - \Delta t) - \dot{\hat{q}}(t - \Delta t)\Delta t \end{array} \tag{9.3}$$

By adding an estimated friction term $\hat{f}$ in the controller in Eq (8.15),we formulate the following controller:

$$f^*_{motion} = (k_s + k_{nd})\eta_p + w_e - k_i\tilde{x} \tag{9.4}$$

$$w_e = \Lambda(x)[\ddot{x}_d + k_s(\dot{x}_d - \dot{\hat{x}})] + \Psi(x, \dot{\hat{x}})(\dot{x}_d + k_s e) + \hat{f} + p(x) \tag{9.5}$$

where k_{nd}, η_p have been defined in Chapter 3.

Combining the force commands in motion and force subspace, the torques that are send to a robot joints can be obtained by:

$$\Gamma = J^T[\Omega f^*_{motion} + \hat{\Lambda}(x)\bar{\Omega} f^*_{force} + \bar{\Omega} f_{sensor}]$$

9.3 Overall System Stability Result and Analysis

To determine the stability of the control system, the following Lyapunov function is used:

$$V = V_0 + V_1 + V_2 \tag{9.6}$$

where the three sub-Lyapunov functions V_0, V_1, and V_2 are defined as:

$$V_0 = \frac{1}{2}\dot{\tilde{x}}^T \Lambda(x)\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^T k_i \tilde{x} + \frac{1}{2}\tilde{\theta}^T K_{ad}^{-1}\tilde{\theta} \tag{9.7}$$

where $\tilde{\theta}$ is the difference between the actual θ and the estimated $\hat{\theta}$, and the velocity observation error $\dot{\tilde{x}}$ is defined as the difference between the actual end-effector velocity $\dot{x}$ and the observed end-effector velocity $\dot{\hat{x}}$, it is obtained by differentiating (3.5) with respect to time:

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \tag{9.8}$$

$$V_1 = \frac{1}{2}e^T e \tag{9.9}$$

$$V_2 = \frac{1}{2}\eta_p^T \Lambda(x)\eta_p \tag{9.10}$$

Hence, $\dot{V}$ can be obtained by:

$$\dot{V} = \dot{V}_0 + \dot{V}_1 + \dot{V}_2 \tag{9.11}$$

9.3.1 Lyapunov Function for Observation Error $\tilde{x}$, $\dot{\tilde{x}}$ and $\tilde{\theta}$

V_0 is defined in (9.7). To form the bound of $\dot{V}_0$, first, take the time derivative of (3.3) and then substitute (9.2) into the resulting expression to yield:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\hat{x}} + p(x) + \hat{f} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x} = f_{motion} \tag{9.12}$$

Subtract (9.12) from (2.2), use (A.3) and (9.8) to yield the following closed-loop observer error system:

$$\Lambda(x)\ddot{\tilde{x}} + \Psi(x, \dot{x})\dot{\tilde{x}} + \Psi(x, \dot{\hat{x}})\dot{\tilde{x}} + k\Lambda(x)\dot{\tilde{x}} + k_i\tilde{x} + f - \hat{f} = 0 \tag{9.13}$$

Differentiate V_0 along (9.13) to get:

$$\begin{aligned}\dot{V}_0 = \ & \dot{\tilde{x}}^T \left[-\Psi(x,\dot{x})\dot{\tilde{x}} - \Psi(x,\dot{\hat{x}})\dot{\tilde{x}} - k\Lambda(x)\dot{\tilde{x}} - k_i\tilde{x}\right] \\ & + \frac{1}{2}\dot{\tilde{x}}^T \dot{\Lambda}(x)\dot{\tilde{x}} + \dot{\tilde{x}}^T k_i\tilde{x} - \dot{\tilde{x}}^T(f - \hat{f})\end{aligned} \tag{9.14}$$

Utilizing (A.2) to get:

$$\dot{V}_0 = -\dot{\tilde{x}}\left[\Psi(x,\dot{\hat{x}}) + k\Lambda(x)\right]\dot{\tilde{x}} - \dot{\tilde{x}}^T(f - \hat{f}) \tag{9.15}$$

Then utilize (A.1) and (A.4) to get the upper bound of $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\|\dot{\hat{x}}\| - km_1\right)\|\dot{\tilde{x}}\|^2 + \zeta_e\|\dot{\tilde{x}}\|^2 \tag{9.16}$$

Substitute for $\dot{\hat{x}}$ from (3.10) into (9.16), and utilize (3.2) to get the new upper bound for $\dot{V}_0$:

$$\dot{V}_0 \leq \left(\zeta_c\zeta_d + \zeta_c\|\eta_p\| + \zeta_c k_s\|e\| - km_1\right)\|\dot{\tilde{x}}\|^2 + \zeta_e\|\dot{\tilde{x}}\|^2 \tag{9.17}$$

9.3.2 Lyapunov Function for Tracking Error e and η_p

The evolution of the position tracking error with time can be derived by differentiating (3.1) with respect to time:

$$\dot{e} = \dot{x}_d - \dot{x}$$

Since $\dot{x}$ is not measurable, we use the estimated term $\dot{\hat{x}}$ to eliminate $\dot{x}$ and get the following equation:

$$\dot{e} = \dot{x}_d - \dot{\hat{x}} - \dot{\tilde{x}} \tag{9.18}$$

Simplify (9.18) by utilizing (3.10) to get:

$$\dot{e} = -k_s e + \eta_p - \dot{\tilde{x}} \tag{9.19}$$

V_1 is defined in (9.9), and the upper bound for the time derivative of V_1 along (9.19) is given by:

$$\dot{V}_1 \leq -k_s\|e\|^2 + \|e\|\|\eta_p\| + \|e\|\|\dot{\tilde{x}}\| \tag{9.20}$$

The evolution of η_p with time can be derived by differentiating (3.10) with respect to time, multiplying both sides of the resulting expression by $\Lambda(x)$, and substituting the expression from (9.12) for $\ddot{\hat{x}}$ to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= \Lambda(x)\ddot{x}_d + k_s\Lambda(x)(\dot{x}_d - \dot{x}) - k\Lambda(x)\dot{\tilde{x}} \\ &\quad -k_i\tilde{x} + \Psi(x,\dot{\hat{x}})\dot{\hat{x}} + p(x) + \hat{f} - f_{motion} \end{aligned} \tag{9.21}$$

Substitute the force command given by (9.4) into (9.21), use the definitions of w_e and η_p to get:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -(k_s + k_{nd})\eta_p - (k + k_s)\Lambda(x)\dot{\tilde{x}} \\ &\quad -\Psi(x,\dot{\hat{x}})\eta_p \end{aligned} \tag{9.22}$$

Rewrite the term $\Psi(x,\dot{\hat{x}})\eta_p$ on the right-hand side of (9.22) in terms of $\dot{\tilde{x}}$, and utilize (A.3) and (9.8) to yield:

$$\begin{aligned} \Lambda(x)\dot{\eta}_p &= -\Psi(x,\dot{x})\eta_p - (k_s + k_{nd})\eta_p \\ &\quad -(k + k_s)\Lambda(x)\dot{\tilde{x}} + \Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{9.23}$$

V_2 is defined in (9.10). Differentiating V_2 along (9.23) and utilizing (A.2) yields:

$$\begin{aligned} \dot{V}_2 &= -(k_s + k_{nd})\eta_p^T\eta_p - (k + k_s)\eta_p\Lambda(x)\dot{\tilde{x}} \\ &\quad +\eta_p^T\Psi(x,\dot{\tilde{x}})\eta_p \end{aligned} \tag{9.24}$$

From (9.24), and using (A.1) and (A.4), we can obtain the following upper bound for $\dot{V}_2$:

$$\begin{aligned}\dot{V}_2 \;\leq\; & -(k_s+k_{nd})\|\eta_p\|^2+(k+k_s)m_2\|\eta_p\|\|\dot{\tilde{x}}\| \\ & +\zeta_c\|\eta_p\|^2\|\dot{\tilde{x}}\| \end{aligned} \tag{9.25}$$

9.3.3 Overall System Stability Analysis

Use the upper bound of $\dot{V}_0$, $\dot{V}_1$ and $\dot{V}_2$, and utilize (3.6), (3.9), and (3.15), we can form the upper bound on $\dot{V}$:

$$\begin{aligned}\dot{V} \leq\; & -k_s\|e\|^2-k_s\left\|\eta_p\right\|^2-k_s\left\|\dot{\tilde{x}}\right\|^2 \\ & +\left\|\eta_p\right\|\left(\|e\|-2k_n\left\|\eta_p\right\|\right)+\left\|\dot{\tilde{x}}\right\|\left(\|e\|-2k_n\left\|\dot{\tilde{x}}\right\|\right) \\ & +(k+k_s)\,m_2\left\|\eta_p\right\|\left(\left\|\dot{\tilde{x}}\right\|-(k+k_s)\,m_2k_n\left\|\eta_p\right\|\right) \\ & +\left(\zeta_c\zeta_d\left\|\dot{\tilde{x}}\right\|^2-\zeta_c\zeta_d\left\|\dot{\tilde{x}}\right\|^2\right)+\zeta_e\|\dot{\tilde{x}}\|^2 \\ & -(k_o-\|err\|)\left(\zeta_c\left\|\dot{\tilde{x}}\right\|^2+\zeta_ck_s\left\|\dot{\tilde{x}}\right\|^2+\zeta_c\left\|\eta_p\right\|^2\right)\end{aligned} \tag{9.26}$$

where we have used the fact derived from (3.15) that $\|err\| \geq \|e\|, \|\eta\|$, and $\|\dot{\tilde{x}}\|$.

By applying the nonlinear damping tool [66] on the terms in the second and third lines on the right hand side of (9.26), a new upper bound on $\dot{V}$ can be formed as:

$$\begin{aligned}\dot{V} \leq\; & -(k_s-\tfrac{1}{k_n})\|e\|^2-k_s\left\|\eta_p\right\|^2-(k_s-\tfrac{1}{k_n}-\zeta_e)\left\|\dot{\tilde{x}}\right\|^2 \\ & -(k_o-\|err\|)\left(\zeta_c\left\|\dot{\tilde{x}}\right\|^2+\zeta_ck_s\left\|\dot{\tilde{x}}\right\|^2+\zeta_c\left\|\eta_p\right\|^2\right)\end{aligned} \tag{9.27}$$

From (9.27) we can see that, if $k_s > 1/k_n + \zeta_e$ and $k_0 \geq ||err||$, we can get:

$$\dot{V} \leq 0 \tag{9.28}$$

From (9.28), we can get the conclusion that, estimation errors of friction coefficients $\tilde{\theta}$, position tracking errors e, position estimation errors $\widetilde{x}$, velocity estimation

errors $\dot{\widetilde{x}}$, observed filtered tracking error signal η_p of the observer-controller are all bounded. Furthermore, the end-effector velocity tracking error is also bounded. In fact, after adding and subtracting $\dot{x}$ to the right-hand side of (3.10) and rearranging the terms, we can formulate the following inequality:

$$\|\dot{e}\| = \|\dot{x}_d - \dot{x}\| \leq \left\|\eta_p\right\| + k_s \|e\| + \|\dot{x}\| \tag{9.29}$$

Since each of the terms on the right-hand side of the above equation is bounded, $\|\dot{e}\|$ is also bounded.

9.4 Experimental Results

For our experiment, a JR3 force/torque sensor was installed on the end-effector of the PUMA robot.

Our task is to move the end-effector in XY plane with the following desired position trajectory while maintaining the initial end-effector orientation all the time.

$$\begin{array}{rcl} p_{x_d} & = & p_{x_0} + 50.0\sin(0.4\pi t)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ p_{y_d} & = & p_{y_0} + 50.0\cos(0.4\pi t)\left(1 - e^{-0.05t^3}\right) \text{mm} \\ x_d & = & \left[\begin{array}{cc} p_{x_d} & p_{y_d} \end{array}\right]^T \end{array} \tag{9.30}$$

where p_{x_0} and p_{y_0} are the initial positions in X and Y direction, respectively. The exponential terms are to ensure that the initial desired velocities and accelerations are all zeros.

The controller gains were selected as diagonal gains matrices as following:

$$\begin{array}{rcl} k_{nd} & = & \text{diag}\{120, 120, 120, 35, 35, 35\} \\ k & = & \text{diag}\{108, 108, 108, 32, 32, 32\} \\ k_s & = & \text{diag}\{97, 97, 97, 30, 30, 30\} \\ k_i & = & \text{diag}\{2000, 2000, 2000, 3000, 3000, 3000\} \end{array} \tag{9.31}$$

All the diagonal terms of the 18×18 friction adaptation gains K_{ad} were selected to be 500, all the diagonal terms of τ_{dec} were selected to be 1, and all the initial estimated friction coefficients are set to zeros.

The desired force is set to be -10N along Z-axis. The force measurements are shown in Fig. 9.1. The results show that the contact force maintained around -9.5N, fluctuating between -3N and -40N.

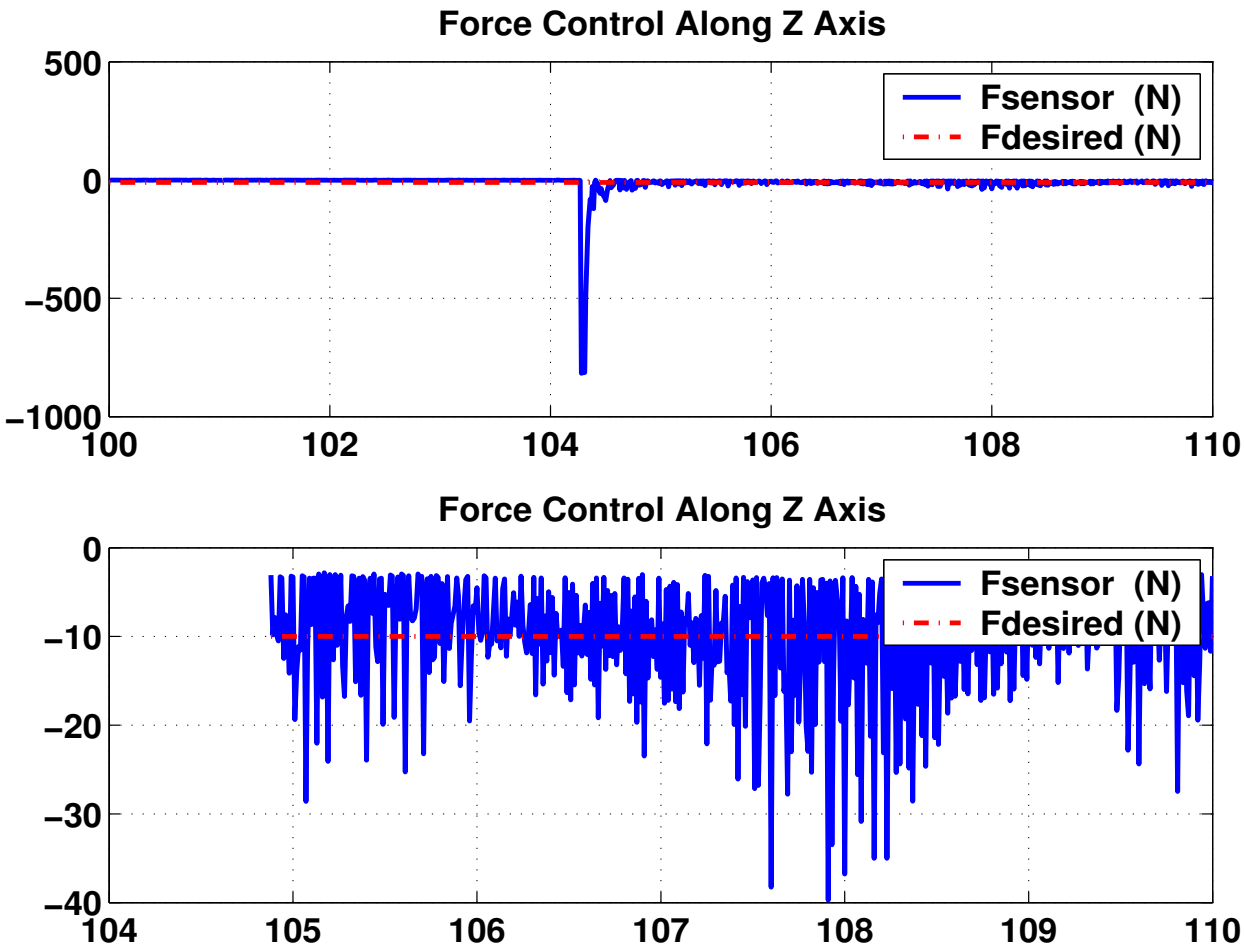

Figure 9.1: Force control response - Using adaptive observer-controller

The tracking errors of the end-effector are shown in Fig. 9.2. Table 9.1 shows the tracking errors at the moment when the end-effector comes into contact with hard surface.

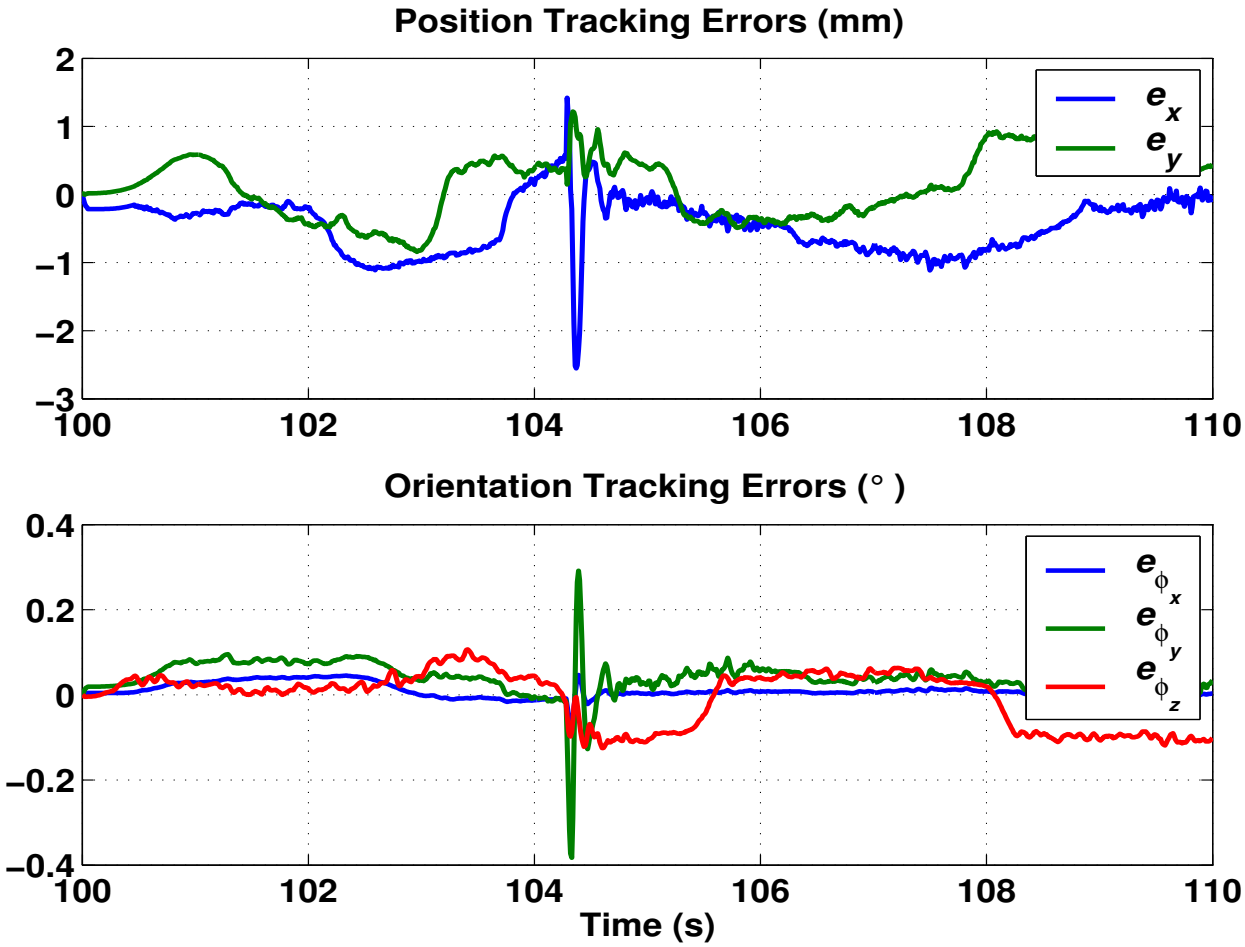

Figure 9.2: End-effector tracking errors - Using adaptive observer-controller

Compared with the results presented in Chapter 8, we can see that, in force control subspace, the force variation range using our proposed adaptive observer-controller is smaller than the controller without adaptation capability; in motion control subspace,

Table 9.1: Tracking errors during impact with hard surface - Using adaptive observer-controller

e_x	e_y	e_{ϕ_x}	e_{ϕ_y}	e_{ϕ_z}
2.55mm	1.22mm	0.07^o	0.38^o	0.13^o

the tracking errors using the adaptive controller is also much smaller than the one without adaptation.

9.5 Conclusions

The proposed adaptive observer-controller is able to achieve much smaller tracking errors than the observer-controller (presented in Chapter 8) that is not equipped with adaptation capability. It is also able to achieve better force control performance than that controller.

The improvement over the one developed in Chapter 8 is that, an adaptive control algorithm has been incorporated into the controller to achieve better control performance.

Like the controller in Chapter 8, when the contact force is large and the controllers gains are not big enough, the system may also face stability problems.

CHAPTER 10

CONTRIBUTIONS & FUTURE WORKS

This thesis derives from the development of advanced control algorithms based on observed velocity. Several controllers have been introduced in the thesis. Our developed observer-controllers are designed to mimic the dynamic behavior of a robot, hence, if the dynamic model of a robot is accurate enough, the observed velocity information can be more accurate than the filtered velocity information. In addition, the observed velocity is obtained from the integration of the calculated acceleration, the effect of integration is to reduce noise level, while the filtered velocity has to use backwards difference algorithm. The differentiation algorithm tends to amplify noise. Hence the observed velocity is "cleaner" than the filtered velocity. Our observer-based controllers are designed to make use of the merits of the "cleaner" observed velocity information. The performance of all the controllers are verified by experimental results.

Basically, three types of observer-controllers are introduced in the thesis. The first type of controllers are developed for trajectory tracking, as presented in Chapters 3 and 4; the second type of controllers are developed to compensate joint frictions, and at the same time, to achieve higher tracking accuracy, as introduced in Chapters 5

and 7; the third type of controllers are used for force control, which are presented in Chapters 8 and 9.

All these controllers can achieve a higher tracking accuracy and better force control performance than the controllers using filtered velocity, which verify the effectiveness of the developed controllers, thanks to the model-based velocity observers.

All the observer-based controllers introduced in this thesis are semi-global stable, and gains selections must satisfy some conditions. Under large degree of parametric uncertainties, payload variation, and contact force, the controller gains need to be large enough in order to make the system stable. However, a robot may vibrate if the gains become too high, hence, it may not be suitable to use an observer-based controller under large degree of uncertainties.

The observer-based controllers proposed in Chapters 3 and 4 were developed based on the assumption that the robot dynamic model was exactly known. Although experimental results indicate that the controllers are robust under parametric uncertainties and payload variations. It is still a challenging work to analyze the robustness of the controllers theoretically. When developing the adaptive controllers introduced in Chapters 5 and 6, we assumed that we possessed exact knowledge of a robot dynamic model except friction, and the adaptation algorithms were developed to identify and compensate joint friction. To design an adaptive controller that can identify not only joint friction, but also other parametric uncertainties still remains a challenge. In Chapters 8 and 9, parallel force and motion control algorithms based on observed velocity information were proposed. However, the force control performance was affected due to the noisy force sensor reading. In order to avoid using noisy force sensor

reading, how to estimate contact force based on the dynamic model of a robot and its environment around its contact point is an interesting and challenging future work.

APPENDIX A

PROPERTIES OF ROBOT DYNAMIC MODEL

For all the proposed observer-controller stability analysis, the following properties of the robot dynamic model need to be used:

Property 1 - The $n \times n$ kinetic energy matrix $\Lambda(x)$ defined in (2.2) satisfies the following inequality [77]:

$$m_1\|z\|^2 \leq z^T\Lambda(x)z \leq m_2\|z\|^2 = \|\Lambda(x)\|_{i2}\|z\|^2 \quad \forall z \in \Re^n \tag{A.1}$$

where m_1 and m_2 are known positive scalar constants. $\|\cdot\|$ represents the standard Euclidean norm, and $\|\cdot\|_{i2}$ represents the matrix induced two norm [72].

Property 2 - In joint space dynamic model (2.1), the centrifugal and Coriolis matrix satisfies the following relationship [2]:

$$B(q,y)z = B(q,z)y \quad \forall y, z \in \Re^n$$

Property 3 - In operational space dynamic model (2.2), the centrifugal and Coriolis matrix $\Psi(x,\dot{x})$ satisfies the following relationships:

$$z^T\left[\frac{1}{2}\dot{\Lambda}(x) - \Psi(x,\dot{x})\right]z = 0 \quad \forall z \in \Re^n \tag{A.2}$$

$$\Psi(x,y)z = \Psi(x,z)y \quad \forall y, z \in \Re^n \tag{A.3}$$

and

$$\|\Psi(x, \dot{x})\|_{i\infty} \leq \zeta_c \|\dot{x}\| \tag{A.4}$$

where ζ_c is a known positive scalar constant and $\|\cdot\|_{i\infty}$ represents the matrix induced infinity norm [72].

The proof of the property indicated by (A.3) is given in Lemma 1 (Appendix B).

APPENDIX B

LEMMAS FOR STABILITY ANALYSIS

To facilitate the overall system stability analysis, we provide the following two lemmas here.

Lemma 1: In operation space, $\Psi(x, \dot{x})$ in (2.2) satisfies the following relationship:

$$\Psi(x, y)z = \Psi(x, z)y \quad \forall y, z \in \Re^n$$

To prove this lemma, the robot manipulator's geometric model associated with the end-effector configuration parameters x can be written as:

$$x = G(q) = \left[\begin{array}{cccc} G_1(q) & G_2(q) & ... & G_n(q) \end{array} \right]^T$$

where q is the $n \times 1$ vector of joint position defined in (2.1).

At a given configuration q, $\dot{x}$ can be expressed as linear function of the joint velocities $\dot{q}$:

$$\begin{aligned} \dot{x} &= J(q)\dot{q} \\ &= \left[\begin{array}{cccc} \frac{\partial}{\partial q_1}G_1(q) & \frac{\partial}{\partial q_2}G_1(q) & ... & \frac{\partial}{\partial q_n}G_1(q) \\ \frac{\partial}{\partial q_1}G_2(q) & \frac{\partial}{\partial q_2}G_2(q) & ... & \frac{\partial}{\partial q_n}G_2(q) \\ ... & ... & ... & ... \\ \frac{\partial}{\partial q_1}G_n(q) & \frac{\partial}{\partial q_2}G_n(q) & ... & \frac{\partial}{\partial q_n}G_n(q) \end{array} \right] \left[\begin{array}{c} \dot{q}_1 \\ \dot{q}_2 \\ ... \\ \dot{q}_n \end{array} \right] \end{aligned} \tag{B.1}$$

Define

$$J_{ij} \triangleq \frac{\partial}{\partial q_j}G_i(q)$$

Equ (B.1) can be rewritten as:

$$\dot{x} = \begin{bmatrix} J_{11}(q) & J_{12}(q) & ... & J_{1n}(q) \\ J_{21}(q) & J_{22}(q) & ... & J_{2n}(q) \\ ... & ... & ... & ... \\ J_{n1}(q) & J_{n2}(q) & ... & J_{nn}(q) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ ... \\ \dot{q}_n \end{bmatrix}$$

Then we can get:

$$\dot{J}(q,\dot{q})\dot{\hat{q}} = \begin{bmatrix} \dot{J}_{11}(q,\dot{q}) & \dot{J}_{12}(q,\dot{q}) & ... & \dot{J}_{1n}(q,\dot{q}) \\ \dot{J}_{21}(q,\dot{q}) & \dot{J}_{22}(q,\dot{q}) & ... & \dot{J}_{2n}(q,\dot{q}) \\ ... & ... & ... & ... \\ \dot{J}_{n1}(q,\dot{q}) & \dot{J}_{n2}(q,\dot{q}) & ... & \dot{J}_{nn}(q,\dot{q}) \end{bmatrix} \begin{bmatrix} \dot{\hat{q}}_1 \\ \dot{\hat{q}}_2 \\ ... \\ \dot{\hat{q}}_n \end{bmatrix} \tag{B.2}$$

where $\hat{q} = \begin{bmatrix} \hat{q}_1 & \hat{q}_2 & ... & \hat{q}_n \end{bmatrix}^T$ is the $n \times 1$ vector of the estimated joint positions.

Define $\dot{J}_i(q,\dot{q})\dot{\hat{q}}$ as:

$$\dot{J}_i(q,\dot{q})\dot{\hat{q}} \triangleq \dot{J}_{i1}(q,\dot{q})\dot{\hat{q}}_1 + \dot{J}_{i2}(q,\dot{q})\dot{\hat{q}}_2 + ... + \dot{J}_{in}(q,\dot{q})\dot{\hat{q}}_n \tag{B.3}$$

Using (B.3), rewritten (B.2) as compact form:

$$\dot{J}(q,\dot{q})\dot{\hat{q}} = \begin{bmatrix} \dot{J}_1(q,\dot{q})\dot{\hat{q}} \\ \dot{J}_2(q,\dot{q})\dot{\hat{q}} \\ ... \\ \dot{J}_n(q,\dot{q})\dot{\hat{q}} \end{bmatrix} \tag{B.4}$$

Since

$$\begin{aligned} \dot{J}_1(q,\dot{q})\dot{\hat{q}} &= \dot{J}_{11}(q,\dot{q})\dot{\hat{q}}_1 + \dot{J}_{12}(q,\dot{q})\dot{\hat{q}}_2 + ... + \dot{J}_{1n}(q,\dot{q})\dot{\hat{q}}_n \\ &= \dot{J}_{11}(q,\dot{\hat{q}})\dot{q}_1 + \dot{J}_{12}(q,\dot{\hat{q}})\dot{q}_2 + ... + \dot{J}_{1n}(q,\dot{\hat{q}})\dot{q}_n \\ &= \dot{J}_1(q,\dot{\hat{q}})\dot{q} \end{aligned}$$

Similarly, we can get:

$$\dot{J}_2(q,\dot{q})\dot{\hat{q}} = \dot{J}_2(q,\dot{\hat{q}})\dot{q} \quad ... \quad \dot{J}_n(q,\dot{q})\dot{\hat{q}} = \dot{J}_n(q,\dot{\hat{q}})\dot{q}$$

Written in compact form:

$$\dot{J}(q,\dot{q})\dot{\hat{q}} = \dot{J}(q,\dot{\hat{q}})\dot{q} \tag{B.5}$$

For a non-redundant robot, $\Psi(x,\dot{x})\dot{\hat{x}}$ can be written as:

$$\begin{aligned}\Psi(x,\dot{x})\dot{\hat{x}} &= J^{-T}\left[B(q,\dot{q}) - A(q)J^{-1}\dot{J}(q,\dot{q})\right]J^{-1}J\dot{\hat{q}} \\ &= J^{-T}\left[B(q,\dot{q})\dot{\hat{q}} - A(q)J^{-1}\dot{J}(q,\dot{q})\dot{\hat{q}}\right]\end{aligned} \tag{B.6}$$

where the relationship between $\Psi(x,\dot{x})$ and $B(q,\dot{q})$ indicated in (2.3) is utilized.

Using *Property 2* and the result indicated by (B.5), (B.6) can be rewritten as:

$$\begin{aligned}\Psi(x,\dot{x})\dot{\hat{x}} &= J^{-T}\left[B(q,\dot{\hat{q}}) - A(q)J^{-1}\dot{J}(q,\dot{\hat{q}})\right]\dot{q} \\ &= J^{-T}\left[B(q,\dot{\hat{q}}) - A(q)J^{-1}\dot{J}(q,\dot{\hat{q}})\right]J^{-1}J\dot{q}\end{aligned} \tag{B.7}$$

Since $\dot{x} = J(q)\dot{q}$, substituting this equation into (B.7), finally we can get the following property indicated by *Lemma 1.*

Lemma 2: If a function $N_d(x,y) \in \Re$ is given by:

$$N_d = f(x)xy - k_d f^2(x)x^2$$

where $x, y \in \Re, f(x) \in \Re$ is a function dependent only on x, and k_d is a positive constant, then $N_d(x,y)$ can be upper bounded as follows:

$$N_d \leq \frac{y^2}{k_d}$$

The bounding of $N_d(x,y)$ in the above manner is often referred to as nonlinear damping [78] since a nonlinear control function, $k_d f^2(x)x^2$ can be used to damp-out an unmeasurable quantity (e.g., y) multiplied by a known, measurable nonlinear function $f(x)$.

BIBLIOGRAPHY

[1] P. K. Khosla and Kanade, "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators," *Int. J. Robot. Res.*, vol. 7, pp. 18–28, 1988.

[2] S. Nicosia and P. Tomei, "Robot control by using only joint position measurements," *IEEE Trans. Automat. Contr.*, vol. 35, no. 9, pp. 1058–1061, 1990.

[3] S. Nicosia, A. Tornambe, and P. Valigi, "Experimental results in state estimation of industrial robots," in *IEEE Proc. Conf. Decis. Contr.*, Honolulu, HI, USA, Dec. 1990, pp. 360–365.

[4] C. Canudas de Wit, K. Åström, and N. Fixot, "Computed torque control via a non-linear observer," *Int. J. Adaptive Contr. and Signal Process.*, vol. 4, no. 6, pp. 443–452, 1990.

[5] C. Canudas de Wit and J. Slotine, "Sliding observers for robot manipulators," *Automatica, J. IFAC*, vol. 27, no. 5, pp. 859–864, 1991.

[6] C. Canudas de Wit and N. Fixot, "Adaptive control of robot manipulators via velocity estimated feedback," *IEEE Trans. Automat. Contr.*, vol. 37, no. 8, pp. 1234–1237, 1992.

[7] C. Canudas de Wit and N. Fixot, "Robot control via robust estimated state feedback," *IEEE Trans. Automat. Contr.*, vol. 36, no. 12, pp. 1497–1501, 1991.

[8] M. Erlic and W. Lu, "Manipulator control with an exponentially stable velocity observer," in *Proc. Am. Contr. Conf.*, Chicago, IL, USA, June 1992, pp. 1241–1242.

[9] M. Erlic and W. Lu, "A reduced-order adaptive velocity observer for manipulator control," in *IEEE Proc. Int. Conf. Rob. Autom.*, Atlanta, GA, USA, May 1993, vol. 2, pp. 328–333.

[10] M. Erlic and W. Lu, "Impedence control without velocity measurements," in *IEEE Proc. Conf. Contr. Appl.*, Vancover, BC, Canada, Sept. 1993, pp. 47–52.

[11] W. Zhu, H. Chen, and Z. Zhang, "A variable structure robot control algorithm with an observer," *IEEE Trans. Robot. Automat.*, vol. 8, no. 4, pp. 486–492, 1992.

[12] H. Berghuis and H. Nijmeijer, "Robust control of robots using only position measurements," in *Proc. IFAC World Congress*, Sydney, Australia, July 1993, vol. 1, pp. 501–506.

[13] H. Berghuis, H. Nijmeijer, and P. Löhnberg, "Observer design in the tracking control problem of robots," in *Proc. IFAC Symp. NOLCOS'92*, Bordeaux, France, June 1992, pp. 588–593.

[14] H. Berghuis and H. Nijmeijer, "A passivity approach to controller-observer design for robots," *IEEE Trans. Robot. Automat.*, vol. 9, no. 6, pp. 740–754, 1993.

[15] Ser Yong Lim, D. Dawson, and K. Anderson, "Re-examining Nicosia's robot observer-controller from a backstepping perspective," *IEEE Trans. Contr. Syst. Technol.*, vol. 4, no. 3, pp. 304–310, 1996.

[16] Ser Yong Lim, D. Dawson, and M. Queiroz, "A partial state feedback controller for trajectory tracking of rigid-link flexible-joint robots using an observed backstepping approach," *J. Robotic Systems*, vol. 12, no. 11, pp. 727–746, 1995.

[17] Ser Yong Lim, J. Hu, and D. Dawson, "An output feedback controller for trajectory tracking of RLED robots using an observed backstepping approach," *IEEE Trans. Robot. Automat.*, vol. 11, no. 4, pp. 149–160, 1996.

[18] Ser Yong Lim, D. Dawson, J. Hu, and M. Queiroz, "An adaptive link position tracking controller for rigid-link flexible-joint robots without velocity measurements," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, no. 3, pp. 412–427, 1997.

[19] L. Hsu and F. Lizarralde, "Variable structure adaptive tracking control of rigid manipulators without velocity measurement," in *Proc. IFAC World Congress*, Sydney, Australia, Sept. 1993, pp. 145–148.

[20] J. Yuan and Y. Stepanenko, "Robust control of robotic manipulators without joint velocity feedback," *Int. J. Robust Nonlinear Contr.*, vol. 1, pp. 203–213, 1991.

[21] T. Burg, J. Hu, and D. Dawson, "A redesigned dcal controller without velocity measurements: Theory and experimentation," in *IEEE Proc. Conf. Decis. Contr.*, Lake Buena Vista, FL, USA, Dec. 1994, pp. 824–828.

[22] K. Kaneko and R. Horowitz, "Repetitive and adaptive control of robot manipulators with velocity estimation," *IEEE Trans. Robot. Automat.*, vol. 13, no. 2, pp. 204–217, 1997.

[23] O. Khatib, "A unified approach to motion and force control of robot manipulators: the operational space formulation," *IEEE Trans. Robot. Automat.*, vol. 3, no. 1, pp. 43–53, 1987.

[24] S. R. Ploen, J. E. Bobrow, and F. C. Park, "Geometric algorithms for operational space dynamics and control," in *IEEE Proc. Int. Conf. Rob. Autom.*, Albuquerque, NM, USA, Apr. 1997, pp. 1606–1611.

[25] K. Kreutz-Delgado, A. Jain, and G. Rodriguez, "Recursive formulation of operational space control," in *IEEE Proc. Int. Conf. Rob. Autom.*, Sacramento, CA, USA, Apr. 1991, pp. 1750–1753.

[26] B. Xian, M. Queiroz, D. Dawson, and I. Walker, "Task-space tracking control of redundant robot manipulators via quaternion feedback," in *IEEE Proc. Conf. Ctrl. Applicat.*, Mexico City, Mexico, Sept. 2001, pp. 363–368.

[27] P. R. Pagilla and M. Tomizuka, "An adaptive output feedback controller for robot arms: Stability and experiments," *Automatica, J. IFAC*, vol. 37, no. 7, pp. 983–995, 2001.

[28] F. Caccavale, C. Natale, and L. Villani, "Task-space control without velocity measurements," in *IEEE Proc. Int. Conf. Rob. Autom.*, Detroit, MI, USA, Apr. 1999, vol. 1, pp. 512–517.

[29] E.C. Dean-León, L.G. García-Valdovinos, and V. Parra-Vega, "Uncalibrated image-based position-force adaptive visual servoing for constrained robots under dynamic friction uncertainties," in *IEEE/RSJ Proc. Int. Conf. Intelli. Robots and Syst.*, Edmonton, Alberta, Canada, Aug. 2005, pp. 2013–2020.

[30] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto, "Fixed-step friction simulation: From classical coulomb model to modern continuous models," in *IEEE/RSJ Proc. Int. Conf. Intelli. Robots and Syst.*, Edmonton, Alberta, Canada, Aug. 2005, pp. 3910–3917.

[31] Yong Liang Zhu and Prabhakar R. Pagilla, "Static and dynamic friction compensation in trajectory tracking control of robots," in *IEEE Proc. Int. Conf. Rob. Autom.*, Washington, DC, USA, May 2002, vol. 3, pp. 2644–2649.

[32] Patrizio Tomei, "Robust adaptive friction compensation for tracking control of robot manipulators," *IEEE Trans. Automat. Contr.*, vol. 45, no. 11, pp. 2164–2169, 2000.

[33] A. Azenha and J. A. T. Machado, "Variable structure control of robots with nonlinear friction and backlash at the joints," in *IEEE Proc. Int. Conf. Rob. Autom.*, Minneapolis, MN, USA, Apr. 1996, vol. 1, pp. 366–371.

[34] G. Song, R. W. Longman, and L. Cai, "Integrated adaptive-robust control of robot manipulators with joint stick-slip friction," in *IEEE Proc. Int. Conf. Control Applicat.*, Hartford, CT, USA, Oct. 1997, vol. 1, pp. 177–182.

[35] L. Cai and G. Song, "A smooth robust nonlinear controller for robot manipulators with joint stick-slip friction," in *IEEE Proc. Int. Conf. Rob. Autom.*, Atlanta, GA, USA, May 1993, vol. 3, pp. 449–454.

[36] S. Jeon and M. Tomizuka, "Limit cycles due to friction forces in flexible joint mechanisms," in *IEEE/ASME Proc. Int. Conf. Avd. Intelli. Mechatronics*, Monterey, California, USA, July 2005, pp. 723–728.

[37] S. Jeon and M. Tomizuka, "Limit cycles due to friction forces in flexible joint mechanisms," in *Proc. 3rd IFAC Symposium on Mechatronic Systems*, Manly Beach, Sydney, Australia, Sept. 2004, pp. 705–710.

[38] N. Takesue, R. Kikuuwe, A. Sano, H. Mochiyama, and H. Fujimoto, "Tracking assist system using virtual friction field," in *IEEE/RSJ Proc. Int. Conf. Intelli. Robots and Syst.*, Edmonton, Alberta, Canada, Aug. 2005, pp. 2134–2139.

[39] M.S. Tsai, M.T. Lin, and H.T. Yau, "Command-based iterative learning control for compensation of servo lag and friction effects," in *IEEE/ASME Proc. Int. Conf. Avd. Intelli. Mechatronics*, Monterey, California, USA, July 2005, pp. 957–962.

[40] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *IEEE/ASME Trans. J. Dyn. Syst. Meas. Control*, vol. 103, pp. 126–133, 1981.

[41] A. Blomdell et al., "Extending an industrial robot controller: Implementation and applications of a fast open sensor interface," in *IEEE Robotics & Automation Magazine*, Sept. 2005, pp. 85–94.

[42] N. Hogan, "Impedance control: An approach to manipulation," *IEEE/ASME Trans. J. Dyn. Syst. Meas. Control*, vol. 107, no. 4, pp. 1–7, 1985.

[43] R. Volpe and P. Khosla, "Theoretical analysis and experimental verification of a manipulator/sensor/environment model for force control," in *Proc. Second Annual Int. Symp. Experimental Robot. Automat.*, Sacarmento, CA, USA, Apr. 1991, vol. 3.

[44] L. F. Baptista and José M. G. Sá da Costa, "Robot force control experiments in non-rigid environments," in *Proc. Second Annual Int. Symp. Experimental Robot. Automat.*, Vienna, Austria, Sept. 2000, vol. 1, pp. 159–164.

[45] D. Schutter J and V. Brussel H, "Compliant robot motin ii. a control approach based on external control loops," *Trans Int. J. Robot Res.*, vol. 7, pp. 18–33, 1988.

[46] D. Shutter J, "Improved force control laws for advanced tracking applications," in *IEEE Proc. Int. Conf. Robot Automat.*, Philadelphia, PA, USA, Sept. 1988, pp. 1497–1502.

[47] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Intergration for the next generation: Embedding force control into industrial robots," in *IEEE Robotics & Automation Magazine*, Sept. 2005, pp. 53–64.

[48] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 289–293, 1993.

[49] S. Chiaverini, B. Siciliano, and Luigi Villani, "Force and position tracking: Parallel control with stiffness adaptation," vol. 8, no. 1, pp. 1497–1502, Feb. 1998.

[50] R. J. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *IEEE Trans. Robot. Automat.*, vol. 4, pp. 549–556, 1988.

[51] J. Roy and L. L. Whitcomb, "Adaptive force control of position/velocity controlled robots: Theory and experiment," *IEEE Trans. Robot. Automat.*, vol. 18, no. 2, pp. 121–137, Apr. 2002.

[52] L. Cai and G. Song, "Robust position/force control of robot manipulators during contact tasks," in *Proc. American Control Conf.*, Baltimore, Maryland, USA, June 1994, pp. 216–220.

[53] B. Siciliano and L. Villani, "Adaptice compliant control of robot manipulators," vol. 4, no. 5, pp. 705–712, 1996.

[54] S. Chiaverini, B. Siciliano, and Luigi Villani, "An adaptive force/position control scheme for robot manipulators," vol. 7, no. 2, pp. 293–3003, 1997.

[55] Y. P. Chen and J. L. Chang, "Sliding-mode force control of manipulators," in *Proc. Natl. Sci. Counc. ROC(A)*, 1999, vol. 23, pp. 281–288.

[56] Y. P. Chen and Z. Z. Lu, "Force/position sliding-mode control of a robot manipulator in a non-rigid environment," vol. 1, no. 2, pp. 122–127, 1999.

[57] B. Siciliano and L. Villani, "An output feedback parallel force/position regulator for a robot manipulator in contact with a compliant environment," vol. 29, no. 5, pp. 295–300, 1997.

[58] B. Siciliano and L. Villani, "A force/position regulator for robot manipulators without velocity measurements," in *IEEE Proc. Int. Conf. Rob. Autom.*, Minneapolis, MN, USA, 1996, pp. 2567–2572.

[59] O. Khatib, *Advanced Robotics: Robots & the Human*, Professional Activities Centre, Faculty of Engineering, National University of Singapore, 2002.

[60] Marcio. S. de Queiroz, Darren. M. Dawson, and Fu Min Zhang, *Lyapunov-Based Control of Mechanical Systems*, Birkhauser, Boston, 2000.

[61] Oussama Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[62] Rodrigo Jamisola, Marcelo H. Ang, Tao Ming Lim, and Ser-Yong Lim, "Dynamics identification and control of an industrial robot," *The Ninth Intl. Conf. On Advanced Robotics*, pp. 323–328, 1999.

[63] J. Burdick B. Armstrong, O. Khatib, "The explicit dynamic model and inertial parameters of the puma 560 arm," *IEEE Intl. Conf. Robotics and Automation*, pp. 510–518, 1986.

[64] P.I. Corke, "A robotics toolbox for matlab," *IEEE Robotics and Automation Magazine*, vol. 3, no. 1, pp. 24–32, March 1996.

[65] Rodrigo Jamisola, "Dynamics identification and control of an industrial robot," Master of Engineering, Thesis, 2001, National University of Singapore.

[66] Petar V. Kokotović, "The joy of feedback: Nonlinear and adaptive," *IEEE Control Syst. Mag.*, vol. 12, no. 3, pp. 7–17, 1992.

[67] J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[68] Qing Hua Xia, Ser Yong Lim, and Marcelo H. Ang Jr, "An operational space observer-controller for trajectory tracking," in *IEEE Proc. Int. Conf. Advanc. Rob.*, Coimbra, Portugal, June/July 2003, vol. 2, pp. 923–928.

[69] Qing Hua Xia, Ser Yong Lim, Marcelo H. Ang Jr, and Tao Ming Lim, "Implementation of an output feedback controller in operational space," in *IEEE/RSJ Proc. Int. Conf. Intelli. Robots and Syst.*, Las Vegas, NV, USA, Oct. 2003, pp. 2761–2766.

[70] Qing Hua Xia, Ser Yong Lim, Marcelo H. Ang Jr, and Tao Ming Lim, "Robust observer-based controller and its application in robot control," in *IEEE/RSJ Proc. Int. Conf. Intelli. Robots and Syst.*, Edmonton, Alberta, Canada, Aug. 2005, to be published.

[71] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*, Macmillan Publishing Company, NY, 1993.

[72] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1978.

[73] Qing Hua Xia, Ser Yong Lim, Marcelo H. Ang Jr, and Tao Ming Lim, "Adaptive joint friction comensation using a model-based operational space velocity observer," in *IEEE Proc. Int. Conf. Rob. Autom.*, New Orleans, LA, USA, Apr./May 2004, pp. 3081–3086.

[74] Qing Hua Xia, Ser Yong Lim, Marcelo H. Ang Jr, and Tao Ming Lim, "Adaptive friction compensation for operational space tracking control with global asymptotic stability," in *3rd IFAC Symposium on Mechatronic Systems*, Manly Beach, Sydney, Australia, Sept. 2004, pp. 187–192.

[75] Qing Hua Xia, Ser Yong Lim, Marcelo H. Ang Jr, and Tao Ming Lim, "Adaptive friction compensation using a velocity observer," in *Experimental Robotics IX, Spring Tracts in Advanced Robotics*. vol. 21, pp. 77–86, Springer-Verlag Berlin Heidelberg 2006.

[76] M.Raibert and J.J.Craig, "Hybrid position/force control of manipulators," *Trans ASME J. Dyn, Systems Meas. Control*, vol. 102, pp. 126–133, 1981.

[77] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*, Wiley, NY, 1989.

[78] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, New York, NY, 1995.