ROBUST ADAPTIVE CONTROL OF TIME-VARYING MECHANICAL SYSTEMS: ANALYSIS AND EXPERIMENTS

By

KIU LING PAU

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1996

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1998

ROBUST ADAPTIVE CONTROL OF TIME-VARYING MECHANICAL SYSTEMS: ANALYSIS AND EXPERIMENTS

Thesis Approved:

Prabhakar Pagilla Sept 25/98

Thesis Advisor

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Prabhakar Pagilla, my thesis advisor, who has always been encouraging, supporting and helping me throughout the course of this research. I am also grateful to my committee members, Dr. Gary Young and Dr. Eduardo Misawa, for their help and guidance.

To my parents Ngie Siong Pau and Suk Yuk Yong, my sincere thanks for their love and support during all years of my education. Finally, I wish to thank my lab-mates, Biao for helping me with the experiments and Mr. Jerry Dale for his help on setting up the experiments.

TABLE OF CONTENTS

1 INTRODUCTION 1

1.1 Literature Review 2

1.2 Thesis Contribution 5

1.3 Thesis Outline 6

2 DYNAMIC MODELING OF TIME-VARYING LAGRANGIAN SYSTEMS 7

2.1 Time-Varying Mass and Inertia 8

2.2 Lagrangian Formulation 9

2.2.1 Kinetic Energy of the Manipulator 11

2.2.2 Potential Energy of the Manipulator 13

2.2.3 Dynamic Equations for n-link manipulators 13

2.2.4 Matrix Form 15

2.3 Dynamic Model Properties 16

3 CONTROLLER DESIGN FOR TRAJECTORY CONTROL 19

3.1 Background 19

3.2 Model Formulation for Robust Control Design 22

3.3 The Proposed Controllers 22

3.3.1 Robust Control 23

3.3.2 Nonlinear Compensators Design 27

3.3.3 Adaptive Robust Control 30

3.3.4 Adaptive Control 33

3.3.5 Integral Saturation Adaptive Robust Controller 35

4 EXPERIMENTAL BACKGROUND 37

4.1 Hardware Setup 37

4.2 Dynamic Equations and Properties 42

4.2.1 Dynamic Equations of NSK Two-Link SCARA Robot With Time-Varying Payload 42

4.3 Reference Trajectory 50

4.4 Implementation of Designed Controller 50

5 IMPLEMENTATION RESULTS 55

5.1 Results 55

5.2 Experimental Plots 58

6 CONCLUSIONS 65

Bibliography 66

Appendix: Source Code 70

LIST OF FIGURES

4.1 Experimental Setup 39

4.2 The reference joint trajectory for link 1 and 2 51

4.3 Reference trajectory in Cartesian coordinate 52

5.1 Position error of pumping water in using PD control 58

5.2 Position error of pumping water out using PD control 59

5.3 Pump water in using computed torque control 59

5.4 Pump water out using computed torque control 60

5.5 Position error of pumping water in using adaptive robust controller 60

5.6 Position error of pumping water out using adaptive robust controller 61

5.7 Torque of pumping water out using adaptive robust controller 61

5.8 Position error of pumping water out using pure adaptive controller 62

5.9 Parameter estimate 62

5.10 Position error of pumping water in using pure adaptive controller 63

5.11 Parameter estimate 63

5.12 Position error of pumping water out using PI saturation adaptive robust controller 64

5.13 Position error of pumping water in using PI saturation adaptive robust controller 64

Chapter 1

INTRODUCTION

The trajectory tracking control problem of the robot manipulators is an important research topic since many of the tasks performed by robot manipulators involve such a problem. The tasks include material handling, transportation, part assembly, etc. All these problems simulated the growing interest of many control system researchers. Many of these problems have been well studied and many different control approaches have been proposed to improve the performance of the trajectory tracking of robot manipulators. It is believed that all of these control approaches work well for a rigid robot manipulator with both known and unknown constant parameters. However, for many situations, some of the unknown parameters especially the mass of the payload or the mass of the links are time-varying. For example, carrying a leaking payload, pouring operations, filling operations and so on. Usually, if there is a slow change of the time-varying parameters, then we can neglect the effects due to the time-varying parameters. However, if the change is significant, then the rigid robot dynamics model for constant parameters may not accurately describe the dynamic behavior of the motion. Hence, it is desirable to develop a new model for robot manipulators with time-varying payloads and time-varying link masses.

The purpose of this research is to introduce a framework to deal with the time-varying payload problem in the robot motion control. By using Lagrange's equations, a new dynamic model for the robot manipulator with the time-varying masses is developed. We obtain the robot dynamics model in which the derivative of the time-varying parameters (rate of change of mass and inertia) are isolated from the inertia matrix and gravitational force vector. Also note that the manipulator here is modeled as a set of n rigid bodies connected in a serial

chain. In contrast to most of the existing work, where the Lyapunov direct method is used to study the stability, we present modified controllers that are applicable to trajectory control of robot manipulator consisting of time-varying masses.

Theoretical analysis and computer simulation are important parts of controller design but experimental implementation is necessary to check the practicality of the designed controllers. Since inherent factors such as unmodeled high-frequency dynamics and measurement noise are generally neglected in the stability analysis , we believe that the ultimate justification for the value and application of the designed controllers lies in its actually hardware implementation. Based on this perspective, this research also examines the designed controllers experimentally on an NSK two-link SCARA robot with a time-varying payload. It is shown that all the proposed controllers are stable and the tracking error either converges to zero or is bounded.

1.1 Literature Review

This section will first review motion control research of the robotic systems. The controller design for uncertain robotic system which of constant or slow change time-varying uncertain parameters are briefly reviewed. Finally, recent work for the robotic system consisting of time-varying parameters are reviewed.

A standard method for deriving the dynamic equations of mechanical systems is via the so called Euler-Lagrange equations. Detail derivation of the dynamic equations and its application can be found in [1, 2, 3]. In [4] and [5], derivation of Lagrange's equations for time-dependent generalized coordinate system is shown. [6] and [7] present a detail and standard method for deriving the dynamic equations for the rigid robot manipulators. However, all these derivations are based on the assumption that the mass of the link and payloads are constant or slowly changing with respect to time.

There are basically two common approaches used to control uncertain systems: the adaptive control approach and the robust control approach. In the adaptive approach, the controller attempts to converge the tracking error of the closed-system to zero by estimating the uncertain parameters of the system. For the robust approach, the controller has a fixed-

structure which yields "acceptable" performance for a given uncertain parameters set. In general, the adaptive control approach is more applicable than the robust control approach to a wide range of uncertainty and requires no prior knowledge of the uncertain bounds, but robust controller are simpler to implement and no time is required to tune the controller to the plant variations.

Significant research has been done in both adaptive and robust control of rigid manipulators. A survey of the robust control for rigid robot manipulators is given in [8]. This survey discusses five major robust control designs for rigid manipulators. These are feedback-linearization approach, passivity-based approach, variable-structure approach, robust saturation approach and robust adaptive approach. In our research, we emphasis on designing nonlinear compensators to eliminate the uncertainties of the system.

[9] proposed a robust control law of the robot manipulators by using Lyapunov stability theory to guarantee ultimate boundedness of the tracking error. This control strategy is based on the work of [10, 11, 12]. [13] proposed a linear robust feedback control law with a constant gain and conduct similar stability analysis as in [9]. It is also shown that globally asymptotic stability can be guaranteed as feedback gain approach infinity. Another simple robust nonlinear control is proposed by Spong in [14]. In [14], the uncertain bounds only depend on the inertia parameters, not desired trajectory and state vector. [15] proposed a control law that requires no a priori information about the upper bound on the uncertainty. A simple estimation law is used to estimate the upper bound so that uniform ultimate boundedness of the state can be guaranteed. All of these controllers are derived based on the the Lyapunov's direct method and the work of [10, 11, 12]. Other control approach such as variable-structure controllers are proposed in [16, 17] and their modified versions can be found in [18, 19].

The nonlinear compensator can be divided into three types: discontinuous type, saturation type and smooth type. Discontinuous and saturation type nonlinear compensators can be found in [6, 20, 21]. [22] presents a general design approach for robust controller design of uncertain dynamics systems. However, only practical stabilityof the closed-loop system is guaranteed. [23] proposed a smooth time-varying compensator that guarantees asymptotic stability of the system state. Application of this smooth compensator to compensate joint

stick-slip friction can be found in [24].

We also review here different adaptive control designs for motion control of robot manipulators. A discussion of adaptive controller design for robot manipulators can be found in [25]. This paper provides a framework for comparison of controllers which have been shown to be globally convergent. In [26, 27], a new version of adaptive controller is presented by introducing a modified error and convergence of both position error and velocity error were shown. In [28], the controller [26, 27] is modified to enhance its robustness. Furthermore observer based adaptive controller design can be found in [29, 30]. [20, 18, 31] also provide a detailed discussion on the application of adaptive control in the robotics industry.

Several modifications have been done to the general adaptive controller to enhance its robustness. A general design procedure for designing robust adaptive controller can be found in [32]. In [32], a new approach is shown to unify almost all global adaptive results for both continuous-time plant with or without modeling error. Application of robust adaptive control method to robotic manipulators is proposed in [33]. [34, 35] use smooth nonlinear compensator to design an adaptive robust controller and global asymptotic stability of the closed-loop system is guaranteed using this controller. Different approach of robust adaptive controller can be found in [36, 37]. The adaptive robust controller is the combination of best quantities of the adaptive controller and the robust controller. Some recent work can be found in [38], a scaled projection method is used to derived the adaptive robust controller. A new adaptive robust controller for controlling a high performance machine tool in the presence of both parameter uncertainties and uncertain nonlinearities is proposed in [39].

Practical experiments with comparisons between PD control, inverse dynamics and adaptive control is presented in [27]; the adaptive controller presented in [26] is implemented by using a recursive Newton-Euler formulation. In [39], comparison experiment between the proposed adaptive robust controller and PD and DOB (disturbance observer)is done on the X-Y table of a Matsunura 510VSS high-speed vertical machining center. Comparative experimental study of robust saturation based control scheme on a two-link robot manipulators is presented in [40]. The proposed controller can deal with different sources of uncertainty separately, and possesses an enhanced fine-tuning ability.

All of the controllers reviewed above work well under an assumption that the mass of the links or payloads are constant or slowly time-varying. However, under certain circumstance such as pouring and filling operation, the mass of the payload is not constant or rapidly time-varying. [41, 42] proposed a new dynamic model for robot manipulators consisting of the time-varying parameters. In [41, 42], the properties of the element by element product of matrices is used to isolate the time-varying parameters from the inertia matrix. However, the time-varying parameters robot models presents in the [41, 42] is not derived based on Lagrange's equations, but just simply assuming that the parameters of the system are time-varying. A switching type controller is proposed in these papers to yield asymptotic stability of the tracking error. A robust adaptive control for robot manipulators consisting of time-varying parameters is presented in [33]. In [33], the upper bound of the unknown parameters are required to obtain robustness of the closed-loop system.

1.2 Thesis Contribution

In this report, a new dynamics equations of the robot manipulators consisting of time-varying mass is derived based on the Lagrange's equations. Properties of the new model are carefully examined and the passivity property of this model is also shown. A new adaptive controller is designed under the assumption that the time-varying payload has finite number of non-zero derivatives. For this controllers design, we assume that the time rate of change of payload is a finite length polynomial in time. A robust adaptive controller based on the work of Corless-Leitmann is proposed. This controller includes an adaptation law for the upper bound of the parametric uncertainty. Using Lyapunov stability theory, uniform ultimate boundedness of the closed-loop system is shown for this controller. A modified robust adaptive control is also presented. In this controller, once the system solution enters a boundary layer, integral action is added to further reduce the tracking error and asymptotic stability is achieved. Stability of all the proposed control algorithms is proposed.

A unique experimental platform is designed for the implementation of the proposed controllers. The experimental platform consists of a two-link direct drive manipulator with

a time-varying payload on its end-effector. The time-varying nature for the payload is obtained by pumping fluid in and out of the vessel during the manipulator motion. Satisfactory experimental results validate the effectiveness of the proposed controllers.

1.3 Thesis Outline

The organization of this report is as follows. Chapter 2 establishes a new robot manipulator dynamic model which consists of time-varying mass. The controller designs for the time-varying mass system are given in chapter 3 and the stability analysis of the designed controllers is also showed. In Chapter 4, the hardware setup used in the experiments are described. Dynamic model of the two-link manipulator is derived. Implementation of designed controller to the two-link manipulator to illustrate the application and effectiveness of the proposed controllers and design procedure are also given in Chapter 4. In Chapter 5, experimental results are discussed. Chapter 6 lists some remarks of this work and suggestions for future research.

Chapter 2

DYNAMIC MODELING OF TIME-VARYING LAGRANGIAN SYSTEMS

This chapter deals with modeling of robot manipulators with time-varying masses. Application of a robot manipulator carrying a time-varying payload are many. For example, carrying a leaking payload, pouring operations, filling operations and so on. Typically, if there is a slow change of the time-varying masses, then we can neglect the effects due to the time varying masses. However, if the change is significant, then the rigid robot dynamics model for constant parameters may not accurately describe the dynamics. Hence, the objective of this chapter is to develop a new model for robot manipulators with time-varying payloads and time-varying link masses. Our approach is to first derive the kinetic and potential energy of the manipulator consisting of time-varying masses, then use Lagrange's equations to obtain a new model for robot manipulators with time-varying payloads and time-varying link masses. Note that the manipulator here is modeled as a set of n rigid bodies connected in a serial chain.

The material in this chapter is organized as follows. The validity of applying Lagrange's equations of motion for the time-varying case is verified. The derivation of the dynamic equations for time-varying systems by using the Lagrange's equations of motion is presented. Finally, the properties of the time-varying masses dynamic equation are also briefly discussed.

2.1 Time-Varying Mass and Inertia

In this section, we will study the relation between the mass and the inertia of the system. We can show that inertia of a system is always directly proportional to the mass of the system and has the same sign.

Assume that an incompressible, homogeneous fluid inside a uniform cross-sectional ($A(x,y)$) container is leaking. The fluid level function $h(t)$ is a time-varying function. If the container is empty initially, i.e. $\mathbf{V}(0) = 0$, then the volume of the fluid inside the container can be written as

$$d\mathbf{V}(t) = A(x,y)dh \qquad \text{or} \qquad \mathbf{V}(t) = \int A(x,y)dh \tag{2.1}$$

and the mass of the fluid is

$$\begin{aligned} m(t) &= \rho(x,y,z)\mathbf{V}(t) \\ &= \int_B \rho(x,y,z)A(x,y)dh \end{aligned} \tag{2.2}$$

where $\rho(x,y,z)$ is the density of the working fluid, and B is the region of three-dimensional space that is occupied by the fluid. Notice that the mass is only a function of time for homogeneous fluid and uniform cross-section container. The moment of inertia of the entire body about the principal axis is defined as

$$I = \int_m r^2 dm \tag{2.3}$$

where r is the distance from the element to the axis. Notice that the moment of inertia is always a positive quantity. Using equation (2.2), we can rewrite the moment of inertia as

$$I = \int_m r^2 \rho(x,y,z)A(x,y)dh \tag{2.4}$$

Since the working fluid level, $h(t)$, is a time-varying function, the moment of inertia is also a time varying function. Thus we can conclude that if the mass of the link i is a time-varying function, then the inertia of the link i is also a time-varying function. Furthermore, we can also show that the rate of change of the inertia always has the same sign as the rate of change of mass. From (2.3), if we take the derivative on the both sides, we obtain

$$dI = r^2 dm \tag{2.5}$$

Since $r^2 \geq 0$, so it is obvious that the sign of dI is the same as dm, i.e. if $\dot{m}_i$ of link i is positive at time t_i, then the $\dot{I}_i$ for that particular link is also positive at time t_i.

2.2 Lagrangian Formulation

In this section, we will study whether the kinetic energy of a system can be written in the quadratic form for systems with time-varying masses. We derive expressions for both kinetic and potential energies for systems with time-varying masses. Lagrange's equations are then used to derive the dynamic equation.

Lemma 2.1. *For both constant and time-varying mass system, the kinetic energy of the system is always a homogeneous quadratic form in the generalized velocities provided that the generalized constraints do not contain time explicitly.*

Proof. For the constant mass particle, the linear momentum p of the particle is defined to be the product of the particle mass and its velocity as

$$p \equiv mv. \tag{2.6}$$

where m and v represent the mass and velocity of the system. Then Newton's second law may be expressed as

$$F = \frac{dp}{dt} = \frac{d}{dt}(mv). \tag{2.7}$$

The change in kinetic energy of a system is defined as work done on the system by a force F in transforming the system from point 1 to point 2, and it can written as

$$\begin{aligned} dT &= W_{12} = \int_1^2 F \cdot ds \\ &= m\int_1^2 \frac{dv}{dt} \cdot v dt \\ &= \frac{m}{2}\int_1^2 \frac{d}{dt}(v^2)\, dt \end{aligned} \tag{2.8}$$

So the kinetic energy of the system can be expressed as

$$T = \frac{1}{2}mv^2. \tag{2.9}$$

For the time varying mass system, momentum can still expressed as

$$p(t) = m(t)\, v. \tag{2.10}$$

then we can write the differential equation of the kinetic energy as

$$\frac{d(m(t)\, T)}{dt} = F \cdot p \tag{2.11}$$

By using the Newton's second law, equation (2.11) can be written as

$$m(t)\, T = \int p \cdot dp \tag{2.12}$$

Since $dp(t) = v dm(t) + m(t)\, dv$, then

$$\begin{aligned} T &= \frac{1}{m(t)} \int m(t)\, v\, (v dm(t) + m(t)\, dv) \\ &= \frac{1}{2} m(t)\, v^2 \end{aligned} \tag{2.13}$$

Therefore, we can conclude that the kinetic energy of the system is always a quadratic form of velocity regardless of whether the mass of the system is constant or time-varying. However, this may not be true if the generalized constraints depend on time explicitly. Next, we are going to show that the kinetic energy of the system, T, is always a homogeneous quadratic form in the generalized velocities only if the generalized constraints do not contain time explicitly.

If the generalized constraints contain time explicitly, the Cartesian coordinate of any arbitrary mass m_i can be expressed in terms of n generalized coordinates as

$$r_i = r_i(q_1, q_2, \cdots, q_n, t) \tag{2.14}$$

then

$$\begin{aligned} v_i &= \frac{dr_i}{dt} \\ &= \sum_j^n \frac{\partial r_i}{\partial q_j} + \frac{\partial r_i}{\partial t} \end{aligned} \tag{2.15}$$

The total kinetic energy of the system can be written as

$$\begin{aligned} T &= \sum_i^n \frac{1}{2} m_i v_i^2 \\ &= \sum_i^n \frac{1}{2} m_i \left(\sum_j^n \frac{\partial r_i}{\partial q_j} + \frac{\partial r_i}{\partial t} \right)^2 \end{aligned} \tag{2.16}$$

It is clear that on carrying out the expansion, T can be written as the sum of the three homogeneous functions of the generalized velocities,

$$T = T_0 + T_1 + T_2 \tag{2.17}$$

where

$$\begin{aligned} T_0 &= \sum_i^n \frac{1}{2} m_i \left(\frac{\partial r_i}{\partial t} \right)^2, \\ T_1 &= \sum_j^n \left(\sum_i^n m_i \frac{\partial r_i}{\partial t} \cdot \frac{\partial r_i}{\partial q_j} \right) \dot{q}_j, \\ T_2 &= \frac{1}{2} \sum_{j,k}^n \left(\sum_i^n m_i \frac{\partial r_i}{\partial q_j} \cdot \frac{\partial r_i}{\partial q_k} \right) \dot{q}_j \dot{q}_k. \end{aligned} \tag{2.18}$$

Note that the T_0 is independent of the generalized velocities, T_1 is linear in the generalized velocities, and T_2 is quadratic in the generalized velocities. If the generalized constraints do not contain time explicitly, i.e. $\frac{\partial r_i}{\partial t} = 0$, then T is always a homogeneous quadratic form in the generalized velocities. In other word, T can be written as a homogeneous quadratic function for both constant and time-varying masses system only when the constraints are independent of time. For serial manipulators, this would imply that the length of the links should be fixed.

2.2.1 Kinetic Energy of the Manipulator

Consider an n link manipulator, the translation velocity, v_{ci}, and the angular velocity, ω_{ci}, of link i can be written as

$$\begin{aligned} v_{ci} &= J_{v_{ci}}(q)\dot{q}, \\ \omega_{ci} &= R_i^T(q) J_{\omega_i}(q)\dot{q} \end{aligned} \tag{2.19}$$

where $q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}^T$ is the joint vector, $R_i \in R^{3\times 3}$ is the rotational matrix that transforms a moving vector in the i-th frame to the inertial frame. $J_{v_{ci}} \in R^{3\times 3}, J_{\omega_i} \in R^{3\times 3}$ are Jacobian matrices of the manipulator that relate joint velocities to linear and angular velocities, respectively. Then the total kinetic energy of the manipulator of link i is

$$K = \frac{1}{2}\dot{q}^T \sum_{i=1}^n \left[m_i(t)\, J_{v_{ci}}^T(q) J_{v_{ci}}(q) + J_{\omega_i}^T(q) R_i(q) I_i(t)\, R_i^T(q) J_{\omega_i}(q) \right] \dot{q} \tag{2.20}$$

where $m_i(t), I_i(t)$ are the mass and inertia matrix of link i respectively. Note that $m_i(t)$ is a scalar time function, however $I_i(t)$ is the 3×3 symmetry matrix defined as

$$I(t) = \begin{bmatrix} \int(y^2+z^2)dm & -\int xydm & -\int xzdm \\ -\int xydm & \int(x^2+z^2)dm & -\int yzdm \\ -\int xzdm & -\int yzdm & \int(x^2+y^2)dm \end{bmatrix} \tag{2.21}$$

Define

$$M(q,m,I) = \sum_{i=1}^{n} \left[m_i(t)\, J_{v_{ci}}^T(q) J_{v_{ci}}(q) + J_{\omega_i}^T(q) R_i(q) I_i(t)\, R_i^T(q) J_{\omega_i}(q) \right] \tag{2.22}$$

which is always referred to as the manipulator inertia tensor. The matrix $M(q,m,I)$ incorporates all the mass properties of the entire manipulator for both translational and rotational motion about the centroid. $M(q,m,I)$ is a symmetric positive definite matrix because m and I of the manipulator are always positive. Note also that $M(q,m,I)$ also involves the Jacobian matrices, which vary with arm configuration. Therefore $M(q,m,I)$ represents only the instantaneous composite mass properties of the manipulator at the current arm configuration. The kinetic energy of the manipulator can be written as

$$K = \frac{1}{2}\dot{q}^T M(q,m,I)\dot{q} \tag{2.23}$$

where q, m are functions of time. Note that the kinetic energy is always a strictly positive quantity unless the system is at rest. Also notice that the kinetic energy of time-varying mass system not only depends on the generalized coordinate q, but also on the mass and inertia of the manipulator. However the kinetic energy is still a quadratic function of generalized velocity, $\dot{q}$, which is similar to the constant mass manipulators case.

2.2.2 Potential Energy of the Manipulator

The potential energy of the robot manipulator is given by

$$\begin{aligned} V &= \int_B g^T r dm = g^T r_c dm \\ V &\equiv V(q,m) \end{aligned} \tag{2.24}$$

Notice that the potential energy is only a function of the joint variables and the link masses, but not a function of the inertia.

2.2.3 Dynamic Equations for n-link manipulators

Define the Lagrangian as

$$\begin{aligned} L(q,m,I) &= K - V \\ &= \frac{1}{2}\dot{q}^T M(q,m,I)\dot{q} - V(q,m) \end{aligned} \tag{2.25}$$

or

$$L(q,m,I) = \frac{1}{2}\sum_{i,j}^{n} d_{i,j}(q,m,I)\dot{q}_i\dot{q}_j - V(q,m) \tag{2.26}$$

where $M(q,m,I) = [d_{ij}]$. By assuming that the generalized constraints do not contain time implicitly, $L(q,m,I)$ is only a function of the generalized coordinates q, and both time-varying mass $m(t)$ and inertia $I(t)$. By using the Lagrange's equations of motion

$$\frac{d}{dt}\left(\frac{\partial L}{d\dot{q}_i}\right) - \frac{\partial L}{dq_i} = \tau_i \tag{2.27}$$

We have

$$\begin{aligned} \frac{\partial L}{d\dot{q}_k} &= \sum_{j}^{n} d_{k,j}(q,m,I)\dot{q}_j \\ \frac{d}{dt}\left(\frac{\partial L}{d\dot{q}_k}\right) &= \sum_{j}^{n} d_{k,j}(q,m,I)\ddot{q}_j + \sum_{i,j}^{n}(\frac{d}{dt}d_{k,j}(q,m,I))\dot{q}_j \end{aligned} \tag{2.28}$$

For simplicity, let $d_{k,j} = d_{k,j}(q,m,I)$, then

$$\frac{d}{dt}\left(\frac{\partial L}{d\dot{q}_k}\right) = \sum_{j}^{n} d_{k,j}\ddot{q}_j + \sum_{i,j}^{n}\left[\frac{\partial d_{k,j}}{\partial q_i}\dot{q}_i + \frac{\partial d_{k,j}}{\partial m_i}\dot{m}_i + \frac{\partial d_{k,j}}{\partial I_i}\dot{I}_i\right]\dot{q}_j \tag{2.29}$$

Define that

$$\begin{aligned} \frac{\partial d_{k,j}}{\partial m_i} &= J_{v_{ci}}^T(q)J_{v_{ci}}(q) \equiv \alpha(q) \\ \frac{\partial d_{k,j}}{\partial I_i} &= J_{\omega_i}^T(q)R_i(q)R_i^T(q)J_{\omega_i}(q) \equiv \beta(q) \end{aligned} \tag{2.30}$$

$$\frac{d}{dt}\left(\frac{\partial L}{d\dot{q}_k}\right) = \sum_{j}^{n} d_{k,j}\ddot{q}_j + \sum_{j}^{n}\left[\frac{\partial d_{k,j}}{\partial q_i}\dot{q}_i + \alpha(q)\dot{m}_i + \beta(q)\dot{I}_i\right]\dot{q}_j \tag{2.31}$$

and

$$\frac{\partial L}{dq_k} = \frac{1}{2}\sum_{i,j}^{n}\frac{\partial d_{i,j}}{\partial q_k}\dot{q}_i\dot{q}_j - \frac{\partial V}{\partial q_k} \tag{2.32}$$

The Lagrange's equations become

$$\sum_{j}^{n} d_{k,j}\ddot{q}_j + \sum_{j}^{n}\left[\frac{\partial d_{k,j}}{\partial q_i} - \frac{1}{2}\sum_{i,j}^{n}\frac{\partial d_{i,j}}{\partial q_k}\right]\dot{q}_i\dot{q}_j + \sum_{j}^{n}\left[\alpha(q)\dot{m}_i + \beta(q)\dot{I}_i\right]\dot{q}_j - \frac{\partial V}{\partial q_k} = \tau_k \quad (2.33)$$

which can also be written as

$$M(q,m,I)\ddot{q} + C(q,\dot{q},m,I)\dot{q} + F(q,\dot{m},\dot{I})\dot{q} + g(q,m) = \tau \quad (2.34)$$

where the k,j-th element of the matrix $C(q,\dot{q},m,I)$ is defined as

$$\begin{aligned} c_{kj} &= \sum_{i=1}^{n} c_{ijk}(q,m,I)\dot{q}_i \\ &= \sum_{i=1}^{n}\frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i \end{aligned} \quad (2.35)$$

Note also c_{ijk} are known as Christoffel symbols of the first kind. For a fixed k, we have $c_{ijk} = c_{jik}$, which reduces the number of computations of these symbols by a factor of $\frac{1}{2}$. The term which involves joint variables q, and both derivative of mass and inertia, $\dot{m}$ and $\dot{I}$ is defined as follows

$$F(q,\dot{m},\dot{I}) = \sum_{i=1}^{n}\frac{\partial d_{k,j}}{\partial m_i}\dot{m}_i + \sum_{i=1}^{n}\frac{\partial d_{k,j}}{\partial I_i}\dot{I}_i \quad (2.36)$$

Equation (2.34) consists of four different types of terms. The first term represents inertia torques, and it involves the second derivative of the generalized coordinates. The second terms are quadratic terms in the first derivatives of the generalized coordinate, q. It accounts for the Coriolis ($i \neq j$) and Centrifugal ($i = j$) effects of motion. The third term involves the product between the first derivatives of the mass/inertia and the generalized coordinates, q. The last terms is represents gravity and it involves only q and $m(t)$ but not their derivatives.

2.2.4 Matrix Form

In this section, we will write the dynamic equations in matrix form. From equation (2.25), the Lagrangian of the system can be written as

$$L = \frac{1}{2}\dot{q}^T M(q,m,I)\dot{q} - V(q,m) \quad (2.37)$$

where q is the $(n \times 1)$ generalized coordinates vector, $M(q, m, I)$ is $(n \times n)$ the inertia matrix, $V(q, m)$ is the scalar potential energy function. Then the Lagrange's equations are given by

$$\begin{aligned} \frac{\partial L}{d\dot{q}} &= M(q, m, I)\dot{q} \\ \frac{d}{dt}\left(\frac{\partial L}{d\dot{q}}\right) &= M(q, m, I)\ddot{q} + \dot{M}(q, m, I)\dot{q} \end{aligned} \tag{2.38}$$

Also

$$\frac{\partial L}{dq} = \frac{1}{2}\left(\frac{\partial}{\partial q}\left(\dot{q}^T M(q, m, I)\dot{q}\right)\right)^T - \frac{\partial V}{\partial q} \tag{2.39}$$

Thus the dynamics equation is

$$M(q, m, I)\ddot{q} + \left\{\dot{M}(q, m, I)\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}\left(\dot{q}^T M(q, m, I)\dot{q}\right)\right)^T\right\} + g(q, m) = \tau \tag{2.40}$$

Notice that the time derivative of the inertia matrix $M(\cdot)$ can be written as

$$\dot{M}(q, m, I) = \frac{\partial M(q, m, I)}{\partial q}\dot{q} + \frac{\partial M(q, m, I)}{\partial m}\dot{m} + \frac{\partial M(q, m, I)}{\partial I}\dot{I} \tag{2.41}$$

By defining

$$C(q, \dot{q}, m, I) = \frac{\partial M(q, m, I)}{\partial q}\dot{q}\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}\left(\dot{q}^T M(q, m, I)\dot{q}\right)\right)^T \tag{2.42}$$

and also

$$F(q, \dot{m}, \dot{I}) = \frac{\partial M(q, m, I)}{\partial m}\dot{m} + \frac{\partial M(q, m, I)}{\partial I}\dot{I} \tag{2.43}$$

We obtain the dynamic equations in matrix form,

$$M(q, m, I)\ddot{q} + C(q, \dot{q}, m, I)\dot{q} + F(q, \dot{m}, \dot{I})\dot{q} + g(q, m) = \tau \tag{2.44}$$

2.3 Dynamic Model Properties

In this section, useful properties of the manipulator dynamics are briefly discussed. These properties are important in both stability and performance analysis. All these properties are derived from the fundamental physical behavior of the manipulator dynamics.

Property 1 The inertia matrix $M(q, m, I)$ of the time-varying system is a symmetric positive definite matrix which satisfies

$$\lambda_m I \leq M(q, m, I) \leq \lambda_M I \tag{2.45}$$

where λ_m ($\lambda_M < \infty$) denotes the positive minimum (maximum) eigenvalue of $M(q, m, I)$ for all manipulator configuration q. This property is valid only when both time-varying mass, $m(t)$ and inertia, $I(t)$, are some bounded time functions.

Property 2 The matrix $F(q, \dot{m}, \dot{I})$ of the time-varying mass system is a symmetric matrix. The positive definiteness of matrix $F(q, \dot{m}, \dot{I})$ depends on the sign of $\dot{m}$ and $\dot{I}$. The following are all possible cases:

- If $\dot{m}$ and $\dot{I}$ are constant, the matrix can be written as

$$F(q, \dot{m}, \dot{I}) = \sum_{i=1}^{n} \left[\dot{m}_i(t)\, J_{v_{ci}}^T(q) J_{v_{ci}}(q) + J_{\omega_i}^T(q) R_i(q) \dot{I}_i(t)\, R_i^T(q) J_{\omega_i}(q) \right]$$

  We should realize that the matrix $F(q, \dot{m}, I)$ has exactly the same structure as $M(q, m, I)$ except the both m and I are replaces with $\dot{m}$ and $\dot{I}$, respectively. Furthermore we also know that both $\dot{m}$ and $\dot{I}$ always have the same sign. Therefore we can conclude that if $\dot{m}$ of the system is constant and always positive, then the matrix $F(q, \dot{m}, \dot{I})$ is a symmetric positive definite matrix which satisfies

$$\lambda_F I \leq \|F(q, \dot{m}, \dot{I})\| \leq \lambda_F I \tag{2.46}$$

  where λ_f ($\lambda_F < \infty$) denotes the positive minimum (maximum) eigenvalue of $F(q, \dot{m}, \dot{I})$ for all configuration q. On the other hand, we can also show that if $\dot{m}$ is a constant and always negative, then matrix $F(q, \dot{m}, \dot{I})$ is a symmetry negative definite matrix with the same bound as shown above.

- If $\dot{m}$ and $\dot{I}$ are some time dependent functions, it is always possible to write $F(q, \dot{m}, \dot{I})$ as

$$F(q, \dot{m}, \dot{I}) = \frac{\partial M(q, m, I)}{\partial m} \dot{m} + \frac{\partial M(q, m, I)}{\partial I} \dot{I}$$

  where $\frac{\partial M(q,m,I)}{\partial m}$ and $\frac{\partial M(q,m,I))}{\partial I}$ are always symmetric positive definite matrix. Again, if $\dot{m}$ is bounded by some positive definite function, then $F(q, \dot{m}, \dot{I})$ is a bounded

symmetric positive matrix. On the other hand, it is also true if $\dot{m}$ is bounded by some negative function. However, if $\dot{m}$ is not bounded by any function, then no conclusion about definiteness of $F(q, \dot{m}, \dot{I})$ can be drawn.

Property 3 The matrix $N(q, \dot{q}, m, I) = \dot{M}(q, m, I) - 2C(q, \dot{q}, m, I) - F(q, \dot{m}, \dot{I})$ is skew-symmetric, i.e. matrix $N(q, \dot{q}, m, I)$ always satisfies the following

$$N(q, \dot{q}, m, I) = -N(q, \dot{q}, m, I)^T \qquad \text{or} \qquad x^T N(q, \dot{q}, m, I)x = 0 \tag{2.47}$$

for any $(n \times 1)$ vector x.

Proof. Given the inertia matrix $M(q, m, I)$. By using chain rule, the kj-th component of the inertia matrix is given as

$$\begin{aligned} \dot{d}_{kj} &= \frac{\partial d_{kj}}{\partial q_i}\dot{q} + \frac{\partial d_{kj}}{\partial m}\dot{m} + \frac{\partial d_{kj}}{\partial I}\dot{I} \\ &= \sum_{i=1}^{n}\left[\frac{\partial d_{kj}}{\partial q_i}\right]\dot{q}_i + \sum_{i=1}^{n}\left[\frac{\partial d_{kj}}{\partial m}\right]\dot{m}_i + \sum_{i=1}^{n}\left[\frac{\partial d_{kj}}{\partial I}\right]\dot{I}_i \end{aligned} \tag{2.48}$$

note that $\dot{d}_{kj} - f_{kj} = \sum_{i=1}^{n}\left[\frac{\partial d_{kj}}{\partial q}\right]\dot{q}_i$. Therefore, the kj-th component of $N(q, \dot{q}, m, I) = \dot{M}(q, \dot{q}, m, I) - 2C(q, \dot{q}, \dot{m}, \dot{I}) - F(q, m, I)$ is given by

$$\begin{aligned} n_{kj} &= \dot{d}_{kj} - 2c_{kj} - f_{kj} \\ &= \sum_{i=1}^{n}\left[\frac{\partial d_{kj}}{\partial q_i} - \left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\right]\dot{q}_i \\ &= \sum_{i=1}^{n}\left[\frac{\partial d_{ij}}{\partial q_k} - \frac{\partial d_{ki}}{\partial q_j}\right] \end{aligned} \tag{2.49}$$

Since the inertia matrix $M(q, m, I)$ is symmetric ($d_{ij} = d_{ji}$), then from above, we can conclude that

$$n_{jk} = -n_{kj} \tag{2.50}$$

which completes the proof.

Property 4 The dynamic equation (2.34) is linear in the unknown parameters. This property may be expressed as

$$M(q, m, I)\ddot{q} + C(q, \dot{q}, m, I)\dot{q} + F(q, \dot{m}, \dot{I})\dot{q} + g(q, m) = Y(q, \dot{q}, \ddot{q})\phi \tag{2.51}$$

where ϕ is the parameter vector and $Y(q, \dot{q}, \ddot{q})$ is a matrix which depends on the joint variables, joint velocities and joint acceleration.

The above properties are important in understanding the dynamics and are useful during stability analysis. For a revolute-joint manipulator, matrix $M(q, m, I)$ is not only positive definite but also its dependence on q is in the form of trigonometric function, sine and cosine. This implies that if both time-varying mass and inertia can be bounded by a constant, then $M(q, m, I)$ can also be bounded by a constant. On the other hand, if the both time-varying mass and inertia are bounded by some functions, then the bounds of $M(q, m, I)$ are not a constant but are general functions.

Chapter 3

CONTROLLER DESIGN FOR TRAJECTORY CONTROL

In this chapter, we use Lyapunov-based control schemes and passivity-based control schemes to design robust and adaptive controllers. In comparison with the inverse dynamics method, passivity based controllers are expected to have better robustness properties because they do not rely on the exact cancellation of nonlinearities. Moreover, passivity based controllers also require less computation time as compared to inverse dynamics method. The outline of this chapter is as follows: First, we give some background material that will be used in the rest of the report. Then, we formulate the closed loop dynamics for the system in the presence of parameter uncertainty. Our control designs will be composed of an inner controller and an outer controller. The inner controller deals with the cancellation of known terms. We will discuss several different control strategies for the outer control. Stability of all designed controllers will be analyzed.

3.1 Background

Class K Function

A function $\gamma(\cdot) : R_+ \rightarrow R_+$ belongs to class **K** if and only if it is continuous and satisfies

$$r_1 \leq r_2 \Rightarrow \gamma(r_1) \leq \gamma(r_2) \qquad \forall\, r_1, r_2 \in R.$$

$$\gamma(0) = 0, \qquad \text{and} \qquad r > 0 \Rightarrow \gamma(r) > 0.$$

A function $\gamma(\cdot): R_+ \to R_+$ belongs to class **KR** if and only if it belong to **K** and

$$\lim_{r\to\infty} \gamma(r) = \infty.$$

Global Practical Stabilizability

In this section we give a definition of global practical stabilizability which is the desired behavior we would like our closed loop system to possess in the presence of uncertainties. The uncertain system we consider is of the form

$$\dot{y}(t) = F(t, y(t), u(t), \omega) \tag{3.1}$$

where $t \in R$ is the time variable, $y(t) \in R^n$ is the state variable, and $u \in R^m$ is the control input. All the uncertainty in the system is represented by the lumped uncertain element $\omega \in \Omega$. The only information assumed about ω is that the set Ω is not an empty set. Let G_r denote the closed ball of radius r, that is

$$G_r = \{y \in R^n : \|y\| \leq r\}. \tag{3.2}$$

Note that $\|\cdot\|$ denotes the Euclidean 2-norm. System (3.1) is said to be globally practical stabilizable with respect to the closed ball G_{d_0} if and only if, given any $\underline{d} > d_0$, there exists a feedback control $p: R \times R^n \to R^m$ for which the following properties hold:

(i) Existence of Solutions.

Given $(y_0, t_0) \in R^n \times R$, the closed loop system

$$\dot{y}(t) = F(t, y(t), p(t,y), \omega) \tag{3.3}$$

has a solution $y(\cdot): [t_0, t_1) \to R^n$, $y(t_0) = y_0$, $\forall t_1 > t_0$.

(ii) Uniform Boundedness.

Given $r \in (0, \infty)$, there exists a positive quantity $d(r) < \infty$ such that for all solutions $y(\cdot): [t_0, t_1) \to R^n$, $y(t_0) = y_0$, of the closed loop system (3.3)

$$\|y_0\| \leq r \Rightarrow \|y(t)\| \leq d(r), \text{ for all } t \in [t_0, t_1).$$

(iii) Extension of Solutions.

Every solution $y(\cdot) : [t_0, t_1) \to R^n$ of closed loop system (3.3) can be extended over $[t_0, \infty)$.

(iv) Uniform Ultimate Boundedness.

Given any $\overline{d} > \underline{d}$ and any $r \in (0, \infty)$, there is a $T(\overline{d}, r) \in [t_0, \infty)$ such that for every solution $y(\cdot) : [t_0, \infty) \to R^n$, $y(t_0) = y_0$ of the closed loop system (3.3)

$$\|y_0\| \leq r \Rightarrow y(t) \in G_{\overline{d}}, \text{ for all } t \geq t_0 + T(\overline{d}, r).$$

(v) Uniform Stability.

Given any $\overline{d} > \underline{d}$, there is a positive $\delta(\overline{d})$ such that for every solution $y(\cdot) : [t_0, \infty) \to R^n$, $y(t_0) = y_0$ of the closed loop system (3.3)

$$y_0 \in G_{\delta(\overline{d})} \Rightarrow y(t) \in G_{\overline{d}} \text{ for all } t \geq t_0.$$

The following theorem is useful in proving the stability of our controller designs. The proof can be found in [12].

Theorem 3.1. *Consider an uncertain system (3.1) with $\omega \in \Omega$ and suppose that P is a collection of feedback control functions, $p : R \times R^n \to R^m$. If there exists a candidate Lyapunov function $V : R \times R^n \to R_+$ and a class K function $\sigma : R_+ \to R_+$ such that for each $\varepsilon > 0$ there exists $p^\varepsilon \in P$ which assures for all $\omega \in \Omega$*

$$\dot{y}(t) = F(t, y(t), p^\varepsilon(t, y(t)), \omega) \tag{3.4}$$

and

$$\frac{\partial V(t,y)}{\partial t} + \frac{\partial V(t,y)}{\partial y} F(t, y(t), p^\varepsilon(t, y(t)), \omega) \leq -\sigma(\|y\|) + \varepsilon \tag{3.5}$$

for all $t \in R$, $y(t) \in R^n$, then P is a practically stabilizing family with respect to $\{0\}$ where $\{0\}$ is the zero state.

3.2 Model Formulation for Robust Control Design

The dynamic model formulation used in our robust control design is of the parametric form (Spong 1992):

$$Y(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)\phi = \tau \tag{3.6}$$

where ϕ is an m-dimensional vector of unknown physical parameters of the robot manipulator, and Y is an $n \times m$ matrix of known functions of the generalized coordinates and their derivatives. Note that for the time-varying mass system, we can only write the dynamic model in the parametric form when the length of the manipulator links is known. With this assumption, we have

$$Y(\cdot)\phi = Y_0(\cdot)\phi_0 + Y_1(\cdot)\phi_1 + Y_2(\cdot)\phi_2. \tag{3.7}$$

where ϕ_0 is the vector of constant parameters, ϕ_1 consists of the time-varying mass and inertia, and ϕ_2 consists of the first derivatives of the time-varying mass and inertia. Note that $Y_1(\cdot), Y_2(\cdot)$ may consist of not only the joint variables, but also the link lengths. In this formulation, we assume that some components of ϕ are not exactly known, only an estimated value $\hat{\phi}$ is known. The uncertainty is represented by

$$\tilde{\phi} = \hat{\phi} - \phi. \tag{3.8}$$

If ϕ is some bounded function, then the model uncertainty is bounded as

$$\left\|\tilde{\phi}\right\| \leq \rho \qquad \rho > 0$$

3.3 The Proposed Controllers

The controller design problem is as follows: Given the desired trajectory $q_d(t)$, and with some or all of the manipulator parameters being unknown but the bounds are known, derive a robust control law for the actuator torques such that the manipulator output $q(t)$ tracks the desired trajectories and also overcomes the effects of presence of uncertainty in the system. The Lyapunov second method will be used for both robust and adaptive controller design in this chapter. The robust control design technique that we discuss is based on the

theory of guaranteed stability of uncertain systems as developed in Leitmann and Corless (1981).

3.3.1 Robust Control

In this subsection the problem of counteracting model parameter uncertainty is considered. Some different robust control schemes will be discussed. The robust controller is derived in two steps. First, a simple inner loop control law is designed by using the feedback linearization technique to cancel all the known nonlinear terms. The outer loop control law is designed by using the second method of Lyapunov. The Lyapunov-based control laws are more robust than the inverse dynamics based control law because the Lyapunov-based control laws do not rely on the exact cancellation of the system nonlinearities, whereas inverse dynamics based control laws do.

Closed-Loop Dynamics

Consider the following change of the coordinates,

$$\begin{aligned} \dot{\zeta} &= \dot{q}_d - \Lambda e \\ e_v &= \dot{q} - \dot{\zeta} = \dot{e} + \Lambda e \end{aligned} \tag{3.9}$$

where $e = q - q_d$ denotes the tracking error in the manipulator joint coordinates and Λ is a constant, positive definite $(n \times n)$ matrix. The control input is given by $\tau = \tau_0 + \tau_1$, where τ_0 is given by the inner control law and τ_1 by the outer control law. Choose the inner control law as

$$\tau_0 = M_0(q, m, I)\ddot{\zeta} + C_0(q, \dot{q}, m, I)\dot{\zeta} + F_0(q, \dot{m}, \dot{I})\dot{\zeta} + g_0(q, m) - K_D e_v \tag{3.10}$$

where $M_0(\cdot), C_0(\cdot), F_0(\cdot)$ and $g_0(\cdot)$ are the known estimates of the $M(\cdot), C(\cdot), F(\cdot)$ and $g(\cdot)$, respectively, and K_D is the positive definite gain. This control law gives the error equation as

$$M(q, m, I)\dot{e_v} + C(q, \dot{q}, m, I)e_v + \frac{1}{2}F(q, \dot{m}, \dot{I})e_v + (K_D + \frac{1}{2}F(\cdot))e_v = Y(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)\tilde{\phi} + \tau_1 \tag{3.11}$$

where $\tilde{\phi}$ is defined in (3.8) , and $Y(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)$ is given by

$$Y(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)\tilde{\phi} = \tilde{M}(\cdot)\ddot{\zeta} + \tilde{C}(\cdot)\dot{\zeta} + \tilde{F}(\cdot) + \tilde{g}(\cdot) \tag{3.12}$$

where

$$\begin{aligned} \tilde{M}(\cdot) &= M_0(\cdot) - M(\cdot), \\ \tilde{C}(\cdot) &= C_0(\cdot) - C(\cdot), \\ \tilde{F}(\cdot) &= F_0(\cdot) - F(\cdot), \\ \tilde{g}(\cdot) &= g_0(\cdot) - g(\cdot) \end{aligned} \tag{3.13}$$

Note that $Y(\cdot)\tilde{\phi}$ is independent of the joint acceleration. This is important in implementation because the joint acceleration is always not measurable and its response is discontinuous if the saturation type controllers are employed. Also note that the nonlinearity/perturbation term $Y(\cdot)\tilde{\phi}$ is state-dependent, and therefore it cannot be assumed to be a priori bounded by some positive constant. Moreover, the matrix function $Y(\cdot)$ consists of a highly nonlinear term $\tilde{C}(\cdot)$ which causes $Y(\cdot)\tilde{\phi}$ more difficult to deal with in the stability analysis.

Bounding of the Nonlinear Dynamics in $Y(\cdot)\tilde{\phi}$

Suppose that we have only limited knowledge of the parameters, but we do know the structure of the dynamic equation, then it is always possible to find a bounding function for $Y(\cdot)\tilde{\phi}$. Following is a way to find a bounding function for $Y(\cdot)\tilde{\phi}$. From [21], by defining $x = [\ e \quad \dot{e}\ ]^T$, we have

$$\begin{aligned} \|\tilde{M}(q, m, I)\ddot{q}_d\| &\leq \|\tilde{M}(q, m, I)\| \cdot \|\ddot{q}_d\| \leq \beta_1. \\ \|\tilde{M}(q, m, I)\dot{e}\| &\leq \|\tilde{M}(q, m, I)\| \cdot \|\dot{e}\| \leq \beta_2\|x\|. \end{aligned}$$

also

$$\begin{aligned} \|\tilde{C}(q, \dot{q}, m, I)\ddot{q}_d\| &\leq \|\tilde{C}(q, m, I)\| \cdot \|\ddot{q}_d\|^2 + \|\dot{q}_d\| \cdot \|\tilde{C}(q, m, I)\| \cdot \|\dot{e}\| \\ &\leq \beta_3 + \beta_4\|x\|. \\ \|\tilde{C}(q, \dot{q}, m, I)\dot{e}\| &\leq \|q_d\| \cdot \|\tilde{C}(q, m, I)\| \cdot \|e\| + \|e\| \cdot \|\tilde{C}(q, m, I)\| \cdot \|\dot{e}\| \\ &\leq \beta_5 + \beta_6\|x\|^2. \end{aligned}$$

By using the Property 2 from Chapter 2, the time-varying inertia matrix can be bounded as

$$\begin{aligned}\left\|\tilde{F}\left(q,\dot{m},\dot{I}\right)\dot{q}_d\right\| &\leq \left\|\tilde{F}\left(q,\dot{m},\dot{I}\right)\right\|\cdot\|\dot{q}_d\|\leq\beta_7.\\ \left\|\tilde{F}\left(q,\dot{m},\dot{I}\right)e\right\| &\leq \left\|\tilde{F}\left(q,\dot{m},\dot{I}\right)\right\|\cdot\|e\|\leq\beta_8\|x\|.\end{aligned}$$

Also assume that the gravitational effect is bounded as

$$\|\tilde{g}(q,m)\|\leq\beta_9$$

Thus, generally the uncertainty can be bounded by some function such as

$$\left\|Y(\cdot)\tilde{\phi}\right\| = \varsigma_0+\varsigma_1\|x\|+\varsigma_2\|x\|^2. \tag{3.14}$$

where

$$\begin{aligned}\varsigma_0 &= \beta_1+\beta_3+\beta_5+\beta_7+\beta_9,\\ \varsigma_1 &= \beta_2+\beta_4+\beta_8,\\ \varsigma_2 &= \beta_6.\end{aligned}$$

Developing a bounding function is one of the key steps in designing nonlinear robust control law. Existence of these bounding functions is guaranteed by the inherent properties of the robotics system, i.e. mass and inertia of a link is always bounded. However, finding the coefficients of these bounding functions requires a priori knowledge of the uncertainties such as ranges of the parameter variations and maximum variation of load. In most applications, partial knowledge of the uncertainties are always available. However, if the knowledge of the uncertainties are not available, robust control law can still be designed by using a combination of robust control and adaptive control.

Furthermore, if the size of the bounds are not available, then the functional form of the bounding function for robotic systems can still be found because analytical expressions for the robot dynamics are available. Refer to [21], notice that no matter what the size of $\tilde{\phi}$, the following inequality holds:

$$Y(\cdot)\tilde{\phi}\leq\frac{1}{k_r}\left\|\tilde{\phi}\right\|^2+\frac{k_r}{4}\|Y(\cdot)\|^2 \tag{3.15}$$

where gain $k_r > 0$ can be chosen freely by the designer. Since any function that bounds uncertainties can be chosen as the bounding function, we can also define the bounding function $\rho(\cdot)$ as

$$\rho(\cdot) = \frac{1}{k_r}\left\|\tilde{\phi}\right\|^2 + \frac{k_r}{4}\left\|Y(\cdot)\right\|^2.$$

Also notice that if a bound on $\tilde{\phi}$ is known, then the first term is just a constant term, then we can remove the first and a new bounding function for the unknown uncertainties as

$$\rho(\cdot) = \frac{k_r}{4}\left\|Y(\cdot)\right\|^2.$$

This bounding function requires no information about the robot parameters and payload. It is also obvious that the larger the gain k_r, smaller the ultimate bound on the state. In the next section, it is shown that k_r is the key gain to decide whether the controller behaves more robust or adaptive in adaptive robust controller. Since no information about robot parameters and payload is required, it is easy to use during implementation.

Stability Analysis

For the stability analysis, first we have to find a Lyapunov function candidate. Consider the following positive definite function of e_v :

$$V = \frac{1}{2}e_v^T M(q,m,I)e_v. \tag{3.16}$$

Taking the time derivative along the solutions to the error equation gives

$$\begin{aligned}
\dot{V} &= \dot{e_v}^T M(\cdot)e_v + \frac{1}{2}e_v^T\dot{M}(\cdot)e_v \\
&= e_v^T\left(-C(\cdot)e_v - \frac{1}{2}F(\cdot)e_v - K_D e_v + Y(\cdot)\tilde{\phi} + \tau_1\right) + \frac{1}{2}e_v^T\dot{M}(q)e_v \\
&= -e_v^T\left(K_D + \frac{1}{2}F(\cdot)\right)e_v + \frac{1}{2}e_v^T\left(\dot{M}(\cdot) - 2C(\cdot) - F(\cdot)e_v\right) + e_v^T\left(Y(\cdot)\tilde{\phi} + \tau_1\right) \\
&= -e_v^T\left(K_D + \frac{1}{2}F(\cdot)\right)e_v + e_v^T Y(\cdot)\tilde{\phi} + e_v^T\tau_1
\end{aligned} \tag{3.17}$$

Since $F(\cdot)$ is given by

$$F(q,\dot{m},\dot{I}) = \frac{\partial M(q,m,I)}{\partial m}\dot{m} + \frac{\partial M(q,m,I)}{\partial I}\dot{I} \tag{3.18}$$

and the definiteness of $F(\cdot)$ depends on the sign of $\dot{m}$, i.e. $F(\cdot)$ is positive definite when $\dot{m}$ is positive and $F(\cdot)$ is negative definite when $\dot{m}$ is negative. This mean that, when $\dot{m}$ is negative the gain matrix K_D should be chosen such that the matrix $(K_D + \frac{1}{2}F(\cdot))$ is positive definite. By designing different nonlinear compensators, τ_1, both asymptotic and practical stability can be proved. The design of τ_1 (outer control law) will be discussed in detail in the next section.

3.3.2 Nonlinear Compensators Design

In this subsection, by designing the nonlinear compensator, we would like to make the tracking error approach either to a boundary region or to zero asymptotically. For robust controller design, we assume that there exist a known function $\rho(\cdot) : R^n \times R \rightarrow R^l$ such that

$$\left\| Y(\cdot)\tilde{\phi} \right\| \leq \rho^T(e_v, t)$$

for all $(e_v, t) \in R^{2n} \times R$, $\rho^T(e_v, t)$ is a scalar function.

(i) Discontinuous Type Compensator

Theoretically, this compensator will guarantee asymptotic stability of the closed-loop system; however, this compensator appears too forceful due to the compensators' discontinuous nature in practice. Furthermore, this compensator will also excite the high frequency mode and unmodeled dynamics of the system thus resulting in chattering, which is often undesirable. The discontinuous compensator has the form

$$\tau_1 = -\frac{e_v}{\|e_v\|}\rho(\cdot) \tag{3.19}$$

for all $\|e_v\| \neq 0$, otherwise $\tau_1 = 0$. Note that the function is discontinuous at $\|e_v\| = 0$ and this compensator also does not have any boundary layer.

Stability analysis: From above, we have

$$\dot{V} \leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + e_v^T Y(\cdot)\tilde{\phi} + e_v^T \tau_1$$

If $\|e_v\| \neq 0$, then

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v|^2 + e_v^T Y(\cdot)\tilde{\phi} - \frac{e_v^T e_v}{\|e_v\|}\rho(\cdot) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v|^2 + \|e_v\|\|Y(\cdot)\tilde{\phi}\| - \|e_v\|\rho(\cdot) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + \|e_v\|\left(\left\|Y(\cdot)\tilde{\phi}\right\| - \rho(\cdot)\right) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v|^2
\end{aligned}$$

Since V is positive definite and $\dot{V}$ is negative definite, we can conclude that $e_v \to 0$ asymptotically. Also, since $e_v = \dot{e} + \lambda e$, $e, \dot{e} \to 0$ as $t \to 0$. However, chattering will always be introduced due to the discontinuous nature of the control law and it is often undesirable since the high frequency component in the control can excite unmodeled dynamics of the system.

(ii) Saturation Type Compensator

Second type considered is the saturation compensator. This compensator improves the system performance by introducing an error boundary layer. Also the saturation compensator will result in uniform ultimate boundedness, often called practical stability, i.e., the states converge to bounded neighborhood of the origin. But this is inadequate for some application such as high precision manufacturing. The compensator is given by

$$\tau_1 = \begin{cases} -\frac{e_v}{\|e_v\|}\rho(\cdot) & \|e_v\| > \varepsilon \\ -\frac{e_v}{\varepsilon}\rho(\cdot) & \|e_v\| \leq \varepsilon \end{cases} \tag{3.20}$$

where ε is a positive constant, and defines the thickness of the boundary layer.

Stability analysis: From above, if $\|e_v\| > \varepsilon$, then we have

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}(K_D + \frac{1}{2}F(\cdot))\|e_v\|^2 + e_v^T Y(\cdot)\tilde{\phi} - e_v^T\left(\frac{e_v}{\|e_v\|}\rho(\cdot)\right) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + \|e_v\|\|Y(\cdot)\tilde{\phi}\| - \frac{e_v^T e_v}{\|e_v\|}\rho(\cdot) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + \|e_v\|\left(\left\|Y(\cdot)\tilde{\phi}\right\| - \rho(\cdot)\right) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2
\end{aligned}$$

If $\|e_v\| \leq \varepsilon$, then

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + e_v^T Y(\cdot)\tilde{\phi} - e_v^T\left(\frac{e_v}{\varepsilon}\rho(\cdot)\right) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + \|e_v\|\|Y(\cdot)\tilde{\phi}\| + \frac{\|e_v\|^2}{\varepsilon}\rho(\cdot) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + \|e_v\|\rho(\cdot) + \|e_v\|\rho(\cdot) \\
&\leq -\lambda_{\min}\left(K_D\right)\|e_v\|^2 + 2\varepsilon\rho(\cdot)
\end{aligned}$$

Using Theorem 1, uniform ultimate boundedness of the system state is guaranteed. Leitmann and Corless (1981) showed that there exist a time t_ε which the solution trajectory intersects with the boundary layer. However, it is not guarantee that the solution trajectory will remain inside the layer. Therefore if ε is not not chosen properly, switching in and out of the layer may happen and introduce chattering.

(iii) Smooth Type Compensator

Third type of compensator considered is the smooth robust nonlinear compensator. There will be no switching of controller during the process, hence the motion is expected to be more smooth than the saturation compensator. The outer loop control for this type of compensator is time-varying, i.e. explicitly depends on time. The controller has the form

$$\tau_1 = -\rho(\cdot)\tanh[(a+bt)e_v] \tag{3.21}$$

where a, b are positive constants, and $\tanh[(a+bt)e_v] = [\tanh[(a+bt)e_{v1}], \cdots, \tanh[(a+bt)e_{vn}]^T$.

Stability analysis: From above, we have

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 + e_v^T Y(\cdot)\tilde{\phi} - e_v^T\rho(\cdot)\tanh\left[(a+bt)\,e_v\right] \\
&= -\lambda_{\min}(K_D + \frac{1}{2}F(\cdot))\|e_v\|^2 + \|e_v\|\|Y(\cdot)\tilde{\phi}\| - \rho\sum_{i=1}^{n} e_v^T(\cdot)\tanh[(a+bt)e_v]
\end{aligned}$$

Since $\theta \tanh(\alpha\theta) \geq 0$ for all θ and $\alpha > 0$, and also $|\tanh(\theta)| = \tanh(|\theta|)$, then we have

$$\begin{aligned}
\dot{V} &\leq -\lambda_{\min}(K_D + \frac{1}{2}F(\cdot))\|e_v\|^2 + \|e_v\|\|Y(\cdot)\tilde{\phi}\| - \rho e_{vi}^T(\cdot)\tanh[(a+bt)e_v] \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 - \rho(\cdot)\left(\sum_{i=1}^{n}|e_{vi}|\tanh\left[(a+bt)|e_{vi}|\right] - \frac{\left\|Y(\cdot)\tilde{\phi}\right\|}{\rho(\cdot)}\|e_v\|\right) \\
&\leq -\lambda_{\min}\left(K_D + \frac{1}{2}F(\cdot)\right)\|e_v\|^2 - \rho(\cdot)\left(\sum_{i=1}^{n}|e_{vi}|\tanh\left[(a+bt)|e_{vi}|\right] - \alpha_0\sum_{i=1}^{n}|e_{vi}|\right)
\end{aligned}$$

where $\alpha_0 = \frac{\|Y(\cdot)\tilde{\phi}\|}{\rho(\cdot)}$; and $0 \leq \alpha_0 < 1$. Then it is obvious that $\dot{V} \leq 0$ as long as the following condition is satisfied.

$$\sum_{i=1}^{n}|e_{vi}|\tanh\left[(a+bt)|e_{vi}|\right] - \alpha_0\sum_{i=1}^{n}|e_{vi}| \geq 0$$

$$\Rightarrow |e_{vi}|\tanh\left[(a+bt)|e_{vi}|\right] - \alpha_0|e_{vi}| \geq 0$$

Solving the inequality, we have

$$|e_{vi}| \geq \gamma_0 \tag{3.22}$$

where

$$\gamma_0 = \frac{1}{2(a+bt)}\ln\left(\frac{1+\alpha_0}{1-\alpha_0}\right).$$

If (3.22) is satisfied, then e_v will converge to a the closed ball centered at the origin with radius of γ_0. Furthermore, as t approaches to infinity, γ_0 will also approach to 0. Thus as $t \to \infty$, $\dot{V} < 0$ and $\dot{V} = 0$ when $e_v = 0$. Hence we can conclude that e_v asymptotically converges to zero.

3.3.3 Adaptive Robust Control

The nonlinear compensators that we discussed above require a priori knowledge of the bound of uncertainty. However, the upper bound of the time-varying unknown parameter may not be known or if the range of the time-varying unknown parameter is chosen large will result in poor performance. With these reasons, we propose an adaptive robust controller which does require prior knowledge of the bound of uncertainty and guarantees uniform ultimate

boundedness of the system state. The different between the adaptive robust and adaptive controller is the adaptive robust controller adapts the upper bound of the unknown parameters instead of the unknown parameters, which adaptive controller does. The following theorem gives the outer loop controller and its stability.

Theorem 3.2. *: Let $\varepsilon > 0$ and choose the outer control law and estimate $\hat{\rho}_1$ of ρ_1 as*

$$\tau_1 = \begin{cases} -\frac{e_v}{\|e_v\|}\left(\frac{1}{k_r}\hat{\rho}_1(\cdot) + \rho_2(\cdot)\right) & \|e_v\| > \varepsilon \\ -\frac{e_v}{\varepsilon}\left(\frac{1}{k_r}\hat{\rho}_1(\cdot) + \rho_2(\cdot)\right) & \|e_v\| \leq \varepsilon \end{cases} \tag{3.23}$$

and

$$\dot{\hat{\rho}}_1 = \begin{cases} \frac{1}{k_r}\Gamma\|e_v\| & \|e_v\| > \varepsilon \\ 0 & \|e_v\| \leq \varepsilon \end{cases} \tag{3.24}$$

where ρ_1 and ρ_2 are defined by

$$\begin{aligned} \left\|\tilde{\phi}\right\|^2 &\leq \rho_1(\cdot) \\ \frac{k_r}{4}\|Y(\cdot)\|^2 &\leq \rho_2(\cdot) \end{aligned}$$

then, practical stability of the closed-loop system can be guaranteed.

Proof. Choose the Lyapunov function as

$$V = \frac{1}{2}e_v^T M(q,m,I)e_v + \frac{1}{2}\tilde{\rho}_1^T\Gamma^{-1}\tilde{\rho}_1. \tag{3.25}$$

where Γ is a positive definite constant and $\tilde{\rho}_1 = \rho_1 - \hat{\rho}_1$. Taking the time derivative along the solutions to the error equation gives

$$\begin{aligned} \dot{V} &= -e_v^T\left(K_D + \frac{1}{2}F(\cdot)\right)e_v + \frac{1}{2}e_v^T\left(\dot{M}(\cdot) - 2C(\cdot) - F(\cdot)\right)e_v + e_v^T\left(Y^T(\cdot)\tilde{\phi} + \tau_1\right) + \tilde{\rho}_1^T\Gamma^{-1}\dot{\tilde{\rho}}_1 \\ &= -e_v^T\left(K_D + \frac{1}{2}F(\cdot)\right)e_v + e_v^T\left(Y^T(\cdot)\tilde{\phi} + \tau_1\right) + \tilde{\rho}_1^T\Gamma^{-1}\dot{\tilde{\rho}}_1 \end{aligned}$$

From equation (3.15) we obtain

$$\begin{aligned} Y(\cdot)\tilde{\phi} &\leq \frac{1}{k_r}\left\|\tilde{\phi}\right\|^2 + \frac{k_r}{4}\|Y(\cdot)\|^2 \\ &\leq \frac{1}{k_r}\rho_1 + \rho_2. \end{aligned}$$

then

$$\dot{V} \leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + e_v^T \left(\frac{1}{k_r} \left\| \tilde{\phi} \right\|^2 + \frac{k_r}{4} \left\| Y(\cdot) \right\|^2 + \tau_1 \right) + \tilde{\rho_1}^T \Gamma^{-1} \dot{\tilde{\rho}}_1$$

If $\|e_v\| > \varepsilon$, then

$$\begin{aligned}
\dot{V} &\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + \|e_v\| \left(\frac{1}{k_r} \left\| \tilde{\phi} \right\|^2 + \frac{k_r}{4} \left\| Y(\cdot) \right\|^2 \right) - e_v^T \left(\frac{e_v}{\|e_v\|} \left(\frac{1}{k_r} \rho_1(\cdot) + \rho_2(\cdot) \right) \right) \\
&\quad + \tilde{\rho}_1^T \Gamma^{-1} \dot{\tilde{\rho}}_1 \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + \|e_v\| \left(\frac{1}{k_r} \rho_1(\cdot) + \rho_2(\cdot) \right) - \frac{e_v^T e_v}{\|e_v\|} \frac{1}{k_r} \rho_1(\cdot) - \frac{e_v^T e_v}{\|e_v\|} \rho_2(\cdot) - \tilde{\rho}_1^T \Gamma^{-1} \dot{\tilde{\rho}}_1 \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + \|e_v\| \left(\frac{1}{k_r} \rho_2(\cdot) + \rho_2(\cdot) \right) - \frac{\|e_v\|}{k_r} \rho_1(\cdot) - \|e_v\| \rho_2(\cdot) - \tilde{\rho}_1^T \Gamma^{-1} \dot{\tilde{\rho}}_1 \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + \frac{\|e_v\|}{k_r} \tilde{\rho}_1 - \tilde{\rho}_1^T \Gamma^{-1} \dot{\tilde{\rho}}_1 \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + \tilde{\rho}_1^T \left(\frac{\|e_v\|}{k_r} - \Gamma^{-1} \dot{\tilde{\rho}}_1 \right) \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v
\end{aligned}$$

If $\|e_v\| \leq \varepsilon$, then

$$\begin{aligned}
\dot{V} &\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + e_v^T \left(\frac{1}{k_r} \rho_1(\cdot) + \rho_2(\cdot) - \frac{e_v}{\varepsilon} \left(\frac{1}{k_r} \hat{\rho}_1(\cdot) + \rho_2(\cdot) \right) \right) \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + \|e_v\| \left(\frac{2}{k_r} \rho_2(\cdot) + 2\rho_2(\cdot) \right) \\
&\leq -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + 2\varepsilon \rho_2(\cdot) + \frac{2}{k_r} \varepsilon \rho_1(\cdot)
\end{aligned}$$

In oder have the $\dot{V} \leq 0$, the following needed to satisfied:

$$\left\| K_D - \frac{1}{2} F(\cdot) \right\| \geq 0 \tag{3.26}$$

For (3.22), we have

$$\begin{aligned}
\|K_D\| &\geq \left\| \frac{1}{2} F(\cdot) \right\| \\
\lambda_M \{K_D\} &\geq \frac{1}{2} k_F \|q\|
\end{aligned} \tag{3.27}$$

and the adaptation law

$$\dot{\hat{\rho}}_1 = \begin{cases} \frac{1}{k_r} \Gamma \|e_v\| & \|e_v\| > \varepsilon \\ 0 & \|e_v\| \leq \varepsilon \end{cases} \tag{3.28}$$

3.3.4 Adaptive Control

Since the linear parametrization property holds ever for the case of the time-varying masses, it is possible to design an adaptive control law for the system. All the results using this control law are based on the assumption that the unknown time-varying parameter ϕ_1 is continuous and its n-th derivative is constant. From (3.7), we know that the uncertainty of the time-varying masses system can be written as

$$Y(\cdot)\phi = Y_0(\cdot)\phi_0 + Y_1(\cdot)\phi_1 + Y_2(\cdot)\phi_2. \tag{3.29}$$

First we consider a simple case that the mass flow rate is constant, i.e. $\frac{d\phi_2}{dt} = 0$, and $\frac{d\phi_1}{dt} = \phi_2$, then we have

$$\phi_1 = k_1 t + k_0.$$

(3.29) can be written as

$$\begin{aligned} Y(\cdot)\phi &= Y_0(\cdot)\phi_0 + Y_1(\cdot)\left(k_1 t + k_0\right) + Y_2(\cdot)k_1 \\ &= Y_0(\cdot)\phi_0 + Y_1(\cdot)tk_1 + Y_2(\cdot)k_1 + Y_1(\cdot)k_0 \\ &= Y_0(\cdot)\phi_0 + \left(Y_1(\cdot)t + Y_2(\cdot)\right)k_1 + Y_1(\cdot)k_0 \end{aligned} \tag{3.30}$$

Notice that ϕ_0, k_1, k_0 are all constant parameters. Assuming that the constant parameters of the robot are know, i.e. ϕ_0 is known and the time-varying parameters ϕ_1 is unknown, i.e. k_0 and k_1 are unknown, then choose the control law

$$\tau = \hat{M}(\cdot)\ddot{\zeta} + \hat{C}(\cdot)\dot{\zeta} + \hat{F}(\cdot)\dot{\zeta} + \hat{g}(\cdot) - K_D e_v \tag{3.31}$$

with the adaptation law

$$\begin{aligned} \dot{\hat{k}}_1 &= -\Gamma_1 \left(Y_1(\cdot)t + Y_2(\cdot)\right)^T e_v \\ \dot{\hat{k}}_0 &= -\Gamma_0 Y_1(\cdot)^T e_v \end{aligned}$$

can guarantee asymptotic stability of the closed loop system. The following theorem gives a general version of the control law we discussed.

Theorem 3.3. *Assume that the time-varying function ϕ_1 is continuous and has finite number of non-zero derivatives, i.e. there exist some vector constants $k_0, k_1, \cdots, k_n$ such that*

$$\phi_1 \equiv k_0 + k_1 t + k_2 t^2 + \cdots + k_n t^n \tag{3.32}$$

then with inner control law

$$\tau = \hat{M}(\cdot)\ddot{\zeta} + \hat{C}(\cdot)\dot{\zeta} + \hat{F}(\cdot)\dot{\zeta} + \hat{g}(\cdot) - K_D e_v \tag{3.33}$$

and adaptation law

$$\begin{aligned}
\dot{\hat{k}}_0 &= -\Gamma_0 Y_1(\cdot)^T e_v; \\
\dot{\hat{k}}_1 &= -\Gamma_1 \left(Y_1(\cdot)t + Y_2(\cdot)\right)^T e_v; \\
&\vdots \\
\dot{\hat{k}}_{n-1} &= -\Gamma_{n-1} \left(Y_1(\cdot)t + (n-1)Y_2(\cdot)\right)^T t^{n-2} e_v; \\
\dot{\hat{k}}_n &= -\Gamma_n \left(Y_1(\cdot)t + nY_2(\cdot)\right)^T t^{n-1} e_v
\end{aligned} \tag{3.34}$$

will result in a globally asymptotically stable closed loop system.

Proof. From (3.32) we know that

$$\phi_1 = k_0 + k_1 t + k_2 t^2 + \cdots + k_n t^n$$

and

$$\phi_2 = \dot{\phi}_1 = k_1 + 2k_2 t + \cdots + nk_n t^{n-1}$$

then we have

$$\begin{aligned}
Y_1(\cdot)\phi_1 + Y_2(\cdot)\phi_2 &= Y_1(\cdot)\left(k_0 + k_1 t + k_2 t^2 + \cdots + k_n t^n\right) \\
&\quad + Y_2(\cdot)\left(k_1 + 2k_2 t + \cdots + nk_n t^{n-1}\right) \\
&= Y_1(\cdot)k_0 + (Y_1(\cdot)t + Y_2(\cdot))k_1 + (Y_1(\cdot)t + Y_2(\cdot))tk_2 + \cdots \\
&\quad + (Y_1(\cdot)t + nY_2(\cdot))t^{n-1}k_n
\end{aligned}$$

with this parametrization, the global asymptotic stability of the closed loop system can be easily shown.

3.3.5 Integral Saturation Adaptive Robust Controller

To improve the robustness of the control law, a boundary layer is introduced in adaptive robust control. However, with this only uniform ultimate boundedness of the closed loop error can be guaranteed. In addition, chattering may be introduced if the value of ϵ is not properly chosen. In this subsection, an integral term is introduced when the solution enters the layer and global asymptotically stable can be shown.

Theorem 3.4. *: Consider the following outer loop control law and an update law for ρ as*

$$\tau_1 = \begin{cases} -\frac{e_v}{\|e_v\|}\hat{\rho} & t \leq t_\varepsilon \\ -\frac{e_v}{\varepsilon}\hat{\rho}_f - K_I \int e_v dt & t > t_\varepsilon \end{cases} \tag{3.35}$$

and

$$\dot{\hat{\rho}} = \begin{cases} \frac{1}{k_\tau}\Gamma\|e_v\| & \|e_v\| > \varepsilon \\ 0 & \|e_v\| \leq \varepsilon \end{cases} \tag{3.36}$$

where ρ is given by

$$\|Y(\cdot)\tilde{\phi}\| \leq \rho(\cdot)$$

Then, the closed-loop system is global asymptotically stable. Note that $\hat{\rho}_f$ is the constant limit of $\hat{\rho}$. Here we assume that $\hat{\rho}$ is updated only when $t \leq t_\varepsilon$.

Proof. In [43], the existence of t_ε has been shown. The total time spent by the solution trajectory outside the layer is finite. Moreover, it is assumed that the upper bound estimate $\hat{\rho}$ converges to a constant finite value $\hat{\rho}_f$ before t approaches t_ε. Choose Lyapunov candidate as

$$V = \frac{1}{2}e_v^T M(q,m,I)e_v + \frac{1}{2}\tilde{\rho}^T\Gamma^{-1}\tilde{\rho}.$$

From Section (3.3.3), for $t \leq t_\varepsilon$, we get

$$\begin{aligned} \dot{V} &\leq -e_v^T\left(K_D + \frac{1}{2}\tilde{F}(\cdot)\right)e_v \\ &\leq 0. \end{aligned}$$

For $t > t_\varepsilon$, we can choose the Lyapunov function as

$$V = \frac{1}{2} e_v^T M(q, m, I) e_v + \frac{1}{2} \tilde{\rho}^T \Gamma^{-1} \tilde{\rho} + \frac{1}{2} \psi^T K_I \psi \tag{3.37}$$

where

$$\begin{aligned} \psi &= K_I^{-1} Y(\cdot) \tilde{\phi} - \int_{t_\epsilon}^{t} e_v dt \\ \dot{\psi} &= -e_v \end{aligned} \tag{3.38}$$

where K_I is the positive gain matrix. Then the time derivative of the Lyapunov function is

$$\begin{aligned} \dot{V} &= -e_v^T M(q, m, I) \dot{e_v} + \tilde{\rho}^T \Gamma^{-1} \dot{\tilde{\rho}} + \frac{1}{2} \psi^T K_I \dot{\psi} \\ &= -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + e_v^T \left(Y^T(\cdot) \tilde{\phi} - \frac{e_v}{\varepsilon} \hat{\rho}_f - K_I \int_{t_\epsilon}^{t} e_v dt \right) + \psi^T K_I \dot{\psi} \\ &= -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + e_v^T \left(Y^T(\cdot) \tilde{\phi} - \frac{e_v}{\varepsilon} \hat{\rho}_f - K_I \int_{t_\epsilon}^{t} e_v dt \right) + \dot{\psi}^T K_I \psi \\ &= -e_v^T \left(K_D + \frac{1}{2} F(\cdot) \right) e_v + e_v^T \left(Y^T(\cdot) \tilde{\phi} - \frac{e_v}{\varepsilon} \hat{\rho}_f - K_I \int_{t_\epsilon}^{t} e_v dt \right) \\ &\quad -e_v^T K_I \left(K_I^{-1} Y(\cdot) \tilde{\phi} - \int_{t_\epsilon}^{t} e_v dt \right) \\ &\leq -e_v^T \left(K_D + \frac{1}{2} \tilde{F}(\cdot) \right) e_v - \frac{\hat{\rho}_f}{\varepsilon} \|e_v\|^2 . \end{aligned} \tag{3.39}$$

As $t > t_\varepsilon$, then the estimated parameter $\hat{\rho}$ will converges to finite value $\hat{\rho}_f$, so $\dot{\psi} = e_v$ is true only when $t > t_\varepsilon$. Since $\dot{V} \leq 0$ as $t > t_\varepsilon$, the solution trajectory will remain inside the layer in the absence of any disturbances.

In this chapter, several control laws, which are combinations of robust and adaptive approaches were discussed. However, the ultimate justification of value and applicability of these control algorithms lie in actually implementing them on real systems. Therefore, the designed controllers will be applied to the NSK two-link direct drive manipulator in the next chapter.

Chapter 4

EXPERIMENTAL BACKGROUND

In this chapter we describe the implementation of the proposed controllers on a direct drive NSK two-link manipulator. First, the experimental setup is described. Then, the dynamic equations for the NSK two-link manipulator with a time-varying payload are given and their properties are studied. Parameterization of the robot dynamics is illustrated.

4.1 Hardware Setup

A detailed drawing of the experimental facility is shown in Figure 4.1. The experimental setup consists of a two-degree-freedom direct drive manipulator with a vessel sitting on the end of the second link. A pipe is connected from the top of vessel to a pump either to pump fluid in or out of the vessel. Pumping of fluid in or out of the vessel during the motion of robot gives the time varying nature for the payload.

The two-degree-freedom manipulator is driven by two direct drive switched reluctance type NSK motors. The base motor (model 1410) and elbow motor (model 608) have a maximum rated torque of 245 N-m and 39.2 N-m, respectively. Details of the motor specifications are shown in Table 4.1. Sensors for both position and velocity measurement are integrated within each motor, which provide measurement of joint position and joint velocity. The actuator position of each link is measured with a 150-pole resolver, which provides approximately a resolution of 2 arc-seconds. The analog position signal is processed through a 10-bit resolver to a digital converter that provides $150 \times 1024 = 153{,}600$ counts per resolution . This gives a resolution of 0.0000409 radians per encoder counter. A velocity signal is also

available through frequency-through-voltage converter that provides an analog signal that is proportional to joint velocity. However, due to noise, joint velocities used in the experiment were calculated from the joint position by using first order finite difference method. Constant parameters of the NSK manipulator are listed in Table 4.3.

A servo sampling rate of two milli-seconds is used in the implementation. Also "torque mode" is chosen as the operator mode for the NSK motors. Under this mode, the motors behave like current amplifiers which produce a motor torque command that is proportional to the input voltage signal.

The time-varying payload is modeled by a 4" inner diameter PVC pipe. It is closed on both ends by two PVC caps with 4" inner diameter. A 6.25" diameter and 0.75" thick aluminum mount is built to mount the vessel on the second link of the robot. The position of the vessel is fixed by using three 5" screws which connect the mount and the aluminum ring. The length of the vessel is 16", so it provides an approximate volume of 201.06 in^3. The working fluid used in our experiment is water and its density is 998.2 kg/m^3 under room temperature. So the mass of the payload varies from 0 kg to 3.289 kg or 7.253 lb. A garden pump with 30 gallons/ hour is used to pump water in or out of the vessel. A 3/8" inner diameter pipe is used to connect the vessel to the pump which provides approximately 0.206 kg/s or 3.289 lb/s flow rate. Pumping of water in or out can be done by switching the inlet and outlet of the pump to the pipe. The parameters of the vessel are given in Table 4.2.

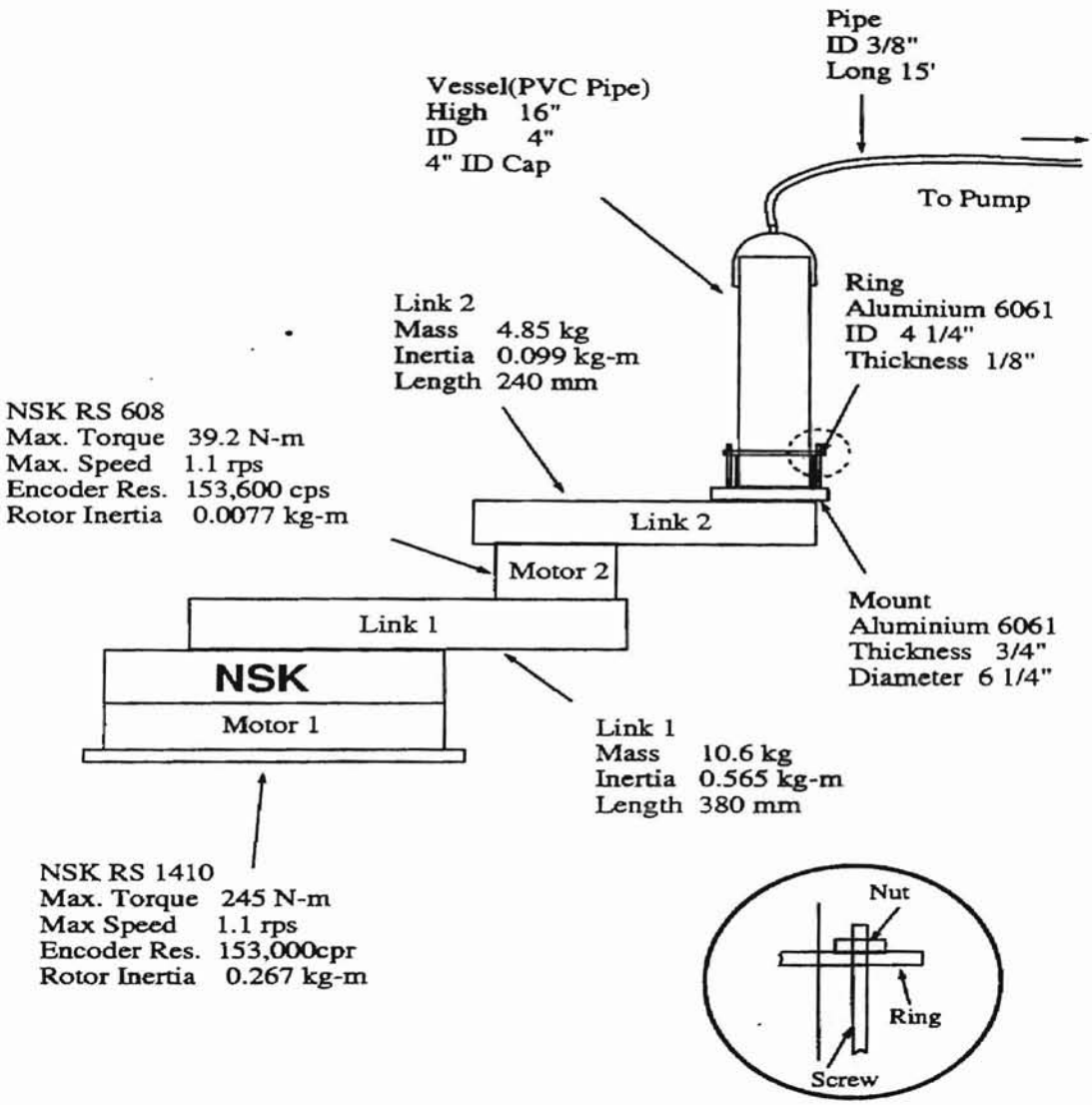

Figure 4.1: Experimental Setup

	Model 1410	Model 608
Maximum Torque (N-m)	245.0	39.2
Rotor Inertia (Kg-m^2)	0.267	0.0077
Maximum Speed (rps)	1.1	1.1
Mass (Kg)	71	13
Resolver Resolution (Counts/rev)	153,600	153,600
Bearing Friction Torque (N-m)	7.84	2.94

Table 4.1: Motor Specifications

Symbol	SI Value	Definition
ρ_w	998.2	Density of water
R_o	0.0508	Outer radius
R_i	0.05	Inner radius
$\dot{m}_p$	0.206	Water flow rate
m_{p1}	1.814	Mass of the vessel
I_{p1}	0.0023	Inertia of the vessel

Table 4.2: Parameters for the vessel and liquid

Symbol	SI Value	Definition
I_1	0.2675	Motor 1 rotor inertia
I_2	0.360	Arm 1 inertia (about c.g.)
I_3	0.0077	Motor 2 rotor inertia
I_{3c}	0.040	Motor 1 stator inertia
I_4	0.051	Arm 2 inertia (about c.g.)
m_1	73.0	Motor 1 mass
m_2	10.6	Arm 1 mass
m_3	12	Motor 2 mass
m_4	4.85	Arm 2 mass
l_1	0.38	Arm 1 length
l_2	0.24	Arm 2 length
l_3	0.139	Arm 1 radius of gyration
l_4	0.099	Arm 2 radius of gyration

Table 4.3: Constant Parameters for NSK Robot Manipulator

$$
\begin{aligned}
m_{12} &= I_3 + I_4 + I_p(t) + l_4^2 m_4 + l_2^2 m_p(t) + (l_1 l_4 m_4 + l_1 l_2 m_p(t)) \cos(q_2) \\
&= \left[I_3 + I_4 + l_4^2 m_4\right] + \left[l_1 l_4 m_4\right] \cos(q_2) + \left[l_2^2 + l_1 l_2 \cos(q_2)\right] m_p(t) + I_p(t) \\
&\equiv p_3 + p_2 \cos(q_2) + v_2(q_2) m_p(t) + I_p(t) \\
&= m_{21} \\
m_{22} &= I_3 + I_4 + I_p + l_4^2 m_4 + l_2^2 m_p \\
&= \left[I_3 + I_4 + l_4^2 m_4\right] + l_2^2 m_p + I_p(t) \\
&\equiv p_3 + v_3 m_p(t) + I_p(t)
\end{aligned}
$$

where the p_i represents the constant inertia parameters and v_i is the variable inertia parameters. Also note that the variable inertia parameters are always functions of the joint variables except for the last link. Next we can compute the Christoffel symbols as follows:

$$
\begin{aligned}
c_{111} &= \frac{1}{2}\frac{\partial d_{11}}{\partial q_1} = 0 \\
c_{121} &= c_{211} = \frac{1}{2}\frac{\partial d_{11}}{\partial q_2} = -\left(p_2 + l_1 l_2 m_p(t)\right) \sin(q_2) \\
c_{221} &= \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = -\left(p_2 + l_1 l_2 m_p(t)\right) \sin(q_2) \\
c_{112} &= \frac{\partial d_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial d_{11}}{\partial q_2} = \left(p_2 + l_1 l_2 m_p(t)\right) \sin(q_2) \\
c_{122} &= c_{212} = \frac{1}{2}\frac{\partial d_{22}}{\partial q_1} = 0 \\
c_{222} &= \frac{1}{2}\frac{\partial d_{22}}{\partial q_2} = 0.
\end{aligned}
$$

By defining $p_7(m_p, q) = (p_2 + l_1 l_2 m_p(t)) \sin(q_2)$, then Coriolis vector is

$$
\begin{aligned}
c(q, \dot{q}, m_p, I_p)\dot{q} &= \begin{bmatrix} -2p_7 \dot{q}_1 \dot{q}_2 - p_7 \dot{q}_2^2 \\ p_7 \dot{q}_1^2 \end{bmatrix} = \begin{bmatrix} -p_7 \left(2\dot{q}_1 + \dot{q}_2\right) \dot{q}_2 \\ p_7 \dot{q}_1^2 \end{bmatrix} \\
&= \begin{bmatrix} -\dot{q}_2 p_7 & -\left(\dot{q}_1 + \dot{q}_2\right) p_7 \\ \dot{q}_1 p_7 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}.
\end{aligned}
$$

Hence

$$
C(q, \dot{q}, m_p, I_p) = -\begin{bmatrix} \dot{q}_2 p_7 & \left(\dot{q}_1 + \dot{q}_2\right) p_7 \\ -\dot{q}_1 p_7 & 0 \end{bmatrix} \tag{4.3}
$$

For simplicity, $C(q, \dot{q}, m_p, I_p)$ can be written in terms of parameter p_7, which is a function of both joint variables and time-varying mass. From the definition, time-varying mass matrix can be written as

$$\begin{aligned} F(q, \dot{m}_p, \dot{I}_p) &= \frac{\partial M}{\partial m_p}\dot{m}_p + \frac{\partial M}{\partial I_p}\dot{I}_p \\ &= \begin{bmatrix} v_1(q_2) & v_2(q_2) \\ v_2(q_2) & v_3 \end{bmatrix} \dot{m}_p + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \dot{I}_p. \end{aligned} \tag{4.4}$$

Notice that the time-varying matrix depends only on the variable inertia parameters and not constant inertia parameters. This is because the variable inertia parameters are only associated with the time-varying mass.

Skew-Symmetry of $(\dot{M} - 2C - F)$

In this subsection, the skew-symmetry property is checked for a two-link robot manipulator carrying a time-varying payload at the end of the second link. From Chapter 2, we prove that $(\dot{M} - 2C - F)$ is a skew-symmetry matrix. To verify this property for the two-link manipulator, the following calculations are performed. First, we calculate the derivative of the inertia matrix

$$\dot{M}(q, m_p, I_p) = \begin{bmatrix} \dot{m}_{11} & \dot{m}_{12} \\ \dot{m}_{21} & \dot{m}_{22} \end{bmatrix}$$

where

$$\begin{aligned} \dot{m}_{11} &= -2p_2 \sin(q_2)\dot{q}_2 - 2l_1 l_2 \sin(q_2)\dot{q}_2 + v_1(q_2)\dot{m}_p(t) + \dot{I}_p(t) \\ &= -2(p_2 + l_1 l_2 \dot{m}_p(t)) \sin(q_2)\dot{q}_2 + v_1(q_2)\dot{m}_p(t) + \dot{I}_p(t) \\ &= -2p_7\dot{q}_2 + v_1(q_2)\dot{m}_p(t) + \dot{I}_p(t) \\ \dot{m}_{12} &= -p_2 \sin(q_2)\dot{q}_2 - l_1 l_2 \sin(q_2)\dot{q}_2 m_p(t) + v_2(q_2)\dot{m}_p(t) + \dot{I}_p(t) \\ &= -(p_2 + l_1 l_2 m_p(t)) \sin(q_2)\dot{q}_2 + v_2(q_2)\dot{m}_p(t) + \dot{I}_p(t) \\ &= -p_7\dot{q}_2 + v_2(q_2)\dot{m}_p(t) + \dot{I}_p(t) \\ &= \dot{m}_{21} \\ \dot{m}_{22} &= v_3\dot{m}_p(t) + \dot{I}_p(t) \end{aligned}$$

note that, $\dot{M}$ can be written as

$$\dot{M}(\cdot) = -\begin{bmatrix} 2p_7\dot{q}_2 & p_7\dot{q}_2 \\ p_7\dot{q}_2 & 0 \end{bmatrix} + \begin{bmatrix} v_1(q_2) & v_2(q_2) \\ v_2(q_2) & v_3 \end{bmatrix} \dot{m}_p(t) + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \dot{I} \tag{4.5}$$

We can write $S(\cdot) = \dot{M}(\cdot) - 2C(\cdot) - F(\cdot)$ as

$$\begin{aligned} S(\cdot) &= -\begin{bmatrix} 2p_7\dot{q}_2 & p_7\dot{q}_2 \\ p_7\dot{q}_2 & 0 \end{bmatrix} + 2\begin{bmatrix} \dot{q}_2 p_7 & (\dot{q}_1 + \dot{q}_2)\, p_7 \\ -\dot{q}_1 p_7 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & (2\dot{q}_1 + \dot{q}_2)\, p_7 \\ -(2\dot{q}_1 + \dot{q}_2)\, p_7 & 0 \end{bmatrix} \end{aligned}$$

Furthermore, since $S^T(\cdot) + S(\cdot) = 0$, we can conclude that $\dot{M}(\cdot) - 2C(\cdot) - F(\cdot)$ is skew-symmetry matrix.

Linearity in the Unknown Parameters

Linearity in the parameters is an important property of robot manipulators that is used in both robust control and adaptive control designs. This property can be expressed as

$$M(q, m_p, I_p)\ddot{q} + C(q, \dot{q}, m_p, I_p)\dot{q} + F(q, \dot{m}_p, \dot{I}_p)\dot{q} = Y(\cdot)\phi \tag{4.6}$$

where

$$\begin{aligned} Y(\cdot) &= \begin{bmatrix} v_{11} & v_{12} & v_{13} & w_{11} & w_{13} & w_{12} & w_{14} \\ v_{21} & v_{22} & v_{23} & w_{21} & w_{23} & w_{22} & w_{24} \end{bmatrix} \\ \phi &= \begin{bmatrix} p_1 & p_2 & p_3 & m_p & I_p & \dot{m}_p & \dot{I}_p \end{bmatrix}^T \end{aligned}$$

and

$$
\begin{aligned}
v_{11} &= \ddot{q}_1 \\
v_{12} &= 2\cos(q_2)(\ddot{q}_1 + \ddot{q}_2) - (2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)\sin(q_2) \\
v_{13} &= \ddot{q}_2 \\
v_{21} &= 0 \\
v_{22} &= \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \\
v_{23} &= \ddot{q}_1 + \ddot{q}_2 \\
w_{11} &= v_1(q_2)\ddot{q}_1 + v_2(q_2)\ddot{q}_2 - \sin(q_2)\left(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2\right) l_1 l_2 \\
w_{12} &= v_1(q_2)\dot{q}_1 + v_2(q_2)\dot{q}_2 + l_1 l_2 \sin(q_2)\dot{q}_1^2 \\
w_{13} &= \ddot{q}_1 + \ddot{q}_2 \\
w_{14} &= \dot{q}_1 + \dot{q}_2 \\
w_{21} &= v_2(q_2)\ddot{q}_1 + v_3\ddot{q}_2) \\
w_{22} &= v_2(q_2)\dot{q}_1 + v_3\dot{q}_2 \\
w_{23} &= \ddot{q}_1 + \ddot{q}_2 \\
w_{24} &= \dot{q}_1 + \dot{q}_2
\end{aligned}
$$

Both constant and time-varying inertia parameters are listed in Table 4.4 and Table 4.5 separately. Furthermore, $Y(\cdot)$ can also be decomposed to associate with constant parameters and time-varying parameters

$$Y(\cdot)\phi = Y_0(\cdot)\phi_0 + Y_1(\cdot)\phi_1 + Y_2(\cdot)\phi_2, \tag{4.7}$$

where

$$
\begin{aligned}
\phi_1 &= \begin{bmatrix} m_p & I_p \end{bmatrix}^T, \\
\phi_2 &= \begin{bmatrix} \dot{m}_p & \dot{I}_p \end{bmatrix}^T,
\end{aligned}
$$

Symbol	SI Value	Definition
p_1	4.2420	Constant inertia parameter 1
p_2	0.1825	Constant inertia parameter 2
p_3	0.1062	Constant inertia parameter 3

Table 4.4: Constant Inertia Parameters

Symbol	SI Value	Definition
v_1	$0.202 + 0.1824\cos(q_2)$	Time-varying inertia parameter 1
v_2	$0.0576 + 0.0912\cos(q_2)$	Time-varying inertia parameter 2
v_3	0.0576	Time-varying inertia parameter 3

Table 4.5: Time-varying Inertia Parameters

Robot Parameters

The time-varying payload mass can be written as

$$m_p = m_{p1} + m_{p2}. \tag{4.8}$$

where m_{p1} is mass of the vessel and m_{p2} is the mass of the liquid inside the vessel. Also

$$I_p = I_{p1} + I_{p2}. \tag{4.9}$$

where I_{p1} and I_{p2} are inertia of the vessel and inertia of the liquid inside the vessel about the center gravity, respectively. The inertia of the cylindrical vessel can be written as

$$\begin{aligned} I_{p1} &= I_{z_{p1}} = \int_m dI_{z_{p1}} \\ &= \int_{R_i}^{R_o} \pi\rho_1 h_1 r^3 dr \\ &= \frac{1}{2}\pi\rho_1 h_1 \left(R_0^4 - R_i^4\right) \\ &= \frac{1}{2}\pi\rho_1 h_1 \left(R_0^2 - R_i^2\right)\left(R_0^2 + R_i^2\right) \\ &= \frac{1}{2} m_{p1} \left(R_0^2 + R_i^2\right). \end{aligned} \tag{4.10}$$

where R_o and R_i are the outer and inner radius of the vessel. Note that since m_{p1} is fixed then I_{p1} is also a constant. The inertia is a function of the mass and radius of the vessel.

The time-varying mass m_{p2} can be written as

$$\begin{aligned} m_{p2}(t) &= \int_V \rho_w dV \\ &= \rho_w \pi R_i^2 h(t). \end{aligned} \tag{4.11}$$

where $h(t)$ is the time function of liquid level. The inertia of the time dependent mass is

$$\begin{aligned} I_{p2} &= \frac{1}{2}\pi\rho_w R_o^4 h\,(t) \\ &= \frac{1}{2} m_{p2} R_o^2 \end{aligned} \tag{4.12}$$

also the derivative of the inertia is

$$\begin{aligned} \dot{I}_{p2} &= \frac{1}{2}\pi\rho_w R_o^4 \dot{h}\,(t) \\ &= \frac{1}{2}\dot{m}_{p2} R_o^2 \end{aligned} \tag{4.13}$$

If the mass is time-varying, then the inertia is also time-varying. Note that if the derivative of the mass, $\dot{m}$ is a constant, then the derivative of the inertia is also a constant. On the other hand, if $\dot{m}$ is a time dependent function, then $\dot{I}$ is also a time dependent function.

In the implementation, the uncertainty term is written as

$$Y(\cdot)\,\tilde{\phi} = \tilde{M}(\cdot)\,\ddot{\zeta} + \tilde{C}(\cdot)\,\dot{\zeta} + \tilde{F}(\cdot)\,\dot{\zeta}$$

where

$$\begin{aligned} \tilde{M}(\cdot) &= \begin{bmatrix} v_1 & v_2 \\ v_2 & v_3 \end{bmatrix} \tilde{m}_p + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tilde{I}_p, \\ \tilde{C}(\cdot) &= l_1 l_2 \sin(q_2) \begin{bmatrix} -\dot{q}_2 & -(\dot{q}_1 + \dot{q}_2) \\ \dot{q}_1 & 0 \end{bmatrix} \tilde{m}_p, \\ F(\cdot) &= \begin{bmatrix} v_1 & v_2 \\ v_2 & v_3 \end{bmatrix} \tilde{m}_p + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tilde{I}_p, \\ \zeta &= \begin{bmatrix} \zeta_{11} \\ \zeta_{21} \end{bmatrix}. \end{aligned}$$

then the $Y(\cdot)\tilde{\phi}$ can be written as

$$Y(\cdot)\tilde{\phi} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} \tilde{m}_p \\ \tilde{I}_p \\ \tilde{m}_p \\ \tilde{I}_p \end{bmatrix}$$

where

$$\begin{aligned}
a_{11} &= v_1\ddot{\zeta}_{12} + v_2\ddot{\zeta}_{21} - l_1 l_2 \sin(q_2)(\dot{q}_2\dot{\zeta}_{11} + (\dot{q}_1 + \dot{q}_2)\dot{\zeta}_{21}), \\
a_{12} &= \ddot{\zeta}_{11} + \ddot{\zeta}_{21}, \\
a_{13} &= v_1\dot{\zeta}_{12} + v_2\dot{\zeta}_{21}, \\
a_{14} &= \dot{\zeta}_{11} + \dot{\zeta}_{21}, \\
a_{21} &= v_2\ddot{\zeta}_{12} + v_3\ddot{\zeta}_{21} + l_1 l_2 \sin(q_2)\dot{q}_1\dot{\zeta}_{11}, \\
a_{22} &= \ddot{\zeta}_{11} + \ddot{\zeta}_{21}, \\
a_{23} &= v_2\dot{\zeta}_{12} + v_3\dot{\zeta}_{21}, \\
a_{24} &= \dot{\zeta}_{11} + \dot{\zeta}_{21}.
\end{aligned} \tag{4.14}$$

It is also possible to write matrix $Y(\cdot)\tilde{\phi}$ as

$$Y(\cdot)\tilde{\phi} = Y_1(\cdot)\tilde{\phi}_2 + Y_2(\cdot)\tilde{\phi}_2$$

where

$$\begin{aligned}
Y_1(\cdot) &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \\
Y_2(\cdot) &= \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix}, \\
\tilde{\phi}_1 &= \begin{bmatrix} \tilde{m}_p \\ \tilde{I}_p \end{bmatrix}, \\
\tilde{\phi}_2 &= \begin{bmatrix} \tilde{m}_p \\ \tilde{I}_p \end{bmatrix}.
\end{aligned}$$

Note the $\tilde{\phi}_1$ represents the time-varying parameters and $\tilde{\phi}_2$ is the first derivative of the time-varying parameters.

4.3 Reference Trajectory

The trajectory used in the experiment is shown in Figure 4.2. The desired trajectory is smooth in joint position, velocity and acceleration. The magnitude of the desired acceleration is adjusted such that the control torque does not excess the motors limit at least for the feedforward case. Figure 4.3 shows the reference trajectory in task space.

4.4 Implementation of Designed Controller

In this section, the controllers that are proposed in Chapter 3 are implemented on the NSK two-link robot manipulator. Following are the gains that are used in implementation.

Case 1: PD control PD control has the form

$$\tau = K_D e_v = K_D \left(\dot{e} + \lambda e \right) \tag{4.15}$$

where $K_D = \begin{bmatrix} 15 & 0 \\ 0 & 8 \end{bmatrix}, \lambda = \begin{bmatrix} 150 & 0 \\ 0 & 40 \end{bmatrix}$

Case 2: Computed torque control The computed torque controller has the form:

$$\tau = M(\cdot)\ddot{\zeta} + C(\cdot)\dot{\zeta} - K_D e_v \tag{4.16}$$

where $K_D = \begin{bmatrix} 22 & 0 \\ 0 & 14 \end{bmatrix}, \lambda = \begin{bmatrix} 150 & 0 \\ 0 & 40 \end{bmatrix}$

Case 3: Robust adaptive with time-varying model The robust adaptive can be divided into inner and outer control law which has the form:

$$\tau = \tau_0 + \tau_1$$

where

$$\tau_0 = M(\cdot)\ddot{\zeta} + C(\cdot)\dot{\zeta} + F(\cdot)\dot{\zeta} - K_D e_v \tag{4.17}$$

and

$$\tau_1 = \begin{cases} -\frac{e}{\|e\|}\left(\frac{1}{k_r}\hat{\rho}_1 + \rho_2\right) & \|e\| \geq \varepsilon \\ -\frac{e}{\varepsilon}\rho_2 & \|e\| < \varepsilon \end{cases} \tag{4.18}$$

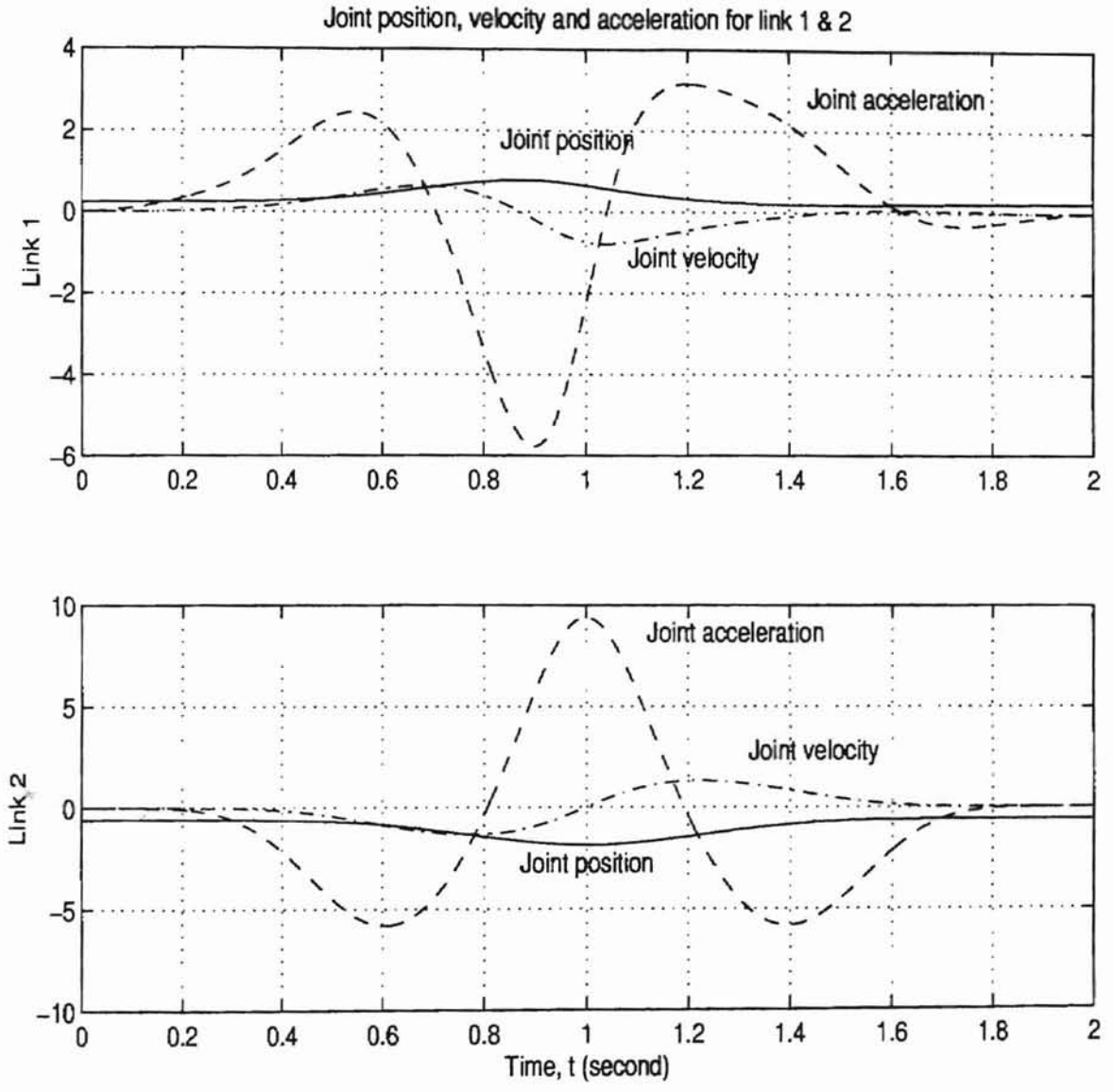

Figure 4.2: The reference joint trajectory for link 1 and 2

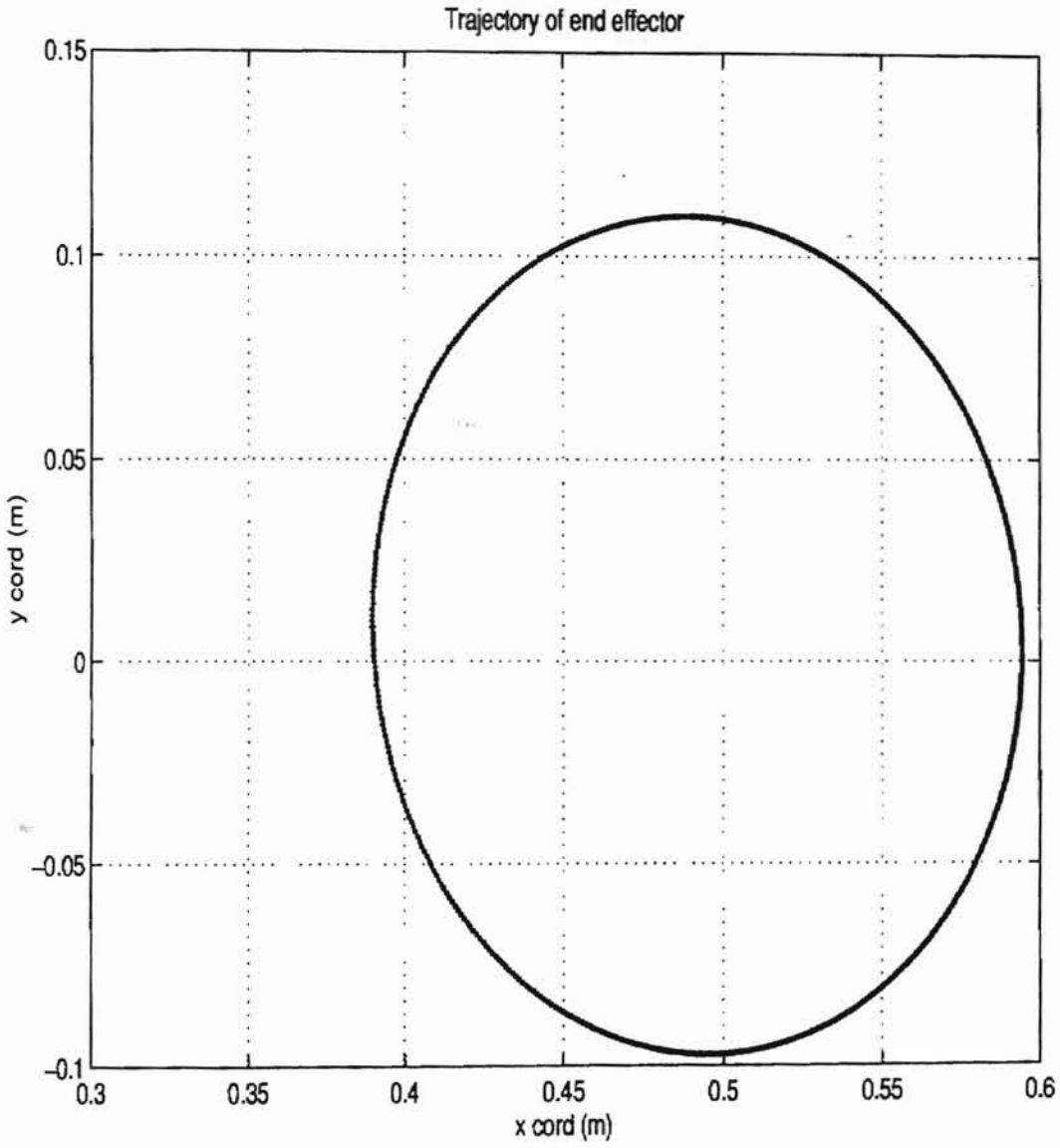

Figure 4.3: Reference trajectory in Cartesian coordinate

where update law is given by

$$\begin{aligned} \hat{\rho}_1 &= \frac{\Gamma}{k_r^2}\int \|e_v\| dt \\ \rho_2 &= \|Y(\cdot)\|^2 \end{aligned} \tag{4.19}$$

with $K_D = \begin{bmatrix} 12 & 0 \\ 0 & 1 \end{bmatrix}, \Gamma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, $k_r = 2$, and $\varepsilon = 0.08$. Notice that a small adaptation gain for the first link is chosen. This is because the parameters of the first link are assumed to be known.

Case 4: Adaptive control by assuming constant flow rate In this case, the constant flow rate of the water is assumed, then we can write

$$\begin{aligned} \tilde{\phi}_2 &= \tilde{k}_1 \\ \tilde{\phi}_1 &= \tilde{k}_1 t + \tilde{k}_0. \end{aligned}$$

then

$$\begin{aligned} Y(\cdot)\tilde{\phi} &= Y_1(\cdot)\tilde{\phi}_1 + Y_2(\cdot)\tilde{\phi}_2 \\ &= Y_1(\cdot)t\tilde{k}_0 + \left(Y_1(\cdot)t + Y_2(\cdot)\right)\tilde{k}_1. \end{aligned} \tag{4.20}$$

where

$$\begin{aligned} Y_1(\cdot)t &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} t, \\ Y_2(\cdot) &= \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix}. \end{aligned}$$

Then choose

$$\tau_0 = M(\cdot)\ddot{\zeta} + C(\cdot)\dot{\zeta} + F(\cdot)\dot{\zeta} - K_D e_v \tag{4.21}$$

and updated laws are

$$\dot{\hat{k}}_0 = -\Gamma_1 \int Y_1(\cdot)\, t e_v dt \tag{4.22}$$

$$\dot{\hat{k}}_1 = -\Gamma_2 \int \left(Y_1(\cdot)t + Y_2(\cdot)\right) e_v dt \tag{4.23}$$

where

$$\Gamma_1 = \begin{bmatrix} 0.045 & 0 \\ 0 & 0.0014 \end{bmatrix},$$

$$\Gamma_2 = \begin{bmatrix} 0.045 & 0 \\ 0 & 0.025 \end{bmatrix}.$$

and the parameters a_i are defined in (4.14).

Case 5: Integral saturation adaptive robust controller The inner control law is chosen same as Case 3 and outer control law is given by

$$\tau_1 = \begin{cases} -\frac{e}{\|e\|}\left(\frac{1}{k_r}\hat{\rho}_1 + \rho_2\right) & t \leq t_p \\ -\frac{e}{\varepsilon}(\frac{1}{k_r}\rho_f + \rho_2) - K_I \int e_v dt & t > t_p \end{cases} \tag{4.24}$$

where $t_\varepsilon = 10$ sec, is the time where upper bound is estimated, and ρ_f will remain constant after t_ε.

Chapter 5

IMPLEMENTATION RESULTS

5.1 Results

All the control laws outlined in Chapter 3 were implemented on the NSK direct drive two-link manipulator. The influence of the unknown time-varying payload is examined experimentally. A sampling time of 2 milli-seconds and a period of 2 seconds is chosen for the trajectory. Eight cycles are implemented for each controller. Both pumping water in and out of the vessel is performed. Experimental results are given in Figure 5.1 through 5.13.

The following controllers were implemented on the experimental setup: (1) PD control (Figures 5.1, 5.2); (2) Computed torque control (Figures 5.3, 5.4); (3) Saturation type adaptive robust control (Figures 5.5, 5.6); (4) Adaptive control (Figures 5.8, 5.9, 5.10, 5.11); (5) Integral saturation adaptive control (Figures 5.12, 5.13). For all cases, pumping of water in and out of the vessel during motion is considered. Also note that in Case 5, the water is pump in/out before robot starts. The purpose of of doing this is to make sure that upper bound is estimated before fifth period.

- In the case PD controller (Figures 5.1 and 5.2), the tracking error of the first link is the same for each cycle. The tracking error in the second link decreases as the mass of the payload decreases (pump water out, Figure 5.2) and increases as the mass of the payload increases (pump water in, Figure 5.1). This is expected as the time-varying payload directly sits on the second link. Another possible reason is that the PD gains are fined. We have chosen high gains to tune the PD controller, and this may cause

the system to destabilize if there are any other disturbances.

- Computed torque control law (Figures 5.3 and 5.4), has better performance as compared to PD controller as expected because it includes as exactly feedforward term to compensate the nonlinear dynamics. From Figures 5.3 and 5.4, we notice that the error do increases/decreasesas the mass of the payload increases/decreases.

- In comparison the PD and computed torque controller, the robust adaptive controller (Figures 5.5 and 5.6) has better tracking error performance and seems to adjust well to the tracking error in the second link, Figures 5.5 and 5.6, improves considerably after several cycles, and this is a significant improvement over PD and computed torque control as the payload is on the second link.

- The results of adaptive controller are given in Figure 5.8 through Figure 5.11. In terms of the tracking error, the pure adaptive controller has similar performance as adaptive robust controller. In addition, Figure 5.9 and Figure 5.11 show the convergence of the updated parameters to some fixed values. We do not know if these are the true values as we do not know the actual flow rate in/out of the experiment.

- Results of integral saturation adaptive controller are given in Figure 5.12 and Figure 5.13. The performance is comparable to pure adaptive and adaptive robust controller. But, the saturation introduces chattering in the second link.

In conclusion, both adaptive robust controller and pure adaptive controller give satisfactory performance in the presence of uncertain time-varying payload.

The following remarks illustrate some other observations from the experiments.

1. Even though the velocity is estimated by the first order backward difference of position, it still appears quite noisy resulting in chattering.

2. It should be noticed that increasing payload mass (pump water in) degrades the performance of all controllers as compared to decreasing payload mass (pump water out). The controllers that are least affected by this phenomena are pure adaptive controller and adaptive robust controller as expected.

3. Since the time-varying payload is on the second link, the tracking error in the second link is directly affected, as can be noticed from the results.

4. The choice of "ϵ", the boundary layer thickness, is critical in the control gain tuning process.

5. The mass flow rate of the fluid in/out of the vessel is quite small for our experimental setup and is constrained by the choice of the size pipe and pump capacity. We expect to see a more contrasting results between several controllers if the mass flow rate is higher.

5.2 Experimental Plots

In this section, the data from the experiments is plotted using MATLAB. The total time for each experiment is 16 seconds, which corresponds to eight cycles. e_1 and e_2 are tracking erros in link 1 and link2, respectively.

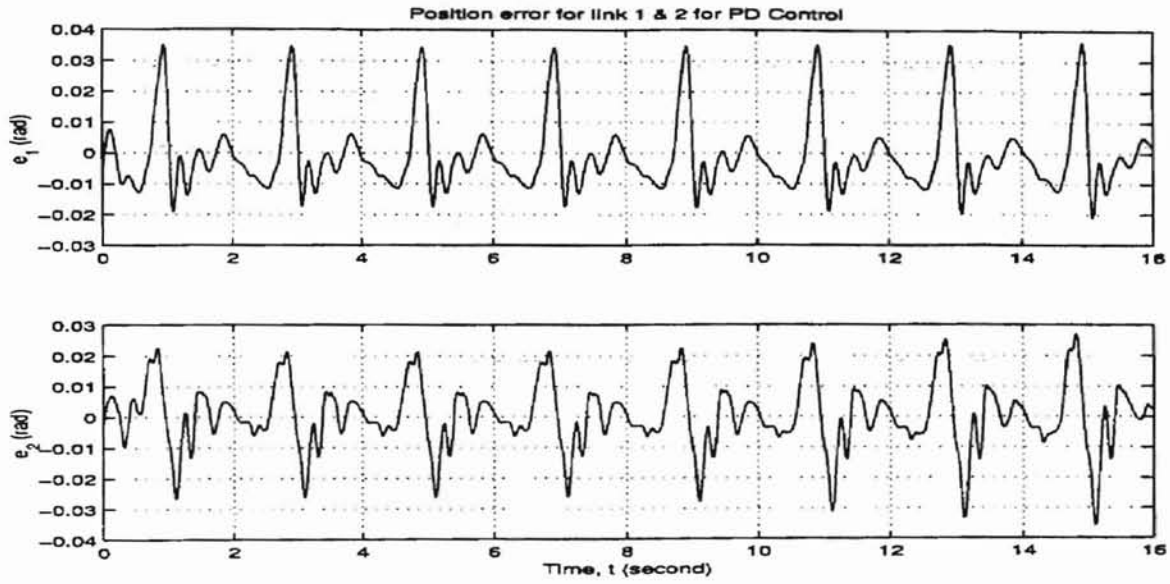

Figure 5.1: Position error of pumping water in using PD control

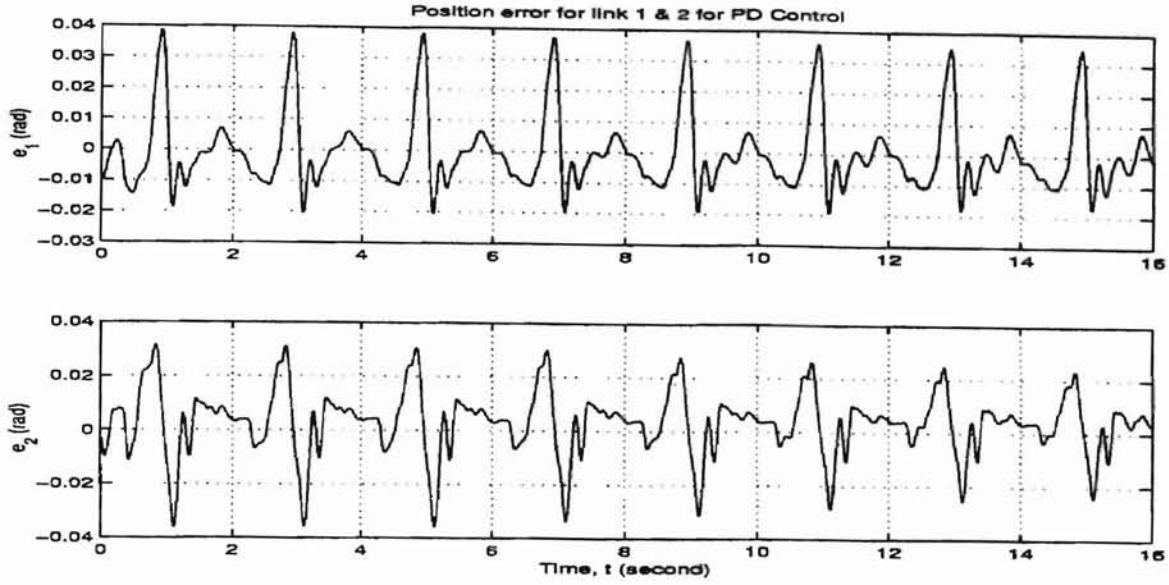

Figure 5.2: Position error of pumping water out using PD control

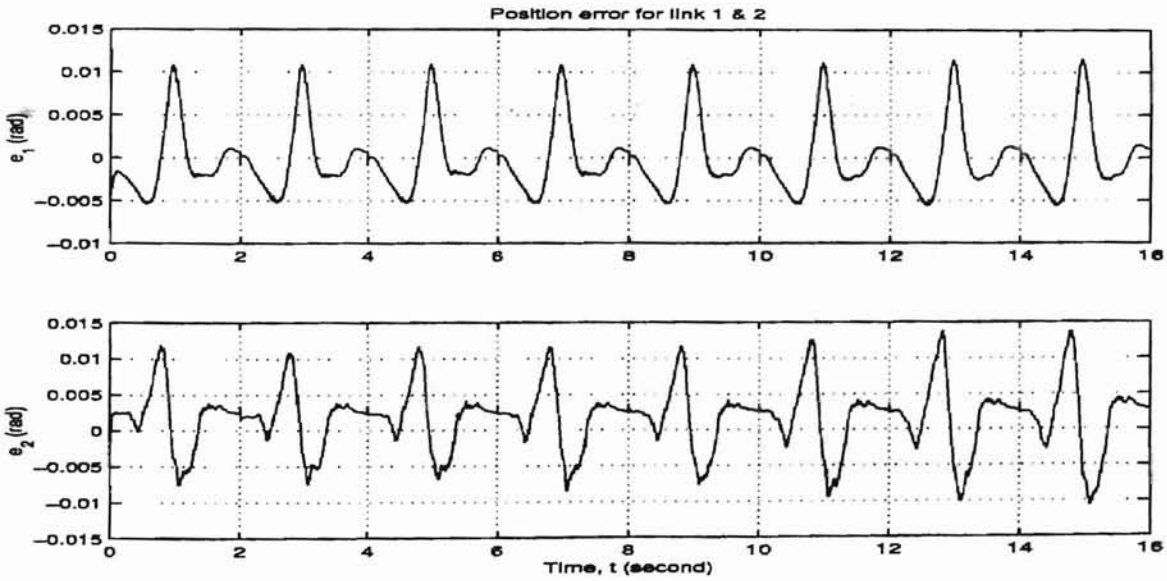

Figure 5.3: Pump water in using computed torque control

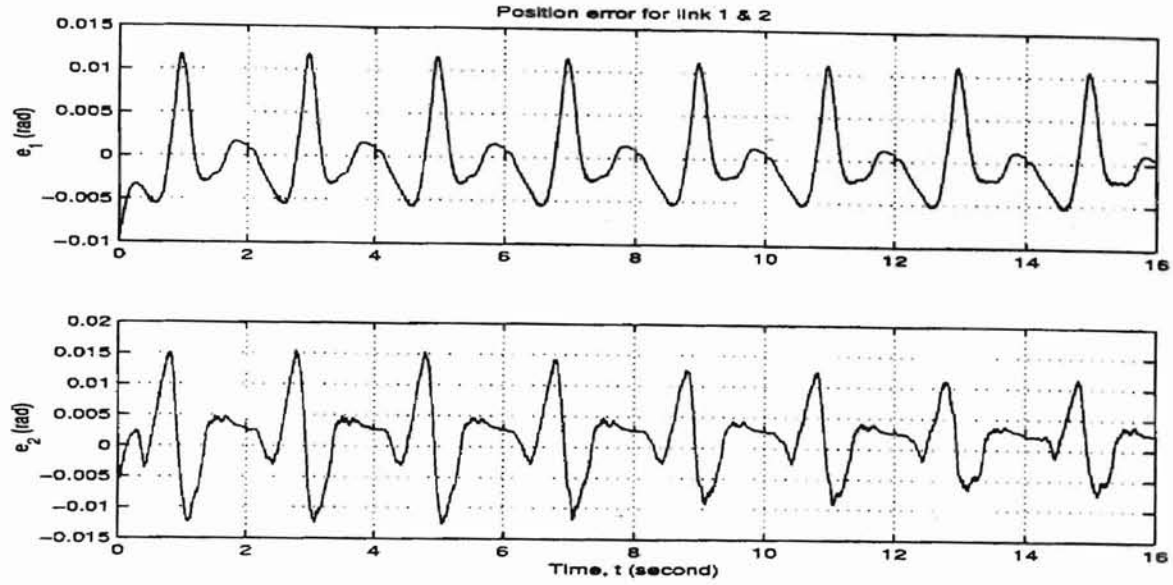

Figure 5.4: Pump water out using computed torque control

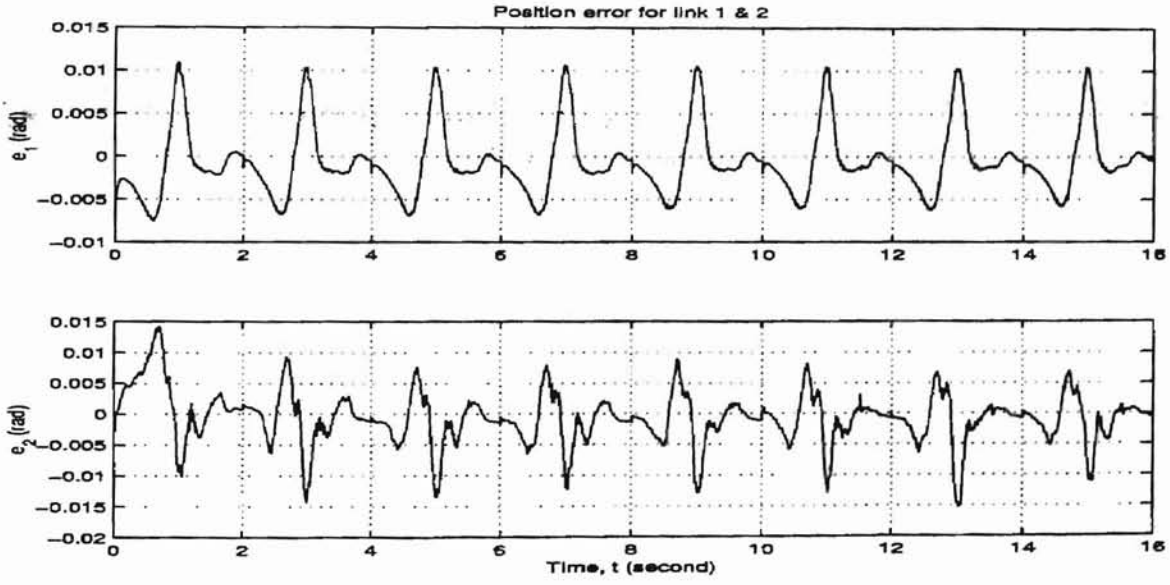

Figure 5.5: Position error of pumping water in using adaptive robust controller

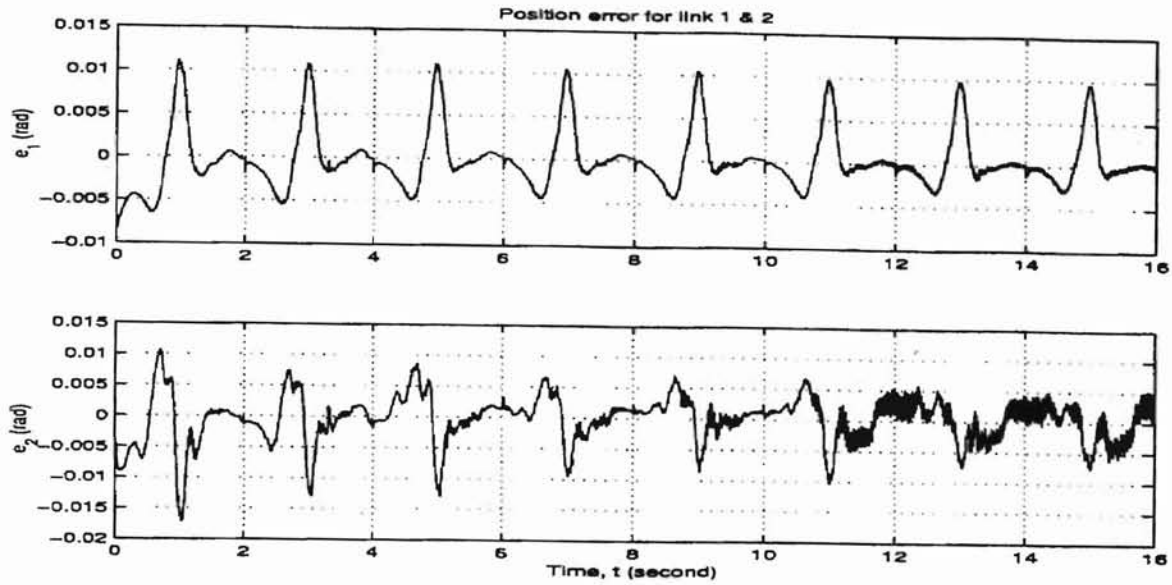

Figure 5.6: Position error of pumping water out using adaptive robust controller

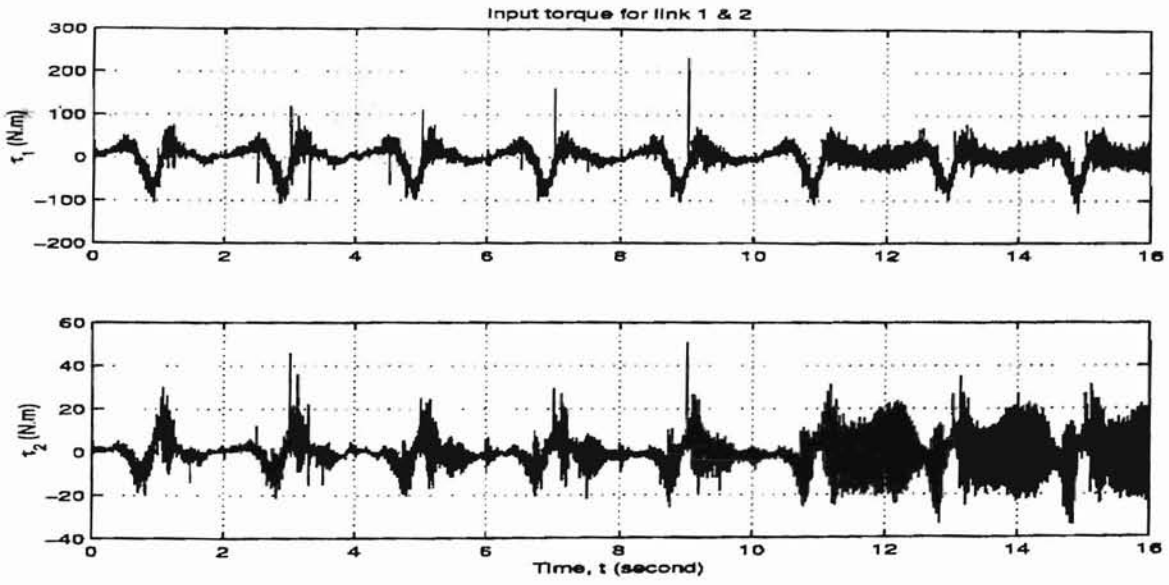

Figure 5.7: Torque of pumping water out using adaptive robust controller

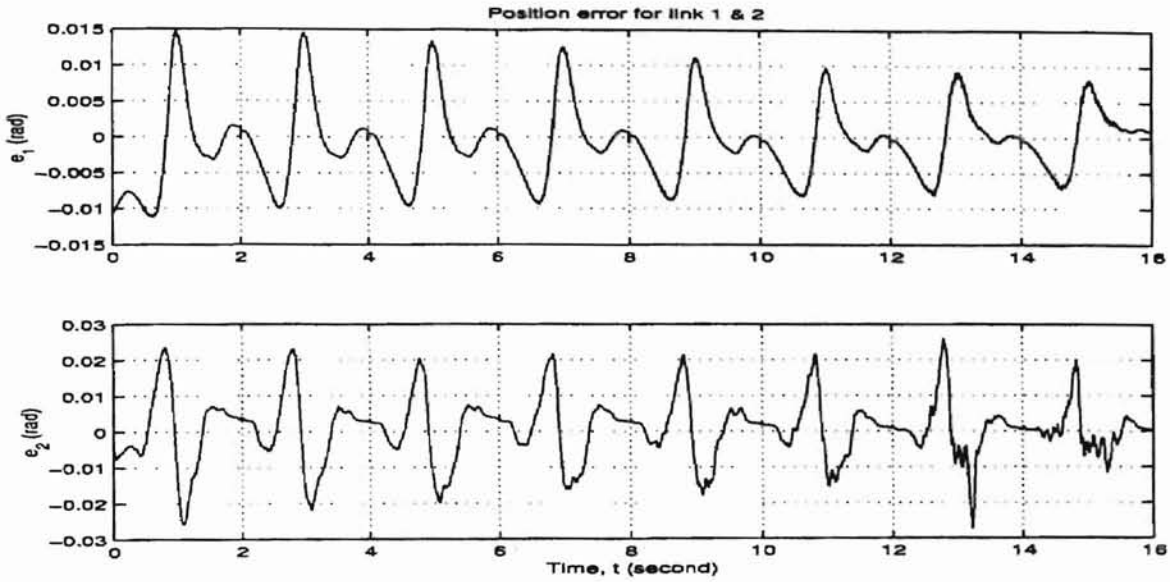

Figure 5.8: Position error of pumping water out using pure adaptive controller

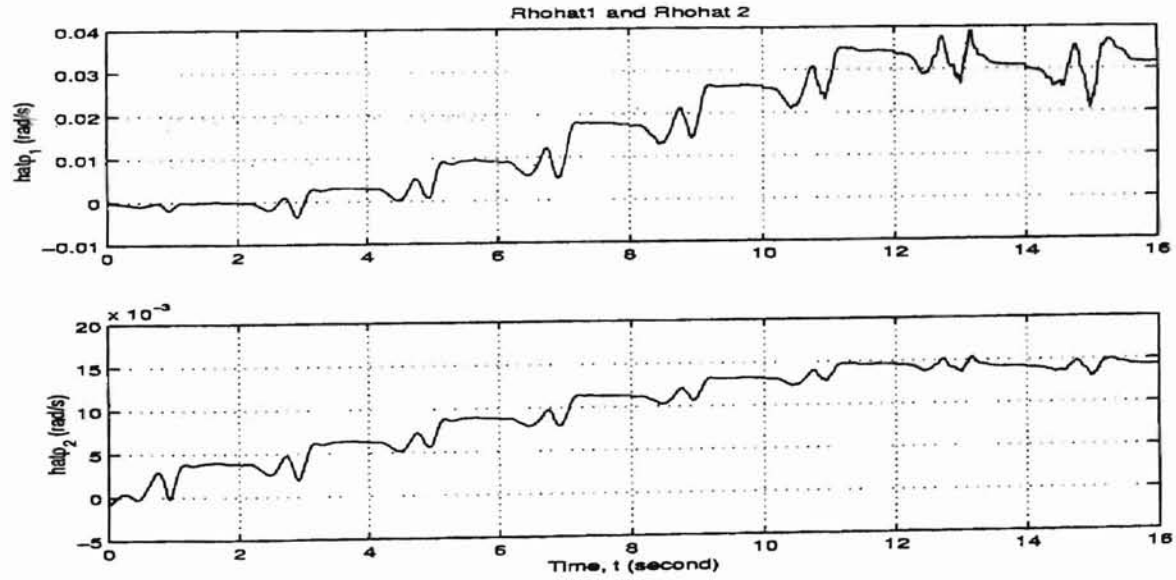

Figure 5.9: Parameter estimate

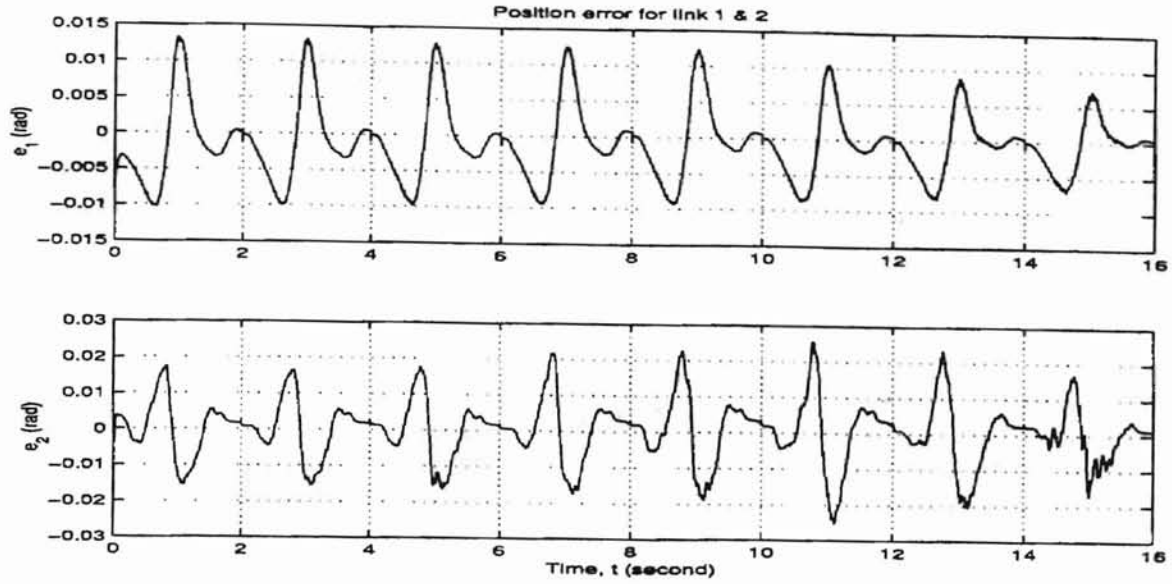

Figure 5.10: Position error of pumping water in using pure adaptive controller

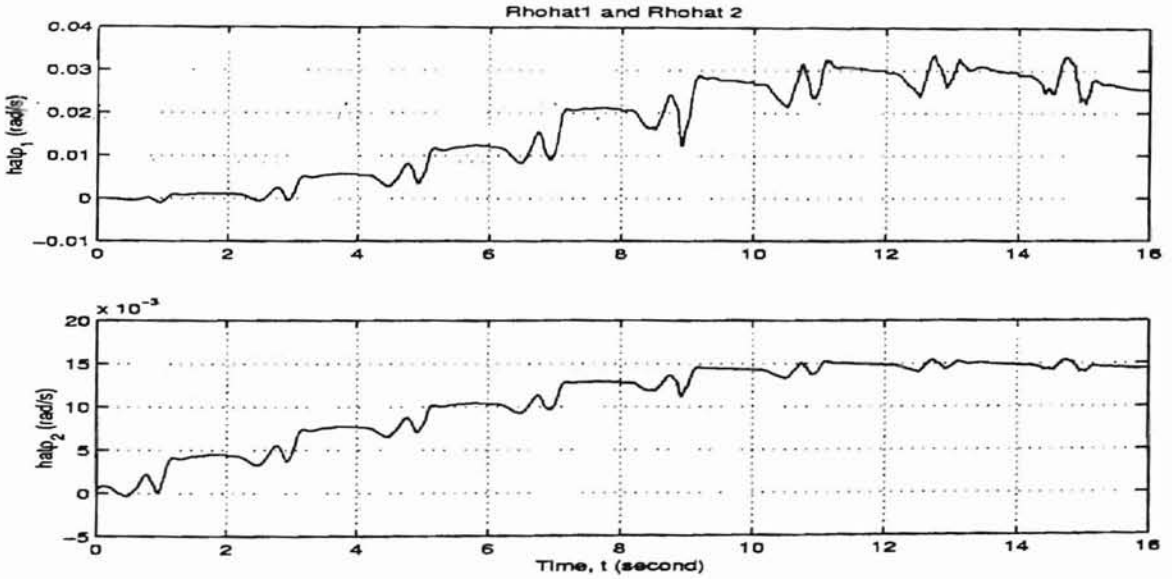

Figure 5.11: Parameter estimate

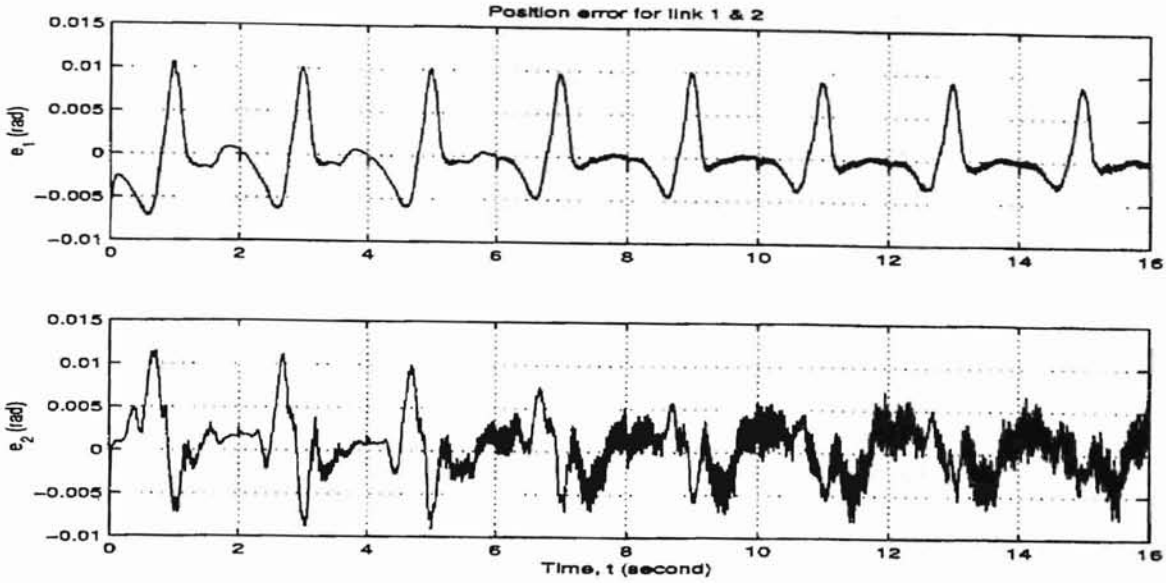

Figure 5.12: Position error of pumping water out using PI saturation adaptive robust controller

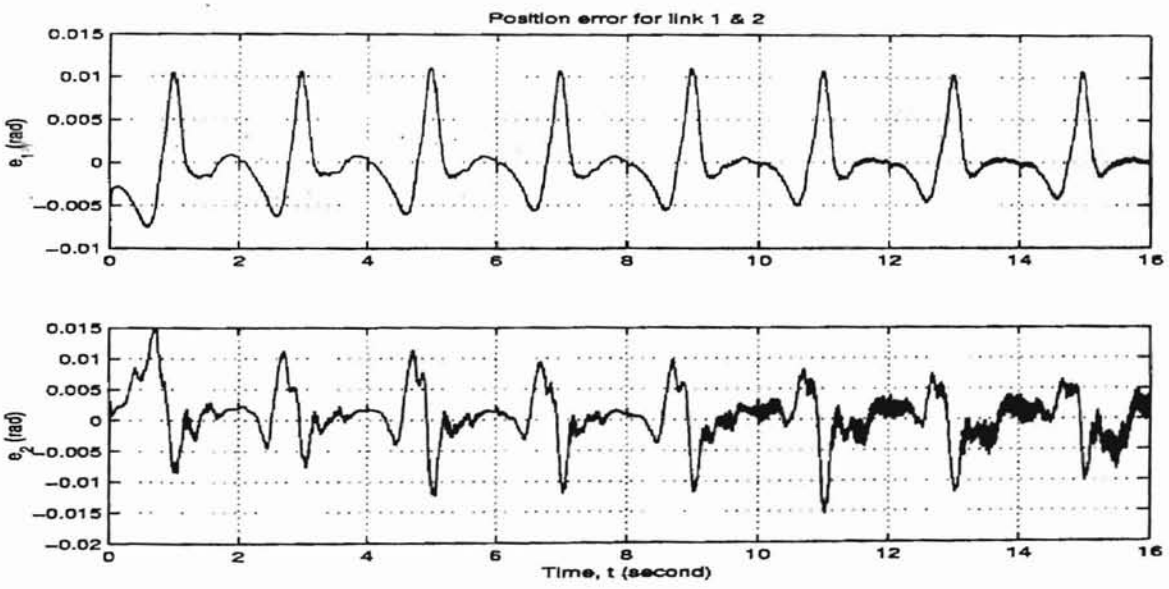

Figure 5.13: Position error of pumping water in using PI saturation adaptive robust controller

Chapter 6

CONCLUSIONS

A new dynamic model for robot manipulators which consisting of time-varying mass is derived. The compact dynamic model developed is suitable for controller designs as parameterization of the dynamics into a known part and an unknown part can be easily done. It is also shown that most of the properties for the traditional time-invariant manipulators are preserved for the time-varying model.

A stable robust adaptive control law and pure adaptive control law have been designed for the time-varying robot manipulator system and a comparison with their counterpart control laws have been presented in this report. Both analytical and experimental comparisons were conducted, and the results confirm the merit of the proposed control laws. The results from the experiments show that the proposed control laws improve the performance of trajectory tracking in presence of an unknown time-varying payload.

Future research in this area can be extended towards analysis and design of performance based control laws, especially for the case of faster time-varying payloads. Since the robot dynamics is highly nonlinear, the controller designs can find application in other nonlinear systems. The design of the experimental platform and its availability for implementation is an added advantage in checking the practicality of the control designs.

BIBLIOGRAPHY

[1] C.W. Kilmister. *Lagrangian Dynamics: An Introduction For Students.* Logos Press Limited, 1967.

[2] J.B. Marion. *Classical Dynamics of Particles and Systems.* Academic Press, 1965.

[3] M.C. Calkin. *Lagrangian and Hamiltonian Mechanics.* World Scientific, 1996.

[4] H. Goldstein. *Classical Mechanics.* Addison-Wesley Publishing Company, 2 edition, 1980.

[5] D.T. Greenwood. *Classical Dynamics.* Prentice-Hall, Inc., 1977.

[6] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley And Sons, 1989.

[7] H. Asada and J.J. Slotine. *Robot Analysis and Control.* John Wiley And Son, 1986.

[8] C. Abdallah, D. Dawson, P. Dorato, and M. Jamshidi. Server of robust control for rigid robots. *IEEE Contr. Syst. Mag.*, 11(2):24–30, 1991.

[9] M.W. Spong, J.S. Thorp, and J.W. Kleinwaks. Robust microprocessor control of robot manipulators. *Automatica*, 23:373–379, 1987.

[10] S. Gutman. Uncertain dynamic system - a lyapunov min-max approach. *IEEE Trans. Autom. Contr.*, AC-24:437–443, 1979.

[11] G. Leitmann. On the efficacy of nonlinear control in uncertain linear system. *J. Dyn. Sys., Meas., and Contr.*, 102:95–102, 1981.

[12] M. Corless and G. Leitmann. Continuous state feedback guarenteeing uniform ultimate boundedness for uncertain system. *IEEE Trans. Autom. Contr.*, AC-26:1139–1144, 1981.

[13] Z. Qu and J. Dorsey. Robust tracking control of robots by a linear feedback law. *IEEE Trans. Autom. Contr.*, 36(9):1081–1084, 1991.

[14] M.W. Spong. On the control of robot manipulators. *IEEE Trans. Autom. Contr.*, 37(11):1782–1786, 1992.

[15] K.M. Koo and J.H. Kim. Robust control of robot manipulators with parametic uncertainty. *IEEE Trans. Autom. Contr.*, 39(6):1230–1233, 1994.

[16] V.I. Utkin. Variable structure system with sliding modes. *IEEE Trans. Auto. Contr.*, AC-22:212–222, April, 1978.

[17] K.D. Young. Controller design a manipulator using theory of variable structure systems. *IEEE Trans. Syst., Man, and Cyber.*, SCM-8:210–218, 1978.

[18] J.J. Slotine. The robust control of robot manipulators. *Int. J. Rob. Res.*, 4(2):49–64, 1985.

[19] J.J. Slotine and S.S. Sastry. Tracking control of nonlinear systems using sliding surfaces with applications to robot manipulators. *Int J. Contr.*, 38:465–492, 1983.

[20] F.L. Lewis, C.T. Abdallah, and D. M. Dawson. *Control of Robot Manipulators.* Macmillan Publishing Company, 1993.

[21] Z. Qu and D.M. Dawson. *Robust Tracking Control of Robot Manipulators.* IEEE Press, 1996.

[22] Y.H. Chen. Design of robust controllers for uncertain dynamical systems. *IEEE Trans. Auto. Contr.*, 33(5):487–491, 1988.

[23] L. Cai and A. Abdallah. A smooth tracking controllers for uncertain robot manipulator. In *Proc. IEEE Int. Conf. Rob. and Auto.*, pages 83–88, Alanta, GA, 1993.

[24] L. Cai and G. Song. Joint stick-slip friction compensation of robot manipulaots by using smooth robust controllers. *J. Rob. Sys.*, 11(6):452–470, 1994.

[25] R. Ortega and M.W. Spong. Adaptive motion control of rigid robots: a tutorial. In *Proc. IEEE Conf. Dec. and Contr.*, pages 1575–1584, Austin, TX, 1988.

[26] J.J. Slotine and W. Li. On the adaptive control of robot manipulators. *Int. J. Rob. Res.*, 6(3):49–59, 1987.

[27] J.J. Slotine and W. Li. Adaptive manipulator control: A case study. *IEEE Trans. Auto. Contr.*, 33(11):995–1003, 1987.

[28] H. Berghuis, R. Ortega, and H. Nijmeijer. A robust adaptive controller for robot manipulators. In *Proc. IEEE Int. Conf. Rob. and Auto.*, pages 1876–1881, Nice, FL, 1992.

[29] C.C. de Wit and N. Fixot. Adaptive control of robot manipulators via velocity estimated state feedback. *IEEE Trans. Auto. Contr.*, 36:1234–1237, 1992.

[30] C.C. de Wit and J.J. Slotine. Sliding observers for robot manipulators. *Automatica*, 27:859–864, 1991.

[31] C.C. de Wit, B. Siciliano, and G. Bastin. *The Theory of Robot Control.* Springer, 1996.

[32] P.A. Ioannou and J. Sun. Robust adaptive control: A unified approach. In *Proc. Conf. Dec. and Contr.*, pages 1876–1881, Tampa, FL, 1989.

[33] J.S. Reed and P.A. Ioannou. Instability analysis and robot adaptive control of robotic manipulators. *IEEE Trans. Rob. and Auto.*, 5(3):381–386, 1989.

[34] L. Cai and R.W. Longman. Integrated adaptive robust control of robot manipulator with joint stick-clip friction. In *Proc. IEEE Int. Conf. on Contr. Appl.*, pages 83–117–182, AlaHartford, CT, 1997.

[35] L. Cai, R.W. Longman, R. Mukherjee, and J. Zhang. Integrated sliding-mode adaptive-robust control. In *Proc. IEEE Int. Conf. Rob. and Auto.*, pages 656–661, Dearborn, MI, 1996.

[36] B. Brogliato and I. Landau. Robust adaptive motion control of rigid robotds subject to input and output bounded disturbances. In *Proc. IEEE Int. Conf. Rob. and Auto.*, pages 1895–1990, Nice France, 1992.

[37] K.S. Narendra and J.D. Boskovic. A combined direct, indirect, and variable structure method fro robust adaptive control. *IEEE Trans. Auto. Contr.*, 37(2):262–268, 1992.

[38] W.M. Lu, F.Y. Hadaegh, and A. Packard. Adaptive robust control for uncertain nonlinear systems. In *Proc. Conf. Dec. and Contr.*, pages 2103–2108, San Diego, CA, 1997.

[39] B. Yao, M. Al-Majed, and M. Tomizuka. High perfromance robust motion control of machine tools: An adaptive robust control approach and comparative experiments. In *Proc. Amec. Contr. Conf.*, pages 2754–2758, Albuquerque, NM, 1997.

[40] G. Liu and A.A. Goldenberg. Comparative study of robust saturation-based control of robot manipulators: Analysis and experiments. *Int. J. Rob. Res.*, 15(5):473–491, 1996.

[41] Y.D. Song and R.H. Middleton. Dealing with the time-varying paprameter problem of robot manipulators performing path tracking tasks. In *Proc. Dec. and Contr.*, pages 3112–3117, Honolulu, Hawaii, 1990.

[42] Y.D. Song and R.H. Middleton. Dealing with the time-varying paprameter problem of robot manipulators performing path tracking tasks. *IEEE Trans. Auto. Contr.*, 37(10):1597–1601, 1992.

[43] B. Brogliato and A. Trofino-Neto. Practical stabilization of a class of nonlinear system with partially known uncertainty. *Automatica*, 31:145–150, 1995.

Appendix: Source Code

pau8.c

```
%
% The following file is the servo file for the adaptive control
% by assuming constant flow rate.
%

#include <cntrl.h>
#include <math.h>

/*===================================
Name: pau8.c Parameterize Adaptive Control (constant mass flow rate)

Control algorithm using trajectory down loaded in advance

All calculation inside the user algorithm base
on Engineering NM-rad unit system
Note:
1. All position read be timesed by c2r=0.00004091 before being used.
2. Tau1 times t2c_1=2047/tau_max1=8.3551 before being outputed.
   Tau2             2047/tau_max2=51.75--------------------.
3. Torque to Voltage is:
   t2v_1=10/tau_max1=0.040816
   t2v_2=10/tau_max2=0.25
4. Relations between kp based on counter and kpe based on
   Engineering unit is:
   kpe1=tau_max1/(2047*c2r) kp1=2925.6 kp1.
   kpe2=477.65 kp2.

===================================*/
```

```
#define TRAJPT 1000

/* Gain base on Engineering Unit */
float VF_kpe1=15,VF_kpe2=8;
float VF_lambdap1=50,VF_lambdap2=20;
float def_kp1=2000,def_kp2=500,def_kd1=15,def_kd2=5;

 /*------------------------------
 Robot constant parameter
 --------------------------------*/

float I_1=0.2675, I_2=0.360, I_3=0.0077, I_3c=0.040, I_4=0.051;
float m_1=73.0, m_2=10.6, m_3=12, m_4=4.85, m_p=0;
float l_1=0.38, l_2=0.24, l_3=0.139, l_4=0.099, I_p=0;

/* Inertia parameter */
float p_1, p_2, p_3;

/* p_1 = 4.2420 */
/* p_2 = 0.1825 */
/* p_3 = 0.1062 */

float tau1,tau2;
float t2c_1=8.3551,t2c_2=51.75;
float err1=0,err2=0,olderr1,olderr2,derr1,derr2;
float dpos1,dpos2, oldpos1, oldpos2;
float mdot_p=0.01, Idot_p=0.001;
float zeta1d,zeta2d,zeta1dd,zeta2dd;
float ev1=0,ev2=0;
float Ts;
float c2r=0.00004091, pi=3.1415927;
float r1,r1d,r1dd,r2,r2d,r2dd;

int i=0,j=0;

float v_1,v_2,v_3,v_4;
float m11,m12,m21,m22;
float c11,c12,c21,c22;
float f11,f12,f21,f22;
```

```
float h11,h12;

float a11,a12,a13,a14,a21,a22,a23,a24;
float Y11,Y12,Y21,Y22;
float time;

float sumr11=0, sumr12=0,sumr21=0,sumr22=0;
float Irho11=0,Irho12=0,Irho21=0,Irho22=0;
float k1_adp=0.05, k2_adp=0.003, k3_adp=0.05, k4_adp=0.003;
float rhohat11, rhohat12,rhohat21, rhohat22;
float u_01, u_02;

  float TF_output1[TRAJPT*8];
  float TF_err1[TRAJPT*8];
  float TF_derr1[TRAJPT*8];
  /*float TF_u_01[TRAJPT*8];
  float TF_ev1[TRAJPT*8];
  */

  float TF_output2[TRAJPT*8];
  float TF_err2[TRAJPT*8];
  float TF_derr2[TRAJPT*8];
  /*float TF_u_02[TRAJPT*8];
  float TF_ev2[TRAJPT*8]; */
float circle=2;

float TF_ref1[TRAJPT],TF_ref2[TRAJPT];
float TF_rv1[TRAJPT],TF_rv2[TRAJPT];
float TF_xc1[TRAJPT],TF_xc2[TRAJPT];

float TF_k[40];
float VF_kl1=2000,VF_kl2=200;
float VF_x1=3,VF_x2=3;

init_control()
{
        Ts=0.002;
p_1 = I_1 + I_2 + I_3c + l_3*I_3*m_2 + l_1*I_1*(m_3+m_4) + I_3 + I_4 + l_4*I_4*m_4;
p_2 = l_1*l_4*m_4;
```

```
p_3 = I_3 + I_4 + l_4*I_4*m_4;

 }

control()
{
 if(Host_Trigger&&(i<=TRAJPT*8))
 {

    /* ------------- READ REF DATA  ------*/

    if(j>=TRAJPT) j=j-TRAJPT;
    r1=TF_ref1[j];
    r2=TF_ref2[j];
    r1d=TF_rv1[j];
    r2d=TF_rv2[j];
    r1dd=TF_xc1[j];
    r2dd=TF_xc2[j];

 /*----------------------------
 Define error and relative error
 ---------------------------------*/

    olderr1=err1;
    olderr2=err2;

    dpos1=c2r*(float)(pos1-oldpos1)/Ts;
    dpos2=c2r*(float)(pos2-oldpos2)/Ts;
    oldpos1=pos1;
    oldpos2=pos2;

    err1=c2r*(float)pos1-r1;
    err2=c2r*(float)pos2-r2;
    derr1=(err1-olderr1)/Ts;
    derr2=(err2-olderr2)/Ts;

    zeta1d=r1d-VF_lambdap1*err1;
    zeta2d=r2d-VF_lambdap2*err2;
    zeta1dd=r1dd-VF_lambdap1*derr1;
```

```
    zeta2dd=r2dd-VF_lambdap2*derr2;

    ev1=derr1+VF_lambdap1*err1;
    ev2=derr2+VF_lambdap2*err2;

/*-----------------------------
 Inertia variable
 ---------------------------------*/

    v_1 = l_1*l_1 + l_2*l_2 + 2*l_1*l_2*cos(pos2);
    v_2 = l_2*l_2 + l_1*l_2*cos(pos2);
    v_3 = l_2*l_2;

 /*-----------------------------
 Inertia Matrix
 -------------------------------*/

    m11 = p_1 + 2*p_2*cos(pos2) + v_1*m_p + I_p;
    m12 = p_3 + p_2*cos(pos2) + v_2*m_p + I_p;
    m21 = m12;
    m22 = p_3 + v_3*m_p + I_p;

 /*-----------------------------
 Centripental/Coriolis Matrix
 -------------------------------*/

    v_4 = (p_2 + l_1*l_2*m_p)*sin(pos2);
    c11 = -v_4*dpos2;
    c12 = -v_4*(dpos1+dpos2);
    c21 = v_4*dpos1;
    c22 = 0;

 /*-----------------------------
 Time-varying Mass Dependent Matrix
 ---------------------------------*/

    f11 = v_1*mdot_p + Idot_p;
    f12 = v_2*mdot_p + Idot_p;
    f21 = f12;
```

```
    f22 = v_3*mdot_p + Idot_p;


    h11 = m11*zeta1dd+m12*zeta2dd+(c11+f11)*zeta1d+(c12+f12)*zeta2d;
    h12 = m21*zeta1dd+m22*zeta2dd+(c21+f21)*zeta1d+(c22+f22)*zeta2d;

/*---------------------------------------
 Y(.) matrix
 ---------------------------------------*/
 a11=v_1*zeta1dd+v_2*zeta2dd-l_1*l_2*sin(pos2)*(dpos2*zeta1d+(dpos1+dpos2)*zeta2d);
 a12=zeta1dd+zeta2dd;
 a13=v_1*zeta1d+v_2*zeta2d;
 a14=zeta1d+zeta2d;
 a21=v_2*zeta1dd+v_3*zeta2dd+l_1*l_2*sin(pos2)*dpos1*zeta1d;
 a22=zeta1dd+zeta2dd;
 a23=v_2*zeta1d+v_3*zeta2d;
 a24=zeta1d+zeta2d;

    time=i*Ts;

    Y11=a11*time+a13;
    Y12=a12*time+a14;
    Y21=a21*time+a23;
    Y22=a22*time+a24;

    sumr11=sumr11+(Y11*ev1+Y21*ev2)*Ts;
    sumr12=sumr12+(Y12*ev1+Y22*ev2)*Ts;

    sumr21=sumr21+(a11*ev1+a21*ev2)*Ts;
    sumr22=sumr22+(a12*ev1+a22*ev2)*Ts;

    Irho11=sumr11;
    Irho12=sumr12;

    Irho21=sumr21;
    Irho22=sumr22;

    rhohat11=k1_adp*Irho11;
    rhohat12=k2_adp*Irho12;
```

```
   rhohat21=k3_adp*Irho21;
   rhohat22=k4_adp*Irho22;


/*   if(ri>=TRAJPT) ri=ri-TRAJPT; */

/*-----------------------------------
Adaptive Controller
-----------------------------------*/
   u_01=Y11*rhohat11+Y12*rhohat12+a11*rhohat21+a12*rhohat22;
   u_02=Y21*rhohat11+Y22*rhohat12+a21*rhohat21+a22*rhohat22;
   /*   u_01=1;
   u_02=0;*/

   tau1=h11-VF_kpe1*ev1-u_01;
   tau2=h12-VF_kpe2*ev2-u_02;

   u1=(int)(tau1*t2c_1);
   u2=(int)(tau2*t2c_2);


   if(i<TRAJPT*8){

          TF_output1[i]=(float)tau1;
          TF_err1[i]=(float)err1;
          TF_derr1[i]=(float)derr1;
         /* TF_u_01[i]=(float)u_01;
          TF_ev1[i]=(float)ev1;*/


          TF_output2[i]=(float)tau2;
          TF_err2[i]=(float)err2;
          TF_derr2[i]=(float)derr2;
         /* TF_u_02[i]=(float)u_02;
         TF_ev2[i]=(float)ev2;*/
        }

       i++;j++;
/*
```

```
    User1=rhohat21;
    User2=rhohat22;
*/

    User1=err1;
    User2=err2;

/*
   User1=u_01;
    User2=u_02;

    User1=what1;
    User2=what2;

    User1=u1;
    User2=u2;
*/
    }
else
        {
        r1=c2r*(float)pos1;
        r2=c2r*(float)pos2;
    oldpos1=pos1;
    oldpos2=pos2;
        r1d=0;
        r2d=0;
        r1dd=0;
        r2dd=0;
        Irho11=0;
        Irho12=0;
        Irho21=0;
        Irho22=0;
        sumr11=0;
        sumr12=0;
        sumr21=0;
        sumr22=0;

        /*---------------------------
        default algorithm
```

```
        ----------------------------*/
err1 = c2r*(pos1-ref1);
err2 = c2r*(pos2-ref2);
derr1=(err1-olderr1)/Ts;
derr2=(err2-olderr2)/Ts;
olderr1=err1;
olderr2=err2;

/*==============================================
Calculate the controller outputs for the next
period.
==============================================*/
tau1 = -(def_kp1 * err1+def_kd1*derr1) ;
tau2 = -(def_kp2 * err2+def_kd2*derr2) ;
u1=(int)(tau1*t2c_1);
u2=(int)(tau2*t2c_2);

/*
User1=tau1;
User2=tau2;
*/
User1=err2;User2=err1;

  if(!Host_Trigger)i=0;j=0;

        }
}
```

rpl.h

```
%
% This is the RPL used in motor for before and after the input
% data or desired trajectory.
%

#include "RPL.H"
RPL()
{
        int i;
```

```
        int j;
        Sample_Time((double).004);
        Jointspace_Max_Vel((double)3.0);
        Jointspace_Acceleration((double)15.0);
        Move_Jointa(0,0);
/*      Move_Jointa(0.245,-0.582762); */
        Wait_For(.5);
        Slave_Trigger(1.0);
        for(j=1;j<100;j++)
        {
                Wait_For(.5);
        }
        Slave_Trigger(0.0);
        Wait_For(0.5);
/*
    Move_Jointr(-0.3,-0.3);
*/
  Wait_For(0.5);
        Move_Jointa(0,0);
}
```

plotpau1.m

```
%
% This is the Matlab that used to plot the data from the experiment
%

t=0:1:7999;
load c:\matlab5\data\simulation\expplot\pau21_o1.dat;
load c:\matlab5\data\simulation\expplot\pau21_o2.dat;
load c:\matlab5\data\simulation\expplot\pau21_o3.dat;

err1=pau21_o1(:,2);
err2=pau21_o1(:,3);

ev1=pau21_o2(:,2);
ev2=pau21_o2(:,3);

tau1=pau21_o3(:,2);
```

```
tau2=pau21_o3(:,3);

derr1=pau21_o4(:,2);
derr1=pau21_o4(:,3);


figure;
subplot(211),plot(t,err1);grid;
title('Position error for link 1 & 2 for PD Control with unknown payload');
ylabel('e_{1} (rad) ')
subplot(212),plot(t,err2);grid;
ylabel('e_{2} (rad)')
xlabel('Time, t (second)');

figure;
subplot(211),plot(t,ev1);grid;
title('Relative error for link 1 & 2 for PD Control with unknown payload');
ylabel('\sigma_{1} (rad) ')
subplot(212),plot(t,ev2);grid;
ylabel('\sigma_{2} (rad) ')
xlabel('Time, t (second)');

figure;
subplot(211),plot(t,tau1);grid;
title('Input torque for link 1 & 2 for PD Control with unknown payload');
ylabel('\tau_{1} (N.m) ')
subplot(212),plot(t,tau2);grid;
ylabel('\tau_{2} (N.m)')
xlabel('Time, t (second)');

figure;
subplot(211),plot(t,derr1);grid;
title('Velocity error for link 1 & 2 for PD Control with unknown payload');
ylabel('dot{e} (rad/s) ')
subplot(212),plot(t,derr2);grid;
ylabel('dot{e}_{2} (rad/s)')
xlabel('Time, t (second)');
```

VITA

Kiu Ling Pau

Candidate for the Degree of

Master of Science

Thesis: ROBUST ADAPTIVE CONTROL OF TIME-VARYING MECHANICAL SYSTEMS: ANALYSIS AND EXPERIMENTS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Sibu, Malaysia on April 24, 1973, the son of Ngie Siong Pau and Suk Yun Yong.

Education: Graduated from Tung Hua Secondary School, Sibu, Malaysia in December, 1991; attended Inti College, Kuala Lumpur, Malaysia until April, 1994; received the Bachelor of Science degree in Mechanical Engineering from Oklahoma State University, Stillwater, Oklahoma in August, 1996; completed the requirements for the Master of Science degree in Mechanical Engineering at Oklahoma State University in December, 1998.

Professional Experience: Experimental setup of the time-varying mass system and conducting experiment on NSK two links SCARA robot.

Professional Memberships: American Society of Mechanical Engineers.

ISA