Adaptive Control of Robot Manipulators With Uncertain Variable Parameters

A Thesis

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fullfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Yulin Qin

August 2018

APPROVAL SHEET

This thesis is submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Electrical Engineering

Yulin Qin, Author

This thesis has been read and approved by the examining committee:

Advisor: Gang Tao

Committee Member: Scott T. Acton

Committee Member: Carl R. Knospe

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

August, 2018

Acknowledgment

I would first like to thank my these advisor Professor Gang Tao of the School of Engineering and Applied Science at University of Virginia. His patient guidance, support and encouragement provide me directions when I was caught in confusions. His wisdom and deep knowledge of control theories always inspire me during my research and study in University of Virginia.

I would also like to thank my friend Wuwei and my family. Without their support, encouragement and accompanies, I could not make this accomplishment.

Abstract

The control of robot manipulators has become a hot topic in these years. With increasing usage of robots in military industry, manufacturing, service industry and daily entertainments for common people, there is an increasing need to design robots with higher performance and various functions. Adaptive control is powerful in solving control problems with uncertainties, and thus become a potent tool in this area. This thesis studies the adaptive control of robot manipulator systems with uncertain and time-varying parameters. A new parametrization scheme is derived to expand the capacity of adaptive control in dealing with such parameter uncertainties. Unlike the existing control methods, each parameter is not estimated by a single estimate, but a group of estimates, which can help robot manipulators perform better in time-varying environments. This control algorithm can guarantee stability and asymptotic tracking ability, despite large and persistent uncertain parameter variations. Compared with classical control methods, the new adaptive control designs will help reduce the effect of the uncertainties of time-varying parameters in the robot working process. Simulation results of control for a planar elbow manipulator, a typical type of robot manipulators, are presented to verify our control performance.

Contents

1 Introduction **1**

1.1 Research Motivation 1

1.2 Literature Review 2

1.3 Thesis Outline 6

2 Background **7**

2.1 Signal Measures 7

2.2 System Stability 10

2.3 Adaptive Control 12

3 Modeling of Robot Manipulators **14**

3.1 Manipulator Dynamic Equations 14

3.1.1 Manipulator Configuration 15

3.1.2 Manipulator Jacobian Matrix 15

3.1.3 Kinetic Energy and Potential Energy 18

3.1.4 Euler-Lagrange Equations 19

3.1.5 Dynamic Equations and Properties 21

3.2 Examples 22

3.2.1 Single-Link Manipulator 22

3.2.2 Two-Link Planar Manipulator 24

3.3 Summary 27

4 Control of Manipulators 28

4.1 Designs with Joint Acceleration Measurements 28

4.1.1 Control Designs 31

4.2 Designs without Joint Acceleration Measurements 35

4.2.1 Nominal Control System 36

4.2.2 Adaptive Control System 38

4.3 Summary 39

5 Adaptive Control Design With Uncertain Variable Parameters 40

5.1 Dynamic Model and Control Problem 41

5.1.1 System with Parameter Variations 41
5.1.2 System Parametrization 44
5.2 Controller Structure and Tracking Error Equation 47
5.2.1 Controller Structure 48
5.2.2 Tracking Error Equation 50
5.3 Nominal Control System 53
5.3.1 Stability Analysis 53
5.3.2 Simulation Study 54
5.4 Adaptive Control System 62
5.4.1 Stability Analysis 63
5.4.2 Simulation Study 64
5.5 Comparison Study 73
5.6 Summary 75

6 Conclusions and Future Topics **77**

6.1 Summary and Conclusions 77
6.2 Future Research Topics 78

I Bibliography 80

List of Figures

3.1 Single-link robot [38] 22

3.2 Two-link planar elbow robot [38] 24

5.1 An example of working process 41

5.2 Planar elbow manipulator [38] 42

5.3 Position tracking error (Case I: nominal control) 56

5.4 Velocity tracking error (Case I: nominal control) 56

5.5 System parameters $\theta^*(t)$ (Case I: nominal control) 57

5.6 Control inputs (Case I: nominal control) 57

5.7 Position tracking error (Case II: nominal control) 59

5.8 Velocity tracking error (Case II: nominal control) 60

5.9 System parameters $\theta^*(t)$ (Case II: nominal control) 60

5.10 Control inputs (Case II: nominal control) 61

5.11 System parameters $\theta^*(t)$ (Case I: adaptive control) 66

5.12 Joint angles contrast with desired joint angles (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 66

5.13 Joint velocities contrast with desired joint velocities (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 67

5.14 Position tracking error (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) . . . 67

5.15 Velocity tracking error (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) . . . 68

5.16 Control input u(t) (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 68

5.17 System parameters $\theta^*(t)$ (Case II: adaptive control) 70

5.18 Joint angles contrast with desired joint angles (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 71

5.19 Joint velocities contrast with desired joint velocities (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 71

5.20 Position tracking error of adaptive control (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 72

5.21 Velocity tracking error of adaptive control (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 72

5.22 Control input u(t) (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 73

5.23 Position tracking error (Case I: the existing adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 74

5.24 Velocity tracking error (Case I: the existing adaptive control, $\theta_i(0) = 80\%\theta_i^*$) 75

List of Tables

5.1 Parameters of Case I (nominal control) 55

5.2 Parameters of Case II (nominal control) 59

5.3 Parameters of Case I (adaptive control) 65

5.4 Parameters of Case II (adaptive control) 69

Chapter 1

Introduction

The control of robot manipulators has become a heated topic in these years. With increasing usage of robots in military industry, manufacturing, service industry and daily entertainments for common people, there are huge needs to design robots with higher performance and various functions. In this chapter, we will give an introduction of current research progress of robot manipulator control.

1.1 Research Motivation

In past decades, with the great development of computer science, composites, and processing technology, the robot has been given with higher performance, which brings robot broad prospect. Therefore, many research works have been done in robot manipulator control. Researchers did many attempts like PD control, robust control and adaptive control. Some also did independent joint control, using PID compensator, state feedback control and feedforward control [38].

When the robot is applied to more fields, there is an increasing need of robots having the capacity to work in complex and unknown conditions. Because in practical applications, there exists various uncertainties and external disturbances. Recently, many researchers begin to study this problem in various conditions and aspects.

Consider a robot manipulator work in a set working procedure. The existing control designs can only stabilize the system, but cannot reduce the disturbance caused by switching working process. So the peak of disturbance remains the same in multiple-time repeating works. It is not good for material fatigue, service life, and operation accuracy. In our research, the objective is to present a new approach of adaptive control design, which can solve conditions with uncertain system parameters and gain better performance.

1.2 Literature Review

In current robot manipulators relative literature, most works were done for systems with time-invariant parameters. In [10,34,37], the most basic design of adaptive control for robot manipulators. The authors presented an adaptive version of the computed torque method for the control of manipulators with rigid links and proposed an adaptive computed torque controller using disturbance accommodation control techniques. They derived a new adaptive robot control algorithm, which consists of a PD feedback part and a full dynamics feed-forward compensation part, with the unknown manipulator and payload parameters being estimated online. In [9], the author presented the study of an adaptive control which tracks a desired time-based trajectory as closely as possible for all times over a wide range of manipulator motion and payloads both in joint-variable coordinates and Cartesian coordinates. In [3], the author presented a sliding-mode control based on a Variable Structure System for a multijoint manipulator. The practical sliding-mode controller, which has a simple nonlinear compensator and proper continuous function. In [15], the author presented a new scheme for the adaptive control of mechanical manipulators which does not require the measurement of joint accelerations and needs less computation. In [17], the author demonstrated the approach on a high-speed two degree of freedom semi-direct-drive robot. It showed that the dynamic parameters of the manipulator trajectory can be precisely controlled. In [18], they gave a tutorial account of several of the most recent adaptive control results for rigid robot manipulators. In [20], the article addressed an application that involves the adaptive control of robot manipulator joint. It tries to explore the potential of using soft computing methodologies in control of plant with unknown internal behavior and environmental changes. In [24], the author introduced two reduced and unconstrained robot models. In [25], they proposed an algorithm for the collision avoidance. In [4], they proposed a non-quadratic stabilization approach to stabilizing a two-link robot manipulator. A non-quadratic Lyapunov function is used as a fuzzy blending of multiple quadratic Lyapunov functions. In [28], it presented an approach for robot manipulator control and learning. When a robot manipulator is controlled to track a periodic reference orbit, the locally-accurate approximation of the closed-loop control system dynamics can be achieved in a local region along the periodic orbit. Then in [27], the author proposed a method for learning and controlling an industrial robot manipulator through fuzzy voice commands guided by visual motor coordination. In [11], the author studied the adaptive state feedback for state tracking control problem for such systems. In [35], the author developed a control of a redundant robot manipulator. That has to carry out a trajectory tracking in operational space while avoiding an obstacle. In [23], the author estimated the gravity force using the generalized gravity regressor which is regardless of the dimension and structure of the robot under the quasi-static state. [26], the author provided a linear time-varying approximate model around some desired manipulator trajectory, to the robot manipulator. In [1], the author presented a comparative study of two control approaches; fuzzy proportional derivative control and the sliding mode control for a two degree of freedom robot manipulator. It presented the implementation of both models based and model free based control. In [2], the author proposed an adaptive control for a robot manipulator, which allowed the displacement of the end-effector following a reference trajectory. In [30], the author proposed a discrete-time adaptive control scheme for controlling robot manipulators. In [32], the article targeted the case that the built-in controller does not provide desirable precision for set-point regulation.,In [6], this paper presented an adaptive backstepping control scheme for a mobile manipulator robot based on the virtual decomposition control. The control scheme was applied on three degrees of freedom manipulator arm mounted on two degree of freedom mobile platform to track the desired workspace trajectory. The desired joint space trajectory was obtained by using the inverse kinematics. [8] proposed a new method to improve the transient performance of the adaptive tracking control system for robot manipulators. It can guarantee the exponential convergence to a predetermined residual set of tracking error in the closed loop system.

Some works for systems with uncertain parameters were also done. [16] presented an adaptive robust controller for robot manipulators using adaptive integral sliding mode control and time-delay estimation. In [29], the author proposed direct and indirect model reference adaptive control strategies for multivariable piecewise affine systems, which constitute a popular tool to model hybrid systems and approximate nonlinear systems. [31] presented dynamic learning from adaptive neural control with prescribed tracking error performance for an n-link robot manipulator subjected to unknown system dynamics and external disturbance. In [22], the authors developed an adaptive tracking control law for a class of uncertain nonlinear systems with Markovian jumping parameters. In [21], researchers proposed a systematic adaptive sliding mode controller design for the robust control of nonlinear systems with uncertain parameters. In [14], the authors designed a hybrid impedance controller. With some robustness against the uncertainties of the environment, the approach is able to implement the desired contact force and track the commanded position in orthogonal subspace without knowing the accurate model of the environment. In [19], the authors studied the position tracking control with finite-time convergence for a class of nonlinear uncertain robot manipulators. It designed a Radial basis function neural network (RBFNN) based adaptive control to compensate for the effect of the unknown dynamics. In [33], it presented control designs for tracking control of robot system with uncertain manipulator dynamics and joint actuator dynamics subject to constrained task space. In [5], they proposed a new dynamic prediction error based adaptive controller for robotic manipulators with uncertain parameters. A multiple-model adaptive control scheme was developed using multiple prediction errors and multiple controllers, incorporated with multiple parameter estimators and a control switching mechanism to find the model that best approximates the manipulator dynamics. In [13], the author proposed two adaptive control schemes to realize the objective of task-space trajectory tracking irrespective of the uncertain kinematics and dynamics. The proposed controllers have the desirable separation property, and they have shown that the first adaptive controller with appropriate mode fictions can yield the improved performance.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, the background of control is given, which includes definitions and theorems of stability, and concepts of adaptive control. In Chapter 3, modeling of robot manipulators is introduced, in order to give readers a basic and detailed process of building robot manipulators model. Then, two common models of robot manipulator are given, which are also used for simulation in the following chapters. In Chapter 4, we introduce control methods for constant system parameters, so as to help readers gain better understand in later chapters. In Chapter 5, the model of robot manipulators with time-varying and control method with joint acceleration measurement is given. Simulation results for nominal control and adaptive control were offered. We also show a comparison work of previous control methods. For all simulation works above, we use two-link planar manipulator to do the corresponding simulations, and the results illustrate the effectiveness of the adaptive control for jumping parameters. Chapter 6 will discuss the conclusions and some future works.

Chapter 2

Background

Before going further about robot manipulator control, the background information of control should be introduced. In this chapter, concepts of stability analysis ans adaptive control will be introduced.

2.1 Signal Measures

The l^q norm of a constant vector $x = (x_1, ..., x_n)^T \in R^n$ with n finite is defined as

$$\| x \|_1 = | x_1 | + \cdots | x_n | \tag{2.1}$$

the l^2 norm of x is defined as

$$\| x \|_2 = \sqrt{x_1^2 + \cdots + x_n^2} \tag{2.2}$$

and the l^∞ norm is defined as

$$\| x \|_\infty = \max_{1 \le i \le n} | x_i | \tag{2.3}$$

For a vector signal $x(t) = (x_1(t), \cdots, x_n(t))^T \in R^n$, we define norms

$$\begin{aligned} \| x(t) \|_2 &= \sqrt{x_1^2(t) + \cdots + x_n^2(t)} \\ \| x(t) \|_1 &= | x_1(t) | + \cdots + | x_n(t) | \\ \| x(t) \|_\infty &= \max_{1 \le i \le n} | x_i(t) | \end{aligned} \tag{2.4}$$

The induced norm of a constant matrix $A \in R^{m*n}$ is defined as

$$\| x(t) \| = \sup_{x \in R^n, \|x\|=1} \| A(t)x \| \tag{2.5}$$

The induced matrix norms are defined as

$$\begin{aligned} \| A \|_\infty &= \max_{1 \le i \le m} \sum_{j=1}^{n} | a_{ij} | \\ \| A \|_1 &= \max_{1 \le i \le m} \sum_{j=1}^{m} | a_{ij} | \\ \| A \|_2 &= \sqrt{\lambda_m ax(A^T A)} \end{aligned} \tag{2.6}$$

Consider a vector signal $x(t) \in R^n$, the L^1 norm is defined as

$$\| x(\cdot) \|_1 = \int_0^\infty \| x(t) \|_1 \, dt = \int_0^\infty (| x_1(t) | + \cdots + | x_n(t) |) dt \tag{2.7}$$

The L^2 norm is defined as

$$\| x(\cdot) \|_2 = \sqrt{\int_0^{\infty} \| x(t) \|_2^2 \, dt} = \sqrt{\int_0^{\infty} (| x_1^2(t) | + \cdots + | x_n(t) |) dt} \tag{2.8}$$

and the L^{∞} norm is defined as

$$\| x(\cdot) \|_2 = \sup_{t \geqslant 0} \| x(t) \|_{\infty} = \sup_{t \geqslant 0} \max_{1 \leq i \leq n} | x_i(t) | \tag{2.9}$$

Signal space L^1 is defined as

$$L^1 = \{ x(t) \in R^n : \| x(\cdot) \|_1 < \infty \} \tag{2.10}$$

signal space L^2 is defined as

$$L^2 = \{ x(t) \in R^n : \| x(\cdot) \|_2 < \infty \} \tag{2.11}$$

and the signal space L^{∞} is defined as

$$L^{\infty} = \{ x(t) \in R^n : \| x(\cdot) \|_{\infty} < \infty \} \tag{2.12}$$

The signal $x(t)$ is uniformly bounded if $x(t) \in L^{\infty}$, for some $\beta > 0$, $\| x(t) \| < \beta$, $\forall t \geqslant 0$. For the signal norms L^1, L^2, L^{∞}, we have: $x(t) \in L^1 \cap L^{\infty} \Longrightarrow x(t) \in L^2$

2.2 System Stability

The concept of stability is significant to control system design. The methods available to examine the poles depend on the representation of the system model. If the classical approach is taken then the poles of the transfer function can be examined. If the modern approach is used then the eigenvalues, which are the poles, of the system matrix A can be analyzed. Either approach can quickly give information on whether or not the system is inherently stable, marginally stable, or unstable.

For adaptive control system stability must be defined another way since knowledge of the system parameters are unavailable and possibly changing. The work of Alexander Mikhailovich Lyapunov, who presented definitions and theorems for studying the stability of solutions to a broad class of differential equations, has been used extensively to address this problem [40]. The work of Lyapunov relies on defining an energy function, formally known as a Lyapunov function candidate, that can be used to determine the stability of a system without having to solve for the solutions to the system explicity. Originally, this Lyapunov function was purely the total mechanical or electrical energy and therefore by nature positive definite. The Lyapnov indirected method can be found in many textbooks about a nonlinear system like [39], [41].

Definition 2.1 *The response of $\dot{x}(t) = Ax(t)$ is stable in the sense of Lyapunov if every finite initial state x_0 excites a bounded response. It is asymptotically stable if every finite state excites abounded response which, in addition, approaches 0 as $t \longrightarrow \infty$ [42].*

Theorem 1 *The equation $\dot{x}(t) = Ax(t)$ is stable if and only if all eigenvalues of A*

have zero or negative real parts and those with zero real parts are simple roots of the minimal polynomial of A. The equation $\dot{x}(t) = Ax(t)$ is asymptotically stable if and only if all eigenvalues of A have negative real parts.

Also in the Lyapunov sense, we can check the stability of matrix A by Lyapunov theorem.

Theorem 2 *The equation $\dot{x} = Ax, A \in R^{n*n}, x \in R^n$ is asymptotically stable if and only if for every positive definite $Q = Q^T \in R^{n*n}$, the Lyapunov equation $A^T A + PA = -Q$ has a unique and positive definite solution $P = P^T \in R^{n*n}$.*

Theorem 1 and 2 are theorems that we usually use to check the stability of close-looped system by classical control. But for adaptive control, those theorems would not work since in adaptive control, there exists uncertainty on the dynamics model, we cannot get the accurate system parameters. So we introduce a new method called Lyapunov direct method to check the system stability when applying adaptive control.

Theorem 3 *(Lyapunov direct method) If in some Ball $B(h)$, there is a positive definite function $V(x,t)$ with $V \leqslant 0$, then the equilibrium state $X_e = 0$ of the system $\dot{x} = Ax$ is stable. If we also have "V(x,t)" is decrescent, then it is uniformly stable.*

Then, we introduce the Barbălart lemma. This lemma makes the precess to analyze system stability more easier.

Lemma 2.1 *(Barbălat Lemma) If a scalar function $f(t)$ satisfies $\dot{f}(t) \in L^{\infty}$, $f(t) \in L^2$, then* $\lim_{t \longrightarrow \infty} \dot{f}(t) = 0$.[39]

2.3 Adaptive Control

Adaptive control applies physical adaptation dynamics that can adjust the controller for a system with uncertainties to reach the desired performance. Different from the classical control method, the adaptive control systems can achieve more possible system operation that is much more flexible.

Direct adaptive control The first ways to adaptive control design is considered as the direct adaptive control. It generates the updated system parameters with the adaptive controller and the adaptive laws without thinking the initial conditions of the target plant and possible disturbances. In a direct adaptive control system, the controller parameters, which are estimates of some unknown ideal plant-model controller matching parameters, are directly updated from adaptive laws, based on our defined tracking error.

Indirect adaptive control The other one is called indirect adaptive control. This method is to estimate plant parameters with the possible disturbances, and update the adaptive controller with the designed adaptive laws. In an indirect adaptive control system, the controller parameters, which are also estimates of some unknown ideal plant-model matching parameters, are simultaneously calculated from a design equation using the on-line estimates of the unknown plant parameters, updated from a parameter estimator based on an estimation error presenting the mismatch between the plant output and its estimated version generated from the parameter estimation.

In either a direct or an indirect adaptive control design, it is significant to use the estimates of some unknown parameters of an ideal controller. The parameter estimates are obtained from an adaptive updated law driven by system performance error, and used in implementing an adaptive controller either directly or through a design equation to map the plant parameter estimates to the controller parameters. More information can be found in [39].

Chapter 3

Modeling of Robot Manipulators

This chapter will discuss the procedure of deducting general dynamic equations of robot manipulators. First, we will introduce the way of getting the Jacobian matrix and Euler-Lagrange equation. Then, some properties of dynamic equations will be discussed. Finally, we will show two classic examples and offer detailed dynamic equations.

3.1 Manipulator Dynamic Equations

We first show the complete derivation procedure of general robot manipulator models, including Jacobian matrix, kinetic an potential energy, Euler-Lagrange equations. At the end of this section, we will also introduce several vital properties of robot manipulator dynamic equations, which will be used in following chapter for control design.

3.1.1 Manipulator Configuration

In this research, the object is robot manipulator with n degrees of freedom (DOF). Each link of the manipulator is regarded as rigid body. The configuration of robot manipulator includes the location information of every points. We use joint variables $\mathbf{q} = (q_1, q_2, ...q_n)$, where $\mathbf{q}$ stands for relative rotation (revolute joint) and relative motion (prismatic joint) between every two links. Robot manipulator in three-dimensional space needs at least six indepent DOF, which are three axial DOF and thee rotations DOF. Otherwise, there will be a restriction of robot manipulator's movement. And if there exists extra DOF, it will become redundant system. Common industrial robot manipulators have less than six DOF, such as articulated manipulator (three revolute joints), cylindrical manipulator (one revolute joint and two prismatic joints) and so on.

3.1.2 Manipulator Jacobian Matrix

Mathematically, the velocity relationships of each links of robot manipulators are determined by the Jacobian of the function that defined by kinematic equations. The Jacobian is a matrix that generalizes the notion of the ordinary derivative of a scalar function. The Jacobian is one of the most significant quantities in the study of robot motion. It is vital in every aspect of robotic manipulation: in the planning and execution of smooth trajectories, in the derivation of the dynamic equations of motion, and in the transformation of forces and torques from the end effector to the manipulator joints.

Derivation of Jacobian Matrix

Consider an n-link manipulator with joint variables $q_1, \cdots, q_n$. Let

$$T_n^0(q) = \begin{pmatrix} R_n^0(q) & o_n^0(q) \\ 0 & 1 \end{pmatrix} \tag{3.1}$$

denote the transformation from the end-effector link to the base link, where $q = [q_1, ..., q_n]$ is the vector of joint variables. As the robot moves about, both the joint variables q_i and the end-effector position o_n^0 and orientation R_n^0 will be functions of time. We will introduce the way to relate the linear and angular velocity of the end effector to the vector of joint velocities $\dot{q}(t)$. Let

$$S(\omega_n^0) = \dot{R}_n^0 (R_n^0)^T \tag{3.2}$$

where ω_n^0 is the angular velocity vector of the end effector, and define

$$\upsilon_n^0 = \dot{o}_n^0 \tag{3.3}$$

as the linear velocity of the end effector. We want to get the expressions of the form

$$\begin{aligned} \upsilon_n^0 &= J_\upsilon \dot{q} \\ \omega_n^0 &= J_\omega \dot{q} \end{aligned} \tag{3.4}$$

where J_v and J_ω are $3 \times n$ matrices. We will introduce them in the next parts. By writing two equations in Equation (3.4) together, we get

$$\xi = J\dot{q} \tag{3.5}$$

where ξ and J are given by

$$\xi = \begin{pmatrix} v_n^0 \\ \omega_n^0 \end{pmatrix} \; and \; J = \begin{pmatrix} v_n^0 \\ \omega_n^0 \end{pmatrix} \tag{3.6}$$

The vector ξ is also called the body velocity. Especially, this velocity vector is not the derivative of a position variable, because the angular velocity vector is not the derivative of any time-varying quantity. The matrix J is called the Jacobian. J is a $6 \times n$ matrix where n is the number of links.[38]

Angular Velocity

The overall angular velocity of the end effector, ω_n^0, can be expressed as

$$\omega_n^0 \;=\; \rho_1 \dot{q}_1 k + \rho_2 \dot{q}_2 R_1^0 k + ... + \rho_n \dot{q}_n R_{n-1}^0 k \;=\; \sum_{i-1}^{n} \rho_i \dot{q}_i z_{i-1}^0 \tag{3.7}$$

where ρ_i equals to 1 when the i_{th} joint is revolute and 0 when the i_{th}c joint is prismatic, since

$$z_{i-1}^0 \;=\; R_{i-1}^0 k \tag{3.8}$$

Thus, J_ω in Equation (3.4) is expressed as

$$J_\omega = [\rho_1 z_0 ... \rho_n z_{n-1}] \tag{3.9}$$

Linear Velocity

We use notation $\dot{o}_n^0$ to represent the linear velocity of the end effector. Based on the chain rule for differentiation, we can get

$$\dot{o}_n^0 = \sum_{i=1}^{n} \frac{\partial o_n^0}{\partial q_i} \dot{q}_i \tag{3.10}$$

Then, we can express the $i_t h$ column of J_v as

$$J_{vi} = \frac{\partial o_n^0}{\partial q_i} \tag{3.11}$$

Now, we are ready to use the Jacobian matrix to derive dynamic equations of the robot manipulator. There are more information of the Jacobian matrix in [38]

3.1.3 Kinetic Energy and Potential Energy

For the robot manipulator with one rigid body and six DOF, the kinetic energy is

$$K = \frac{1}{2} m v^T v + \frac{1}{2} \omega^T I \omega \tag{3.12}$$

where m is mass of the rigid body, vectors v and ω are linear velocity and angular velocity, and I is inertia tensor of the rigid body.

For n-link robot manipulator (n rigid body links), the kinetic energy is

$$K = \frac{1}{2}\dot{q}^T[\sum_{i=1}^{n} m_i J_{vi}(q)^T J_{vi}(q) + J_{\omega i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega i}(q)]\dot{q} \tag{3.13}$$

To simplified the equation, we use

$$D(q) = \sum_{i=1}^{n} m_i J_{vi}(q)^T J_{vi}(q) + J_{\omega i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega i}(q) \tag{3.14}$$

Where $D(q)$ is called inertia matrix, which is symmetric and positive definite. Thus, the equation becomes

$$\begin{aligned} K &= \frac{1}{2}\dot{q}^T D(q)\dot{q} \\ &= \frac{1}{2}\sum_{i,j} d_{i,j}(q)\dot{q}_i\dot{q}_j \end{aligned} \tag{3.15}$$

where q is joint variables, m_i is total mass of each link,

For n-link robot manipulator, the potential energy is

$$P = \sum_{i=0}^{n} m_i g^T r_{ci} \tag{3.16}$$

where m_i is total mass of each link, g is gravitational acceleration, and r_{ci} is the vector from origin to the center of mass of each link.

3.1.4 Euler-Lagrange Equations

The standard form of Euler-Lagrange equation is

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = u_i \quad i = 1, ..., n \tag{3.17}$$

where K is total kinetic energy, P is total potential energy, and L is named as Lagrangian of the system.

$$L \;=\; K - P \;=\; \frac{1}{2}\sum_{i,j} d_{i,j}(q)\dot{q}_i\dot{q}_j - P(q) \tag{3.18}$$

For Euler-Lagrange equation, we have

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} &= \sum_j d_{kj}\ddot{q}_j \;+\; \sum_{i,j}\frac{\partial d_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j \\ \frac{\partial L}{\partial q_k} &= \frac{1}{2}\sum_{i,j}\frac{\partial d_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j \;-\; \frac{\partial P}{\partial q_k} \end{aligned} \tag{3.19}$$

Then, the Euler-Lagrange equations become

$$\sum_j d_{kj}\ddot{q}_j \;+\; \sum_{i,j}[\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}]\dot{q}_i\dot{q}_j - \frac{\partial P}{\partial q_k} \;=\; u_k \tag{3.20}$$

To simplify the equations, we define

$$\begin{aligned} c_{ijk} &= \frac{1}{2}[\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_i} - \frac{\partial d_{ij}}{\partial q_k}] \\ &= \frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k} \\ g_k &= \frac{\partial P}{\partial q_k} \end{aligned} \tag{3.21}$$

Thus, we can rewrite the Euler-Lagrange equations as

$$\sum_i d_{kj}(q)\ddot{q}_j + \sum_{i,j} c_{ijk}(q)\dot{q}_i\dot{q}_j + g_k(q) \;=\; u_k, \quad k = 1, 2..., n \tag{3.22}$$

And the matrix form of Euler-Lagrange equations is written as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) \;=\; u \tag{3.23}$$

3.1.5 Dynamic Equations and Properties

The dynamic equations of n-link robot manipulator are very complex. However, there exists some properties that can benefit us to develop control algorithms. We introduce two of the most common properties that will be used in our research.

Skew Symmetry Property The matrix $S(q,\dot{q}) = \dot{D}(q) - 2C(q,\dot{q})$ is skew symmetric, that is the elements in matrix S satisfy $s_{jk} = -s_{kj}$

Proof: With given inertia matrix $D(q)$, we can get the expression of elements in $\dot{D}(q)$

$$\dot{d}_{kj} = \sum_{i=1}^{n} \frac{\partial d_{kj}}{\partial q_i}\dot{q}_i \tag{3.24}$$

With the definition of $C(q,\dot{q})$ and symmetric property of $D(q)$, we can get the elements of $S(q,\dot{q})$

$$\begin{aligned} s_{kj} &= \dot{d}_{kj} - 2c_{kj} \\ &= \sum_{i=1}^{n}[\frac{\partial d_{ij}}{\partial q_k} - \frac{\partial d_{ki}}{\partial q_j}]\dot{q}_i \\ &= -s_{kj} \end{aligned} \tag{3.25}$$

Thus, $S(q,\dot{q}) = \dot{D}(q) - 2C(q,\dot{q})$ is skew symmetric.[38]

Linearity in the Parameters By sorting the parameters in the dynamic equations of robot manipulator, we can get a linear equations, which relieve our work greatly, that is

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau = Y(q,\dot{q},\ddot{q})\Theta \tag{3.26}$$

Where $Y(q, \dot{q}, \ddot{q})$ is named as regressor, which contains information of joint variables and is assumed to be completely known. And Θ is the parameter vector, which contains information of system structure parameters. Two specific examples are given in the next section.

3.2 Examples

In this section, we will introduce two typical examples of robot manipulator configuration. In later chapters, we will take the model of two-link planar manipulator to show our control methods and simulation results.

3.2.1 Single-Link Manipulator

At this time, we consider a single-link robot manipulator with 1 DOF (one revolute joint), for instance, DC-motor. In Figure 3.1, it is a single-link robot arm with a gear train to a DC motor.

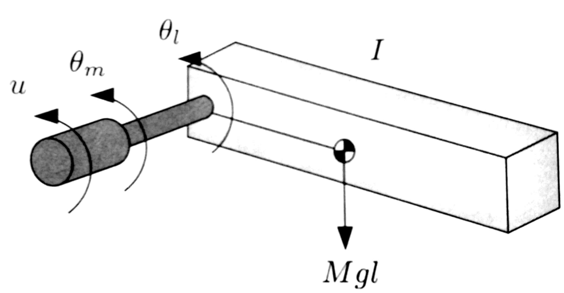

Figure 3.1: Single-link robot [38]

According to last section, we first list the equations of kinetic energy, potential energy and Langrangian of the system. Using angular variable of the link as the join variable. Then, we can get

$$\begin{aligned} K &= \frac{1}{2}(r^2 J_m + J_l)\dot{\theta}^2 \\ &= \frac{1}{2}J\dot{\theta}^2 \end{aligned} \tag{3.27}$$

Where θ is link angle, r is the gear ratio, J_m and J_l are inertia of motor and link respectively. And the potential energy is

$$P = Mgl(1 - \cos\theta) \tag{3.28}$$

Thus, Lagrangian is

$$L = K - P = \frac{1}{2}J\dot{\theta}^2 - Mgl(1 - \cos\theta) \tag{3.29}$$

Then, we can get Euler-Lagrange equations

$$J\ddot{\theta} + Mgl\sin\theta = \tau \tag{3.30}$$

In this example, τ includes motor torque and damping torques of motor and link, that is

$$\begin{aligned} \tau &= u - (rB_m + B_l)\dot{\theta} \\ &= u - B\dot{\theta} \end{aligned} \tag{3.31}$$

Where u is motor torque, B_m and B_l are motor and link damping respectively. Thus, the dynamic equations become

$$J\ddot{\theta} + B\dot{\theta} + Mgl\sin\theta \;=\; u \tag{3.32}$$

3.2.2 Two-Link Planar Manipulator

Two-link robot manipulators are the most common robot manipulators. It is very meaningful to take it as the model for the study of manipulator control. In Figure 3.2, this type of two-link robot arm is called planar elbow robot manipulator, which is a typical model of two-link robots.

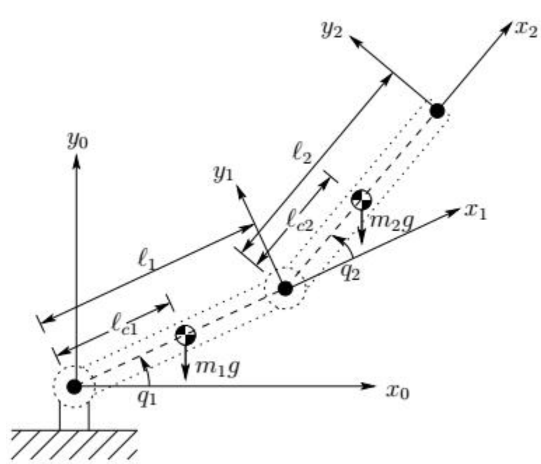

Figure 3.2: Two-link planar elbow robot [38]

As shown in Figure 3.2, q_1 and q_2 are joint variables (revolute joints), l_1 and l_2 are length of links, l_{c1} and l_{c2} indicate the positions of the center of mass. We start with giving the kinetic energy.

$$K = \frac{1}{2}\sum_{i=1}^{2} m_i v_i^T v_i + \omega_i^T I_i \omega_i \tag{3.33}$$

where

$$\begin{aligned} v_i &= J_{v_i}(q)\dot{q} \\ \omega_i &= J_{\omega_i}(q)\dot{q} \end{aligned} \tag{3.34}$$

We can get the Jacobian matrix

$$J_{v_{c1}} = \begin{pmatrix} -l_c \sin q_1 & 0 \\ l_{c1}\cos q_1 & 0 \\ 0 & 0 \end{pmatrix} \tag{3.35}$$

$$J_{v_{c2}} = \begin{pmatrix} -l_1 \sin q_1 - l_{c2}\sin(q_1+q_2) & -l_{c2}\sin(q_1+q_2) \\ l_1\cos q_1 + l_{c2}\cos(q_1+q_2) & l_{c2}\cos(q_1+q_2) \\ 0 & 0 \end{pmatrix} \tag{3.36}$$

After simplifying the equations, we can write kinetic energy in the form

$$K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \tag{3.37}$$

where

$$D(q) = m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}} + \begin{pmatrix} I_1 + I_2 & I_2 \\ I2 & I_2 \end{pmatrix}$$

$$= \begin{pmatrix} m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}^2 + 2l_1 l_{c2}\cos q_2) + I_1 + I_2 & m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2 \\ m_2(l_{c2}^2 + l_1 l_{c2}\cos q_2) + I_2 & m_2 l_{c2}^2 + I_2 \end{pmatrix} \tag{3.38}$$

Then, for Christoffel symbols, by using the formula

$$c_{kj} = \sum_{i=1}^{2} \frac{1}{2}(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}) \tag{3.39}$$

We can get

$$C(q, \dot{q}) = \begin{pmatrix} h\dot{q}_2 & (\dot{q}_1 + \dot{q}_2)h \\ -\dot{q}_1 h & 0 \end{pmatrix} \tag{3.40}$$

where

$$h = -m_2 l_1 l_{c2} \sin q_2 \tag{3.41}$$

Next, the potential energy is

$$P = P_1 + P_2 = m_1 g l_{c1} \sin q_1 + m_2 g(l_1 \sin q_1 + l_{c2} \sin(q_1 + q_2)) \tag{3.42}$$

Then, we can get $\phi(q)$

$$g(q) = \frac{\partial P}{\partial q} = \begin{pmatrix} (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2) \\ m_2 l_{c2} g \cos(q_1 + q_2) \end{pmatrix} \tag{3.43}$$

Thus, the dynamic equation will be

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \;=\; u \tag{3.44}$$

3.3 Summary

In this chapter, we introduced the modeling of robot manipulators. First, we discussed the derivation of the Jacobian matrix, which is vital in modeling. Then, we use the Jacobian matrix to get the kinetic energy and potential energy, which consist the dynamic equations. In the last section, we provide two typical examples, a single-link robot arm and a two-link robot arm, to derive the dynamic equations of robot manipulators

Chapter 4

Control of Manipulators

In this chapter, we will discuss inverse dynamic control and passivity-based motion control. The former one is also regarded as the control method with joint acceleration measurements, while the latter one does not require joint acceleration measurements.

4.1 Designs with Joint Acceleration Measurements

Now, we want to find the application of more complex nonlinear control techniques for trajectory tracking of rigid manipulators. According to [38], we introduce the first control design. We know the dynamic equations of an n-link robot in matrix form is

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} \;+\; g(q) \;=\; u \tag{4.1}$$

The idea of inverse dynamics is to find a nonlinear feedback control law

$$u \; = \; f(q, \dot{q}, t) \tag{4.2}$$

which when substituted into the equation above, results in a linear closed loop system. For general nonlinear systems, this control law may be too difficult to find. As for the manipulator dynamic equations, the problem is much easier. We can see that if we choose the control u according to the equation

$$u \; = \; D(q)a_q \; + \; C(q, \dot{q})\dot{q} + g(q) \tag{4.3}$$

Then, with the knowledge that the inertia matrix D is invertible, the combined system reduces to

$$\ddot{q} \; = \; a_q \tag{4.4}$$

The term a_q represents a new input to the system which is yet to be chosen. Each input a_{qk} can be designed to control a scalar linear system. Meanwhile, assuming that a_{qk} is a function only of q_k and its derivatives, then a_{qk} will affect q_k independently of the motion of the other links.

Now, q_k can be designed to control a linear second order system, the obvious choice is to set

$$a_q \; = -K_0 q \; - \; K_1 \dot{q} \; + \; r \tag{4.5}$$

where K_0 and K_1 are diagonal matrices with diagonal elements consisting of position and velocity gains. The closed loop system becomes the linear system

$$\ddot{q} \; + \; K_1\dot{q} \; + K_0 q \; = \; r \tag{4.6}$$

If one chooses the reference input $r(t)$ as $\ddot{q}^(t) \, + \, K_0 q_d(t) \, + \, K_1 \dot{q}_d(t)$, then the tracking error $e(t) = q - q_d$ satisfies

$$\ddot{e}(t) \; + \; K_1 e(t) \; K_0 e(t) \; = \; 0 \tag{4.7}$$

where a simple choice for the gain matrices K_0 and K_1 is diagonal and positive definite matrix.

Consider again the manipulator dynamic equations. Since $D(q)$ is invertible for $q \in R^n$, we have a chance to solve for the acceleration $\ddot{q}$ of the manipulator as

$$\ddot{q} \; = \; D^{-1}u \; - \; C(q,\dot{q})\dot{q} \; - \; g(q). \tag{4.8}$$

Suppose we could specify the acceleration as the input to the robot manipulator system. Then the dynamics of the manipulator would be given as

$$\ddot{q}(t) \; = \; a_q(t) \tag{4.9}$$

where $a_q(t)$ is the input acceleration vector. The control problem for the system is now easy and the acceleration input a_q can be chosen as before according to the equation above.

Such "acceleration actuators" are not available to us and we must be content with the ability to produce a generalized force (torque) u_i at each joint i. Comparing equations above, we see that the torque u and the acceleration a_q of the manipulator are related by

$$D^{-1}u(t) \; - \; C(q, \dot{q})\dot{q} \; - \; g(q) \; = \; a_q \tag{4.10}$$

By the invertibility of the inertia matrix we may solve for the input torque $u(t)$ as

$$u \; = \; D(q)a_q \; + \; C(q, \dot{q})\dot{q} \; + \; g(q) \tag{4.11}$$

which is the same as the previously derived expression. Hence, the inverse dynamics can be viewed as an input transformation which transforms the problem from one of choosing torque input commands, which is difficult, to one of choosing acceleration input commands, which is easy. There are more details in [38]

4.1.1 Control Designs

The inverse dynamics approach relies on exact cancellation of nonlinearities in the robot equations of motion. The practical implementation of inverse dynamics control requires consideration of various sources of uncertainties such as modeling error, unknown loads, and computation error. Let us return to the Euler-Lagrange equations of motion

$$D(q)\ddot{q} \; + \; C(q, \dot{q})\dot{q} \; + \; g(q) \; = \; u \tag{4.12}$$

Controller Structure

We write the inverse dynamics control input u as

$$u \;=\; \hat{D}(q)a_q \;+\hat{C}(q,\dot{q})\dot{q}+\hat{g}(q) \tag{4.13}$$

where the notation $(\hat{\cdot})$ represents the matrices the computed value of system matrices.

Tracking Error Equation

Consider the plant given by Equation (4.13), suppose that the parameters appearing the equation are not fixed as in the robust control approach, but are time-varying estimates of the true parameters. Substituting Equation (4.13) into Equation (4.12) and express a_q as

$$a_q \;=\; \ddot{q}^d \;-\; K_1(\dot{q}-\dot{q}^d) \;-\; K_0(q \;-\; q^d) \tag{4.14}$$

Using the linear parameterization property, we can see that

$$\ddot{\tilde{q}} \;+\; K_1\dot{\tilde{q}} \;+\; K_0\tilde{q} \;=\; \hat{D}^{-1}Y(q,\dot{q},\ddot{q})\tilde{\theta} \tag{4.15}$$

where Y is the regressor function and $\tilde{\theta} = \hat{\theta} - \theta$, where $\hat{\theta}$ is the estimate of the parameter vector θ. In state space we write the system as

$$\dot{e} \;=\; Ae \;+\; B\Phi\tilde{\theta} \tag{4.16}$$

where

$$A = \begin{pmatrix} 0 & I \\ -K_0 & -K_1 \end{pmatrix}, \ B = \begin{pmatrix} 0 \\ I \end{pmatrix}, \ \Phi = \hat{D}^{-1}Y(q, \dot{q}, \ddot{q}) \tag{4.17}$$

with K_0 and K_1 chosen as before as diagonal matrices of positive gains so that A is a Hurwitz matrix. Let P be the unique symmetric, positive definite matrix P satisfying the matrix Lyapunov equation

$$A^T P + PA = -Q \tag{4.18}$$

Now, we are ready to introduce the nominal control and the adaptive control.[38]

Nominal Control System

For nominal control, we directly use the real system parameters. So, $\hat{C}(q, \dot{q})$ and $\hat{g}(q)$ in the controller can be replaced by $D(q)$, $C(q, \dot{q})$ and $\phi(q)$. Then the controller becomes

$$u = D(q)a_q(t) + C(q, \dot{q})\dot{q}(t) + g(q) \tag{4.19}$$

To verify the boundedness of all closed-loop signals and convergence of tracking error, We choose Lyapunov Function

$$V = e^T Pe \tag{4.20}$$

where matrix P is a symmetric and positive definite matrix that satisfies Lyapunov equation $A^T P + PA = -Q$. Then, with the Lyapunov equation and the tracking error equation in Equation (4.16), we can see that $\dot{V}$:

$$\begin{aligned} \dot{V} &= -e^T Q e \\ &\leqslant 0 \end{aligned} \tag{4.21}$$

where

$$e(t) = \begin{pmatrix} q(t) - q_d(t) \\ \dot{q}(t) - \dot{q}_d(t) \end{pmatrix} \tag{4.22}$$

From the above equation, we can conclude that $e(t) \in L^\infty \cap L^2$. Thus, position tracking error and velocity tracking error will converge to 0 in the end, that is, $\lim_{t \longrightarrow \infty} e(t) = 0$.

Adaptive Control System

For adaptive control, the system parameters $\theta(t)$ are unknown. W choose the parameter update law as

$$\dot{\hat{\theta}} = -\mathit{\Gamma}^{-1} \Phi^T B^T P e \tag{4.23}$$

where $\mathit{\Gamma}$ is a constant, symmetric, positive definite matrix. Then, global convergence to zero of the tracking error with all internal signals remaining bounded can be shown using the Lyapunov function

$$V = e^T P e + \frac{1}{2} \tilde{\theta}^T \mathit{\Gamma} \tilde{\theta} \tag{4.24}$$

To see this we calculate $\dot{V}$ as

$$\dot{V} = -e^T Q e + \tilde{\theta}^T \{ \Phi^T B^T P e + \mathit{\Gamma} \dot{\hat{\theta}} \} \tag{4.25}$$

the latter term following since θ is constant, that is, $\dot{\tilde{\theta}} = \dot{\hat{\theta}}$. Using the parameter update law we have

$$\begin{aligned} \dot{V} &= -e^T Q e \\ &\leqslant 0 \end{aligned} \tag{4.26}$$

From the above equation, we can conclude that $e(t) \in L^\infty \cap L^2$, $\dot{e} \in L^\infty$. Thus, position tracking error and velocity tracking error will converge to 0 in the end, that is, $\lim_{t\longrightarrow\infty} e(t) = 0$.

In order to implement this adaptive inverse dynamics scheme, it requires that the acceleration $\ddot{q}$ is need in the parameter update law and that $\hat{D}$ is invertible. The need for the joint acceleration in the parameter update law brings a serious challenge to the implementation. Acceleration sensors are noisy and introduce additional cost whereas calculating the acceleration by numerical differentiation of position or velocity signals is not feasible in most cases. The invertibility of $\hat{D}$ can be enforced in the algorithm by resetting the parameter estimate whenever $\hat{\theta}$ would otherwise result in $\hat{D}$ becoming singular. The passivity-based approaches that we treat next remove both of these impediments.[38]

4.2 Designs without Joint Acceleration Measurements

In real cases, joint acceleration measurements are available. However, the algorithm will be extremely sensitive to the accuracy of measurements. So, we can use algorithm without joint acceleration measurements alternately, that is, we don't require $\ddot{q}$ to form controller. According to [12,17,18,39], we introduce the control methods in the next parts.

To eliminate acceleration in control law, we first define new variables

$$\begin{aligned} \nu &= \dot{q}_d - \Lambda(q - q_d) \\ s &= \dot{q} - \nu \\ e &= q - q_d \end{aligned} \tag{4.27}$$

We replace acceleration terms by new variables, dynamic equations can be transfered into

$$\begin{aligned} D(q)\dot{s} + C(q,\dot{q})s &= u - D(a)\dot{\nu} - C(q,\dot{q})\nu - g(q) \\ &= u - Y(q, q_d, \dot{q}, \dot{q}_d, \ddot{q}_d)\theta^* \end{aligned} \tag{4.28}$$

Where u is applied torque for the robot manipulator, $Y(q, q_d, \dot{q}, \dot{q}_d, \ddot{q}_d)$ is called regressor which is a known matrix, and θ^* is system parameters matrix which is unknown for adaptive control parts.

4.2.1 Nominal Control System

According to [39], when θ^* is known, we directly use it to form controller. The nominal control law is

$$u(t) = Y(q, q_d, \dot{q}, \dot{q}_d, \ddot{q}_d)\theta^* - K_D s(t) \ , \ K_D = K_D^T > 0 \tag{4.29}$$

Then the dynamic equations can be write as

$$D(q)\dot{s}(t) + C(q,\dot{q})s(t) \;=\; -k_D s(t) \tag{4.30}$$

To verify that the control law can guarantee signal boundedness and tracking ability, we introduce following proofs. For signal boundedness and $\lim_{t\to\infty} e(t) = 0$, we use Lyapunov function

$$V(s) \;=\; \frac{1}{2}s^T D s \tag{4.31}$$

Then

$$\begin{aligned} \dot{V} &= s^T(t)D\dot{s}(t) + \frac{1}{2}s^T(t)\dot{D}s(t) \\ &= -s^T(t)K_D s(t) \\ &\leq 0 \end{aligned} \tag{4.32}$$

where s is defined in (4.30), D is inertia matrix, K_D is a positive definite matrix we decide.

With $\dot{V} \;=\; 0$ only when $s(t) \;=\; 0$, and $\dot{V} \;<\; 0$ in all other cases, we can conclude that

$$s(t) \;\in L^\infty, L^2$$

For e, from $s \;=\; \dot{e} + \Lambda e$, we can infer that

$$e(t) \;\in\; L^\infty, L^2,\; \dot{e} \;\in\; L^\infty$$

So, the nominal control law guarantees that all the closed-loop signals are bounded, and $\lim_{t\to\infty} e(t) = 0$.

4.2.2 Adaptive Control System

According to [17,39], in adaptive approach, the parameter matrix $\theta(t)$ remains unknown, which is a time-varying estimate of true parameter θ^*. Then, we use adaptive control law

$$u(t) \ = \ Y(q, q_d, \dot{q}, \dot{q}_d, \ddot{q}_d)\theta(t) - K_D s(t) \ , \ K_D = K_D^T > 0 \tag{4.33}$$

And for computing $\theta(t)$, the adaptive update law is

$$\dot{\theta}(t) \ = \ -\Gamma^{-1} Y(q, q_d, \dot{q}, \dot{q}_d, \ddot{q}_d)^T s(t), \ \Gamma = \Gamma^T > 0 \tag{4.34}$$

Also, we provide a proof For signal boundedness and $\lim_{t\to\infty} e(t) = 0$, we use Lyapunov function

$$V(s, \tilde{\theta}) \ = \ \frac{1}{2}(s^T D s + \tilde{\theta}^T \Gamma \tilde{\theta}) \tag{4.35}$$

Then we obtain

$$\begin{aligned} \dot{V} \ &= \ s^T D \dot{s} \frac{1}{2} s T \dot{D} s + \tilde{\theta} \gamma \dot{\theta} \\ &= \ -s^T K_D s + s^T Y \tilde{\theta} + \tilde{\theta}^T \gamma \dot{\theta} \\ &= \ -s^T K_D s \\ &\leq 0 \end{aligned} \tag{4.36}$$

Similarly, we can conclude

$$s(t) \ \in L^\infty, L^2, \ e(t) \ \in \ L^\infty, L^2, \ \dot{e} \ \in \ L^\infty$$

So, the nominal control law guarantees that all the closed-loop signals are bounded, and $\lim_{t\to\infty} e(t) = 0$.

4.3 Summary

In this chapter, we introduced the existing control methods of robot manipulators. The first design is used when we have access to the joint accelerations. We present designs of the nominal control and the adaptive control under this condition. The second design is used when we want to avoid using the joint accelerations. We also introduced the nominal and adaptive control methods for it. All designs in this chapter aim at solving control problems with time-invariant parameters. Now, we are ready to study on control designs for robot manipulator systems with time-varying parameters in the next chapters.

Chapter 5

Adaptive Control Design With Uncertain Variable Parameters

In Chapter 3 and 4, we discussed the derivation of robot manipulator dynamic equations and basic adaptive control methods. Especially in Chapter 4, we introduced cases when system parameters are time-invariant, which happen when mass, length of links, inertia and other related parameters remain unchanged. In Section 5.1, the expressions of dynamic with jumping parameters and control objective are discussed. In Section 5.2 and 5.3, the nominal control scheme and adaptive control scheme are to be discussed respectively. A comparison study of existing control scheme that introduced in Chapter 4 will be presented in Section 5.4. In the last section, we will summarize the control objective and contributions of this chapter.

5.1 Dynamic Model and Control Problem

Consider a robot manipulator system with time-varying parameters. The first objective is to give the system dynamic model with changing parameters. It is also significant to present our control problem and controller structure before the control algorithm design.

5.1.1 System with Parameter Variations

In Chapter 3, we summarized dynamic configuration of robot manipulators. In our study, the system parameters are changing during the working process. As shown in Figure 5.1, for instance, when the robot arm holds one product with it, there is a specific group of system parameters. Then the robot arm drops the product and gains another group of system parameters. So, the mass, center of mass and inertia are time-varying in the whole procedure.

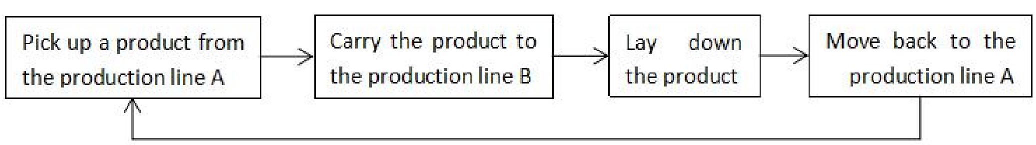

Figure 5.1: An example of working process

With the understanding of Figure 5.1, we will then discuss the dynamic equations when system parameters are time-varying. General dynamic equations of robot manipulators can be developed by the Jacobian matrix, especially for multiple links and complex relative coordinates. To show the procedures clearly, we also take the two-link planar manipulator as an example, which is one of the most common types of manipulator.

For general cases, consider a manipulator with n links, the dynamic equation is expressed as

$$D(q(t),t)\ddot{q}(t) + C(q(t),\dot{q}(t),t)\dot{q}(t) + g(q(t),t) \; = \; u(t) \tag{5.1}$$

where $q(t) \in n \times 1$ is joint variables, $D(q(t),t) \in n \times n$ is the inertia matrix, $C(q(t),\dot{q}(t),t) \in n \times n$ is Christoffel matrix, and $g(q(t),t) \in n \times 1$ is the gravity vector. All elements in these matrices are time-varying. The expression of these matrices for a specific manipulator can be derived by the Jacobian matrix introduced in Chapter 3.

For better observation, we use the two-link robot manipulator as the example, which is one of the most common robot manipulators. In Figure 5.2, it shows the configuration of the two-link planar elbow manipulator.

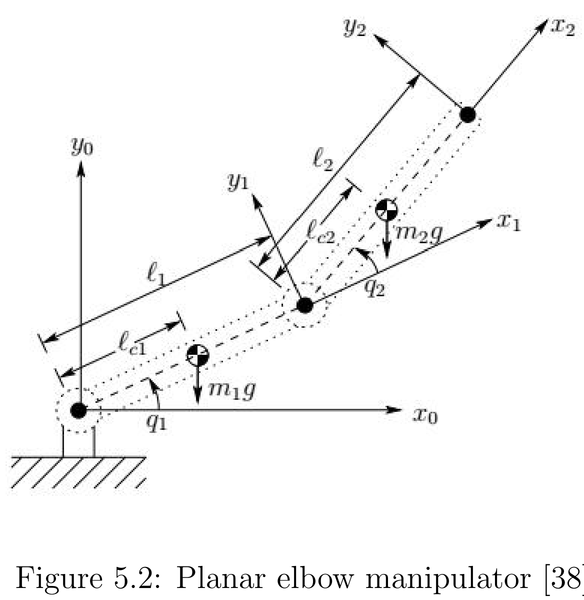

Figure 5.2: Planar elbow manipulator [38]

All system parameters include mass of two links $m_1(t)$ and $m_2(t)$, the length of two links $l_1(t)$ and $l_2(t)$, the center of mass $l_{c1}(t)$ and $l_{c2}(t)$, the inertia of two links $I_1(t)$ and $I_2(t)$, and gravity acceleration g. The new system matrices are represented by the inertia matrix $D(q(t),t)$, the Christoffel matrix $C(q(t),\dot{q}(t),t)$ and the potential matrix $g(q(t),t)$, which include time-varying system parameters $m_i(t)$, $l_i(t)$, $l_{ci}(t)$ and $I_i(t)$ $(i=1,2)$ for all links, and joint angles $q_i(t)$ and angle velocities $\dot{q}_i(t)$ $(i=1,2)$. Then, we express the matrices in Equation (5.1) as

$$D(t) = \begin{pmatrix} m_1(t)l_{c1}^2(t) + m_2(t)(l_1^2(t) + l_{c2}^2(t) + 2l_1 l_{c2}^2(t) + 2l_1(t)l_{c2}(t)\cos q_2) + I_1(t) + I_2(t) & m_2(t)(l_{c2}^2(t) + l_1 l_{c2}(t)\cos q_2) + I_2(t) \\ m_2(t)(l_{c2}^2(t) + l_1 l_{c2}(t)\cos q_2) + I_2(t) & m_2(t)l_{c2}^2(t) + I_2(t) \end{pmatrix} \tag{5.2}$$

$$C(t) = \begin{pmatrix} h(t)\dot{q}_2 & (\dot{q}_1 + \dot{q}_2)h(t) \\ -\dot{q}_1 h(t) & 0 \end{pmatrix} \tag{5.3}$$

with

$$h(t) = -m_2(t)l_1(t)l_{c2}(t)\sin q_2 \tag{5.4}$$

$$g(t) = \begin{pmatrix} (m_1(t)l_{c1}(t) + m_2(t)l_1(t))g\cos q_1 + m_2(t)l_{c2}(t)g\cos(q_1+q_2) \\ m_2(t)l_{c2}(t)g\cos(q_1+q_2) \end{pmatrix} \tag{5.5}$$

Now, all parameters in system matrices are time-varying. Specifically, we will regard them as piecewise constants, which will be discussed in the next section.

5.1.2 System Parametrization

Consider manipulators with time-varying parameters, we first organize all terms in the equations, and express them by piecewise constants. We introduce the indicator function $\chi_i(t)$, which is defined as

$$\chi_i(t) = \begin{cases} 1, & \text{if } t \in \Omega_i \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

where t is time, Ω_i is a specific working mode that we defined, and l represents the number of modes.

We consider that the information of $t \in \Omega_i$ is available, which means we know when the parameters will change. So, the indicator function $\chi_i(t)$ is known all the time.

Before applying the indicator functions to our system, we first group all unknown true system parameters into nine terms to reduce the number of unknown terms, and define them as matrix $\theta^*(t) \in 9 \times 1$:

$$\begin{aligned} \theta^*(t) &= \big(m_1(t)l_{c1}^2(t) \;\; m_2(t)l_1^2(t) \;\; m_2(t)l_{c2}^2(t) \;\; m_2(t)l_1(t)l_{c2}(t) \;\; I_1(t) \\ &\quad I_2(t) \;\; m_1(t)l_{c1}(t)g \;\; m_2(t)l_1(t)g \;\; m_2(t)l_{c2}(t)g\big)^T \\ &= (\theta_1^*(t)\; \theta_2^*(t)\; \theta_3^*(t)\; \theta_4^*(t)\; \theta_5^*(t)\; \theta_6^*(t)\; \theta_7^*(t)\; \theta_8^*(t)\; \theta_9^*(t))^T \end{aligned} \tag{5.7}$$

Next, we apply the indicator function $\chi_i(t)$ to $\theta^*(t)$ to express system parameters as piecewise constants. All terms in $\theta^*(t)$ are expressed as

$$
\begin{aligned}
\theta_1^*(t) &= \sum_{i=1}^{l} \theta_{1i}^* \chi_i(t), \quad \theta_2^*(t) = \sum_{i=1}^{l} \theta_{2i}^* \chi_i(t) \\
\theta_3^*(t) &= \sum_{i=1}^{l} \theta_{3i}^* \chi_i(t), \quad \theta_4^*(t) = \sum_{i=1}^{l} \theta_{4i}^* \chi_i(t) \\
\theta_5^*(t) &= \sum_{i=1}^{l} \theta_{5i}^* \chi_i(t), \quad \theta_6^*(t) = \sum_{i=1}^{l} \theta_{6i}^* \chi_i(t) \\
\theta_7^*(t) &= \sum_{i=1}^{l} \theta_{7i}^* \chi_i(t), \quad \theta_8^*(t) = \sum_{i=1}^{l} \theta_{8i}^* \chi_i(t) \\
\theta_9^*(t) &= \sum_{i=1}^{l} \theta_{9i}^* \chi_i(t)
\end{aligned}
\tag{5.8}
$$

where $\theta_{ki}^*\,(k = 1, 2, ...9,\ i = 1, 2, ...l)$ is the nominal parameter of $\theta_k(t)$ in mode i. Now, the true system parameters are expressed as piecewise constants. There are l working modes. Under each mode, the system have nine groups of parameters which are constants. But between different modes, the values of parameters are changing. For each mode, θ_{ki}^* are given as

$$
\begin{aligned}
\theta_{1i}^* &= m_{1i} l_{c1i}^2, \quad \theta_{2i}^* = m_{2i} l_{1i}^2 \\
\theta_{3i}^* &= m_{2i} l_{c2i}^2, \quad \theta_{4i}^* = m_{2i} l_{1i} l_{c2i} \\
\theta_{5i}^* &= I_{1i}, \quad \theta_{6i}^* = I_{2i} \\
\theta_{7i}^* &= m_{1i} l_{c1i} g, \quad \theta_{8i}^* = m_{2i} l_{1i} g \\
\theta_{9i}^* &= m_{2i} l_{c2i} g
\end{aligned}
\tag{5.9}
$$

where $m_{1i},\ m_{2i},\ l_{1i},\ l_{c1i},\ l_{c2i},\ I_{1i}$ and I_{2i} are the true values in mode i.

Equations (5.8-5.9) indicate the system parameters are piecewise constant. In each mode, parameters remain constant. But in different modes, parameters are changing. We use the indicator functions to determine which mode is the current one. According to Equation (5.6), the indicator functions are known.

Then, we substitute $\theta^*(t)$ in Equation (5.8) to the inertia matrix $D(q(t),t)$, the Christoffel matrix $C(q(t),\dot{q}(t),t)$ and the potential matrix $g(q(t),t)$ in Equation (5.2-5.5). They are expressed as

$$D(q(t),t) = \begin{pmatrix} \theta_1^*(t)+\theta_2^*(t)+\theta_3^*(t)+2\theta_4^*(t)\cos(q_2)+\theta_5^*(t)+\theta_6^*(t) & \theta_3^*(t)+\theta_4^*(t)\cos(q_2)+\theta_6^*(t) \\ \theta_3^*(t)+\theta_4^*(t)\cos(q_2)+\theta_6^*(t) & \theta_3^*(t)+\theta_6^*(t) \end{pmatrix} \tag{5.10}$$

$$C(q(t),\dot{q}(t),t) = \begin{pmatrix} -\theta_4^*(t)\sin(q_2)\dot{q}_2 & \theta_4^*(t)\sin(q_2)(\dot{q}_1+\dot{q}_2) \\ \theta_4^*(t)\sin(q_2)\dot{q}_1 & 0 \end{pmatrix} \tag{5.11}$$

$$g(q(t),t) = \begin{pmatrix} (\theta_7^*(t)+\theta_8^*(t))\cos q_1+\theta_9^*(t)\cos(q_1+q_2) \\ \theta_9^*(t)\cos(q_1+q_2) \end{pmatrix} \tag{5.12}$$

For general cases, an n-link manipulator, there exists an $n\times m$ function $W(q(t),\dot{q}(t),\ddot{q}(t))$ and an m-dimensional vector $\Phi^*(t)$ such that the left side of Equaation (5.1) can be written as

$$\begin{aligned} D(q(t),t)\ddot{q}(t)+C(q(t),\dot{q}(t),t)\dot{q}(t)+g(q(t),t) &= W(q(t),\dot{q}(t),\ddot{q}(t))\Phi^*(t) \\ &= \sum_{i=1}^{l} W(q(t),\dot{q}(t),\ddot{q}(t))\phi_i^*\chi_i(t) \end{aligned} \tag{5.13}$$

where $W(q(t), \dot{q}(t), \ddot{q}(t))$ is called the regressor, which only contains joint variables. Similar to $\theta^*(t)$ in Equations (5.8-5.9), $\Phi^*(t) = (\phi_1^*(t), ..., \phi_m^*(t))^T$ is the true parameter vector with $\phi_z^*(t) = \sum_{i=1}^{l} \phi_{zi}^* \chi_i(t)$ $(z = 1, ..., m)$ and $\phi_i^* = (\phi_{1i}^*, \phi_{21}^*, ..., \phi_{mi}^*)^T$, and ϕ_{zi}^* to be the true parameter of k_{th} parameter in mode i. Matrices $D(q(t), t)\ddot{q}(t)$, $C(q(t), \dot{q}(t), t)$ and $g(q(t), t)$ can be expressed only by $\phi_z^*(t)$ and joint variables as shown in Equations (5.10-5.12).

Now, the dynamic model of two-link manipulators is given by Equation (5.1) and (5.7-5.12), and the general one is given by Equation (5.13). Our control problem is to design the controller structure and adaptive control law for the new robot manipulator model, that ensures the desired system stability and asymptotic tracking properties. With the dynamic model in this section, we are ready for the design of controller structure.

5.2 Controller Structure and Tracking Error Equation

Our control object is designing a controller to make the joint angles and joint velocities of robot manipulators to track reference signals, and make the manipulator stable at the same time. In the whole process, true systems parameters $\theta^*(t)$ stay unknown.

5.2.1 Controller Structure

We first study on the controller structure of two-link manipulators. We define matrices $\hat{D}(q(t),t)$, $\hat{C}(q(t),\dot{q}(t),t)$ and $\hat{g}(q(t),t)$ as the estimates of $D(q(t),t)$, $C(q(t),\dot{q}(t),t)$ and $g(q(t),t)$ in Equation (5.10-5.12). They are expressed as

$$\hat{D}(q(t),t) = \begin{pmatrix} \theta_1(t)+\theta_2(t)+\theta_3(t)+2\theta_4(t)\cos(q_2)+\theta_5(t)+\theta_6(t) & \theta_3(t)+\theta_4(t)\cos(q_2)+\theta_6(t) \\ \theta_3(t)+\theta_4(t)\cos(q_2)+\theta_6(t) & \theta_3(t)+\theta_6(t) \end{pmatrix} \tag{5.14}$$

$$\hat{C}(q(t),\dot{q}(t),t) = \begin{pmatrix} -\theta_4(t)\sin(q_2)\dot{q}_2 & \theta_4(t)\sin(q_2)(\dot{q}_1+\dot{q}_2) \\ \theta_4(t)\sin(q_2)\dot{q}_1 & 0 \end{pmatrix} \tag{5.15}$$

$$\hat{g}(q(t),t) = \begin{pmatrix} (\theta_7(t)+\theta_8(t))\cos q_1+\theta_9(t)\cos(q_1+q_2) \\ \theta_9(t)\cos(q_1+q_2) \end{pmatrix} \tag{5.16}$$

where $\theta_k(t)$ $(k = 1, 2, ..., 9)$ are the estimates of true parameter $\theta_k^*(t)$ $(k = 1, 2, ..., 9)$ in Equation (5.9). We group them into one matrix $\theta(t)$:

$$\theta(t) = (\theta_1(t)\ \theta_2(t)\ \theta_3(t)\ \theta_4(t)\ \theta_5(t)\ \theta_6(t)\ \theta_7(t)\ \theta_8(t)\ \theta_9(t))^T \tag{5.17}$$

System parameters that contained in them are estimates $\theta(t)$ of nominal system parameters $\theta^*(t)$. They are also time-varying, but not piecewise constants. Similar to $\theta^*(t)$, we also apply the indicator functions to $\theta(t)$:

$$
\begin{aligned}
\theta_1(t) &= \sum_{i=1}^{l} \theta_{1i}(t)\chi_i(t), \quad \theta_2(t) = \sum_{i=1}^{l} \theta_{2i}(t)\chi_i(t) \\
\theta_3(t) &= \sum_{i=1}^{l} \theta_{3i}(t)\chi_i(t), \quad \theta_4(t) = \sum_{i=1}^{l} \theta_{4i}(t)\chi_i(t) \\
\theta_5(t) &= \sum_{i=1}^{l} \theta_{5i}(t)\chi_i(t), \quad \theta_6(t) = \sum_{i=1}^{l} \theta_{6i}(t)\chi_i(t) \\
\theta_7(t) &= \sum_{i=1}^{l} \theta_{7i}(t)\chi_i(t), \quad \theta_8(t) = \sum_{i=1}^{l} \theta_{8i}(t)\chi_i(t) \\
\theta_9(t) &= \sum_{i=1}^{l} \theta_{9i}(t)\chi_i(t)
\end{aligned}
\tag{5.18}
$$

where $\theta_{ki}(t)$ $(k = 1, 2..., 9,\ i = 1, 2, ..., l)$ are the estimates of $\theta^*_{ki}(t)$ in Equation (5.9) respectively.

Estimated value $\theta(t)$ is used in the controller structure. In Section 5.3, we will discuss the adaptive laws of updating $\theta(t)$. Now, we consider the controller structure

$$
u(t) = \hat{D}(q(t), t)a_q(t) + \hat{C}(q(t), \dot{q}(t), t)\dot{q}(t) + \hat{g}(q(t), t) \tag{5.19}
$$

where the term $a_q(t)$ is defined as

$$
a_q(t) = \ddot{q}_d(t) - K_1(\dot{q}(t) - \dot{q}_d(t)) - K_0(q(t) - q_d(t)) \tag{5.20}
$$

where K_0 and K_1 are diagonal and positive gain matrices.

For general cases, $\hat{D}(q(t), t)$, $\hat{C}(q(t), \dot{q}(t), t)$ and $\hat{g}(q(t), t)$ can expressed only by $\phi_z(t)$, the estimates of true parameters, and joint variables. The controller structure can be expressed as

$$\tau(t) \ = \ \hat{D}(\Phi(t), q(t))a_q(t) \ + \ \hat{C}(\Phi(t), q(t), \dot{q}(t))\dot{q}(t) \ + \ \hat{g}(\Phi(t), q(t)) \tag{5.21}$$

where $\phi_z(t) = \sum_{i=1}^{l} \phi_{zi}(t)\chi_i(t)$ $(z = 1, ..., m)$, with $\phi_{zi}(t)$ to be the estimates of the true parameter ϕ^*_{zi} in mode i, m denotes the number of groups of parameters, l denotes the number of modes.

5.2.2 Tracking Error Equation

We define $\tilde{q}(t) = q(t) - q_d(t)$, $\dot{\tilde{q}}(t) = \dot{q}(t) - \dot{q}_d(t)$, where $q_d(t)$ and $\dot{q}_d(t)$ are the desired joint angle and joint velocities. We can get the desired values from a set trajectory. We also define $\tilde{D}(t) = \hat{D}(q(t), t) - D(q(t), t)$, $\tilde{C}(t) = \hat{C}(q(t), \dot{q}(t), t) - C(q(t), \dot{q}(t), t)$ and $\tilde{g}(t) = \hat{g}(q(t), t) - g(q(t), t)$. With Equation (5.1), (5.19) and the knowledge of $D(t)$ is invertible, we can compute that

$$\begin{aligned}
\ddot{q}(t) \ &= D^{-1}(t)(u(t) - C(t)\dot{q}(t) - g(t)) \\
\ddot{q}(t) \ &= D^{-1}(t)(\hat{D}(t)a_q(t) + \hat{C}(t)\dot{q}(t) + \hat{g}(t) - C(t)\dot{q}(t) - g(t)) \\
\ddot{q}(t) \ &= D^{-1}(t)\hat{D}(t)(-\ddot{\tilde{q}}(t) - K_1\dot{\tilde{q}}(t) - K_0\tilde{q}(t)) + D^{-1}\hat{D}(t)\ddot{q}(t) \\
&\quad + D^{-1}(t)\hat{D}(t)\ddot{q}(t) - D^{-1}\tilde{C}(t)\dot{q} - D^{-1}\tilde{g}(t) \\
\ddot{\tilde{q}}(t) + K_1\dot{\tilde{q}}(t) + K_0\tilde{q}(t) \ &= \hat{D}^{-1}(t)D(t)(D^{-1}(t)\hat{D}(t) - I)\ddot{q}(t) - \hat{D}^{-1}(t)\tilde{C}(t)\dot{q}(t) \\
&\quad - \hat{D}^{-1}\tilde{g}(t) \\
\ddot{\tilde{q}}(t) + K_1\dot{\tilde{q}}(t) + K_0\tilde{q}(t) \ &= \hat{D}^{-1}(\tilde{D}(t)\ddot{q}(t) + \tilde{C}(t)\dot{q}(t) + \tilde{g}(t))
\end{aligned} \tag{5.22}$$

With the linear parameterization property that the system parameters appear as coefficients of known function of joint variables, we can get a linear relationship result

$$\tilde{D}(t)\ddot{q}(t) + \tilde{C}(t)\dot{q}(t) + \tilde{g}(t) \;=\; \sum_{i=1}^{l} Y(q,\dot{q},\ddot{q})\tilde{\theta}_i(t)\chi_i(t) \tag{5.23}$$

where $\tilde{\theta}_i(t) = \theta_i(t) - \theta_i^*$, with $\theta_i^* = (\theta_{1i}^*, \theta_{2i}^*, ..., \theta_{9i}^*)^T$ being the constant system parameter vector for mode i, and $\theta_i(t) = (\theta_{1i}(t), \theta_{2i}(t), ..., \theta_{9i}(t))^T$ being its estimate, and regressor $Y(q,\dot{q},\ddot{q})$ can be got by expanding all terms in the left of Equation (5.23), and grouping all joint variables into one matrix. $Y(q,\dot{q},\ddot{q})$ is given as

$$Y(q,\dot{q},\ddot{q}) \;=\; \begin{pmatrix} \ddot{q}_1 & \ddot{q}_1 & \ddot{q}_1+\ddot{q}_2 & \cos(q_2)(2\ddot{q}_1+\ddot{q}_2) - \sin(q_2)(\dot{q}_1^2 + 2\dot{q}_1\dot{q}_2) & \ddot{q}_1 & \ddot{q}_1+\ddot{q}_2 & \cos(q_1) & \cos(q_1) & \cos(q_1+q_2) \\ 0 & 0 & \ddot{q}_1+\ddot{q}_2 & \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 & 0 & \ddot{q}_1+\ddot{q}_2 & 0 & 0 & \cos(q_1+q_2) \end{pmatrix} \tag{5.24}$$

Then, we substitute Equation (5.21) into (5.20), we can see that

$$\ddot{\tilde{q}}(t) \;+\; K_1\dot{\tilde{q}}(t) \;+\; K_0\tilde{q}(t) \;=\; \sum_{i=1}^{l} \hat{D}_i^{-1}(t)Y(q,\dot{q},\ddot{q})\tilde{\theta}_i(t)\chi_i(t) \tag{5.25}$$

where $\hat{D}_i^{-1}(t)$ uses the estimate $\theta_i(t)$ for mode i, which is defined as

$$\hat{D}_i^{-1}(t) \;=\; \begin{pmatrix} \theta_{1i}(t)+\theta_{2i}(t)+\theta_{3i}(t)+2\theta_{4i}(t)\cos(q_2(t))+\theta_{5i}(t)+\theta_{6i}(t) & \theta_{3i}(t)+\theta_{4i}(t)\cos(q_2(t))+\theta_{6i}(t) \\ \theta_{3i}(t)+\theta_{4i}(t)\cos(q_2(t))+\theta_{6i}(t) & \theta_{3i}(t)+\theta_{6i}(t) \end{pmatrix} \tag{5.26}$$

Next, we define the position tracking error and velocity tracking error in the vector $e(t)$:

$$e(t) \;=\; \begin{pmatrix} q(t) - q_d(t) \\ \dot{q}(t) - \dot{q}_d(t) \end{pmatrix} \;=\; \begin{pmatrix} \tilde{q}(t) \\ \dot{\tilde{q}}(t) \end{pmatrix} \tag{5.27}$$

Now, we get the tracking error equation $\dot{e}(t)$ from Equation (5.25) and (5.27) as

$$\dot{e}(t) \;=\; Ae(t) \;+\; B\sum_{i=1}^{l} \hat{D}_i^{-1}(t)Y(q,\dot{q},\ddot{q})\tilde{\theta}_i(t)\chi_i(t) \tag{5.28}$$

where

$$A \;=\; \begin{pmatrix} 0 & I \\ -K_0 & -K_1 \end{pmatrix},\; B \;=\; \begin{pmatrix} 0 \\ I \end{pmatrix} \tag{5.29}$$

with K_0 and K_1 chosen as before as diagonal and positive gain matrices so that A is a Hurwitz matrix. Let P be the unique symmetric, positive definite matrix P satisfying the matrix Lyapunov equation

$$A^T P \;+\; PA \;=\; -Q \tag{5.30}$$

where Q is constant and $Q = Q^T > 0$.

Similar to the two-link manipulator, we now consider am n-link manipulator. As defined in Equation (5.13), we group all system parameters into m groups, and use z to denote the z_{th} group of parameters. The system parameter matrix is $\phi_i^*(t) = (\phi_{1i}^*, \phi_{21}^*, ..., \phi_{mi}^*)^T$, and $W(q,\dot{q},\ddot{q})$ is the regressor that only contains joint variables. The tracking error equation of general cases is given as

$$\dot{e}(t) \;=\; Ae(t) \;+\; B\sum_{i=1}^{l} \hat{D}_i^{-1}(\phi_i(t), q(t))W(q,\dot{q},\ddot{q})\tilde{\phi}_i(t)\chi_i(t) \tag{5.31}$$

where $\hat{D}_i(\phi_i(t), q(t))$ is the estimate of $D(q(t), t)$ in mode i, $\tilde{\phi}_i(t) = \phi_i(t) - \phi_i^*$ is the parameter estimate error.

Now, the controller structure is ready for further design of nominal control and adaptive control. We will provide control design, stability analysis and simulation study of both of them in the next sections.

5.3 Nominal Control System

For nominal control, all system parameters are know, so θ_{ki}^*, $(k = 1, 2, ..., 9,\ l = 1, 2, ..., l)$ in Equation (5.9) are known. We use $\theta^*(t)$ to substitute $\theta(t)$ in the matrices of the controller structure in Equation (5.15-5.16). The controller structure becomes

$$u(t) \ = \ D(q(t), t)a_q(t) \ + \ C(q(t), \dot{q}(t), t)\dot{q}(t) \ + g(q(t), t) \tag{5.32}$$

Now, $\hat{D}(q(t), t)$, $\hat{C}(q(t), \dot{q}(t), t)$ and $\hat{g}(q(t), t)$ in the controller actually become $D(q(t), t)$, $C(q(t), \dot{q}(t), t)$ and $g(q(t), t)$ in Equation (5.10-5.12) which contain true value of all system parameters.

5.3.1 Stability Analysis

To verify the boundedness of all closed-loop signals and convergence of tracking error, we use $\dot{e}(t)$ in (5.28), which becomes

$$\dot{e}(t) \ = \ Ae(t) \tag{5.33}$$

Since

$$A = \begin{pmatrix} 0 & I \\ -K_0 & -K_1 \end{pmatrix} \in R^{4\times 4}$$

where K_0 and K_1 are diagonal and positive. We can get position tracking error $e_1(t) = \tilde{q}(t)$ and velocity tracking error $e_2(t) = \dot{\tilde{q}}(t)$ satisfy

$$\begin{aligned} \dot{e}_1(t) &= e_2(t) \\ \dot{e}_1(t) &= -K_0 e_1(t) - K_1 e_2(t) \end{aligned} \tag{5.34}$$

which implies that $\lim_{t\longrightarrow\infty} e(t) = 0$, with positive definite K_0 and K_1 and stable A.

5.3.2 Simulation Study

In this section, we will study simulations of two typical working conditions of the planar elbow robot manipulator. In both cases, parameters are known at this time.

Cases with Repetitively Jumping Parameters

In this case, we set two working modes for the two-link planar manipulator, and make the modes switch back and forth every 10 seconds. The working process is the manipulators repeat picking and dropping the same kind of commodities. The system parameters are given below. In mode 1, the manipulator is in normal conditions. In mode 2, the manipulator holds a commodity by link 2.

For simulation, the whole process is 50 seconds long. The initial values of joint

Table 5.1: Parameters of Case I (nominal control)

Parameters (unit)	Mode 1	Mode 2
Mass of link 1: m_1 (kg)	1	1
Mass of link 2: m_2 (kg)	2.8	3
Length of link 1: l_1 (m)	1	1
Center of mass of link 1: l_{c1} (m)	1/2	1/2
Center of mass of link 2: l_{c2} (m)	0.9	1
Inertia of link 1: I_1 $(kg \cdot m^2)$	1/12	1/12
Inertia of link 2: I_2 $(kg \cdot m^2)$	0.37	2/5
Gravity acceleration: g (m/s^2)	9.8	9.8
K_0	diag{100, 100}	diag {1000, 1000}
K_1	diag{100, 100}	diag{1000, 1000}

variables are $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -\pi, -\pi)$. The desired joint variables are $q_{d1}(t) = q_{d2}(t) = \sin(\pi t)$.

Simulation results

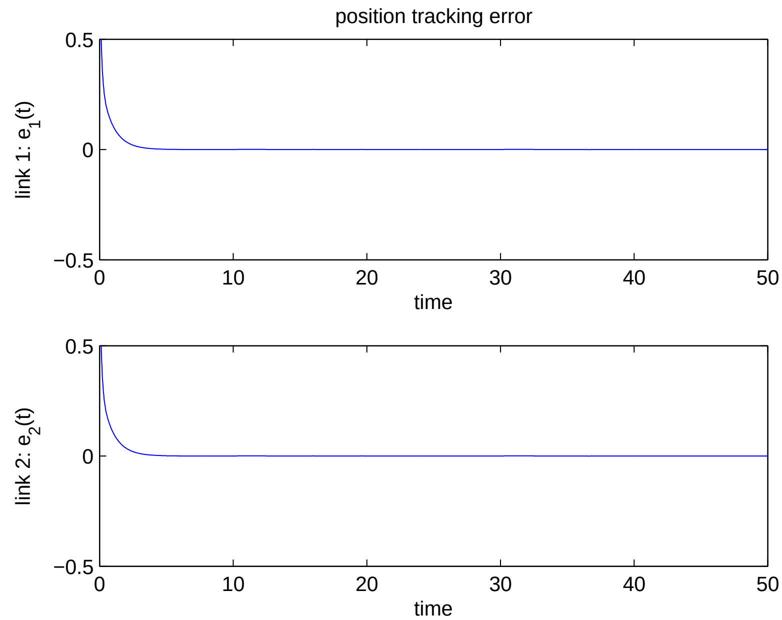

Figure 5.3: Position tracking error (Case I: nominal control)

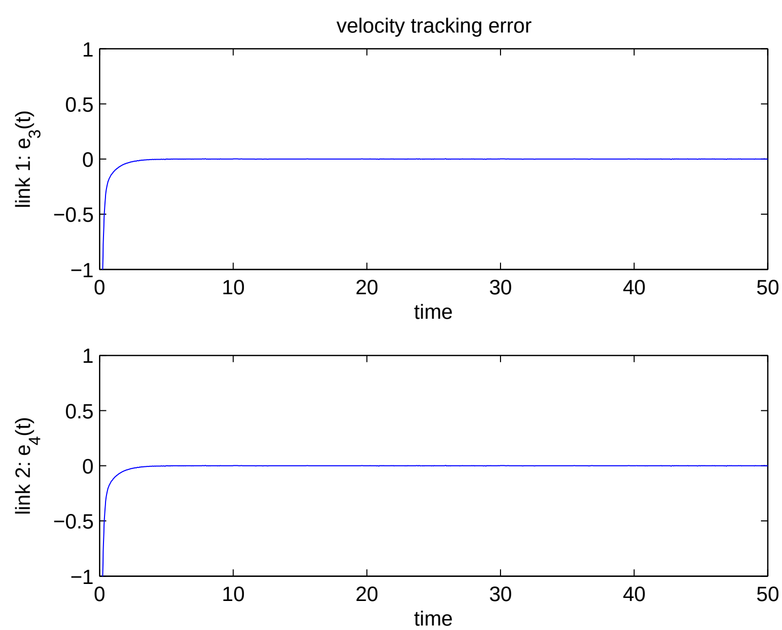

Figure 5.4: Velocity tracking error (Case I: nominal control)

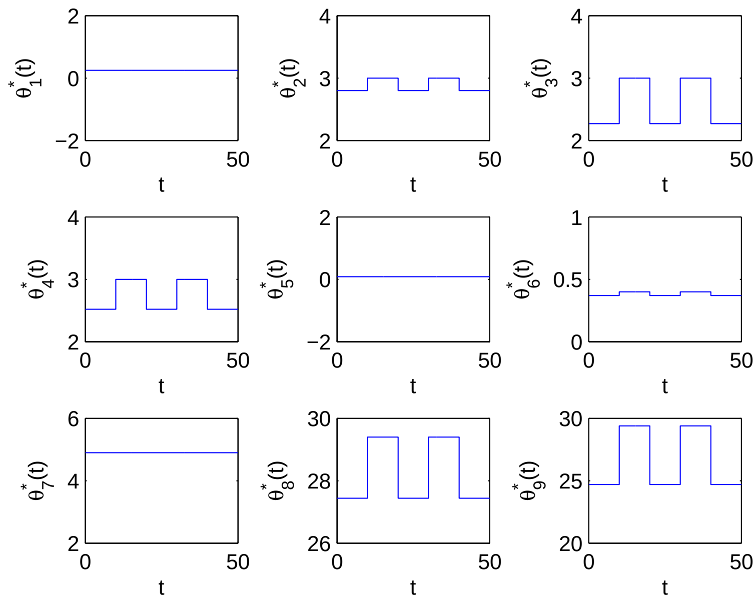

Figure 5.5: System parameters $\theta^*(t)$ (Case I: nominal control)

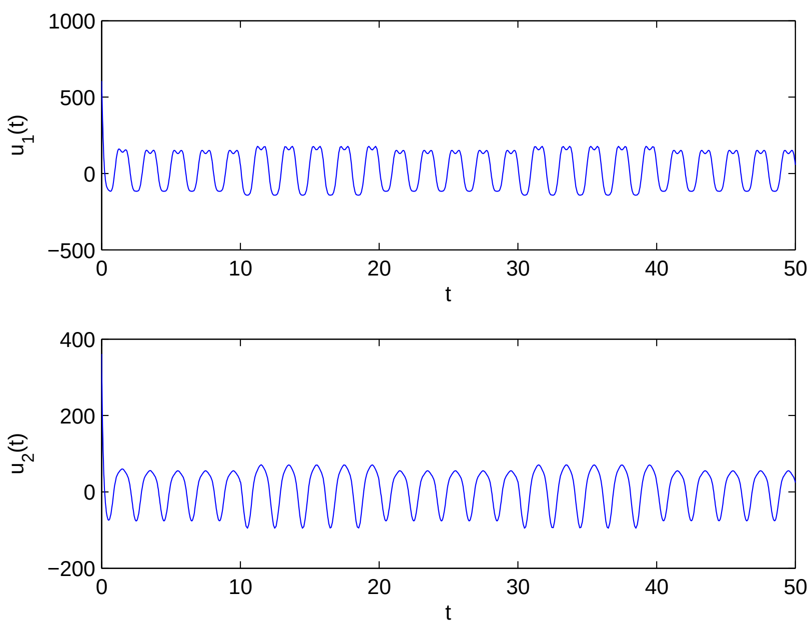

Figure 5.6: Control inputs (Case I: nominal control)

Figure 5.3 and 5.4 show the errors of joint angles and joint velocities. We can see that the plot converge to zero quickly, and remain stable to the end of simulation. In figure 5.5, we present the variations of $\theta_k^*(t)(k = 1, 2, ..., 9)$. All $\theta_k^*(t)$ that contain changing parameters in Table 5.1 have similar variation plots.

As shown in the Figures 5.2-5.5, the system is stable, and the tracking object is realized.

Cases with Time-Varying Parameters

In this case, we will simulate a whole working process with four procedures. For four modes, we switch one mode to another every 20 seconds. The whole simulation is 80 seconds long.

For simulation, the whole process is 80 seconds long. The initial values of joint variables are $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -\pi, -\pi)$. The desired joint variables are $q_{d1}(t) = q_{d2}(t) = \sin(\pi t)$.

Simulation results

Table 5.2: Parameters of Case II (nominal control)

Parameters (unit)	Mode 1	Mode 2	Mode 3	Mode 4
Mass of link 1: m_1 (kg)	1	1	1	1
Mass of link 2: m_2 (kg)	3	6	2.8	3
Length of link 1: l_1 (m)	1	1	1	1
Center of mass of link 1: l_{c1} (m)	1/2	1/2	1/2	1/2
Center of mass of link 2: l_{c2} (m)	1	1	0.9	1
Inertia of link 1: I_1 $(kg \cdot m^2)$	1/12	1/12	1/12	1/12
Inertia of link 2: I_2 $(kg \cdot m^2)$	2/5	4/5	0.37	2/5
Gravity acceleration: g (m/s^2)	9.8	9.8	9.8	9.8
K_0	diag{100, 100}	diag {1000, 1000}	diag{100, 100}	diag {1000, 1000}
K_1	diag{100, 100}	diag {1000, 1000}	diag{100, 100}	diag {1000, 1000}

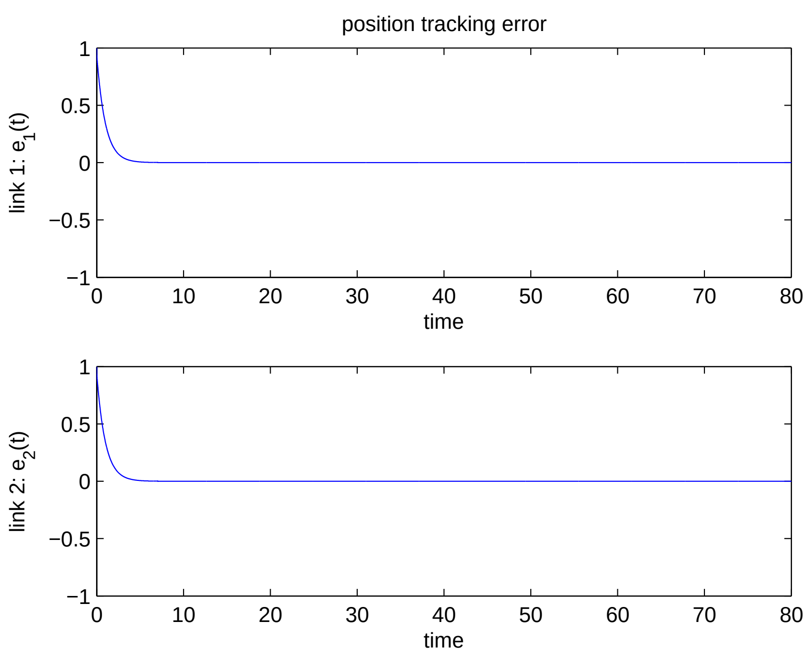

Figure 5.7: Position tracking error (Case II: nominal control)

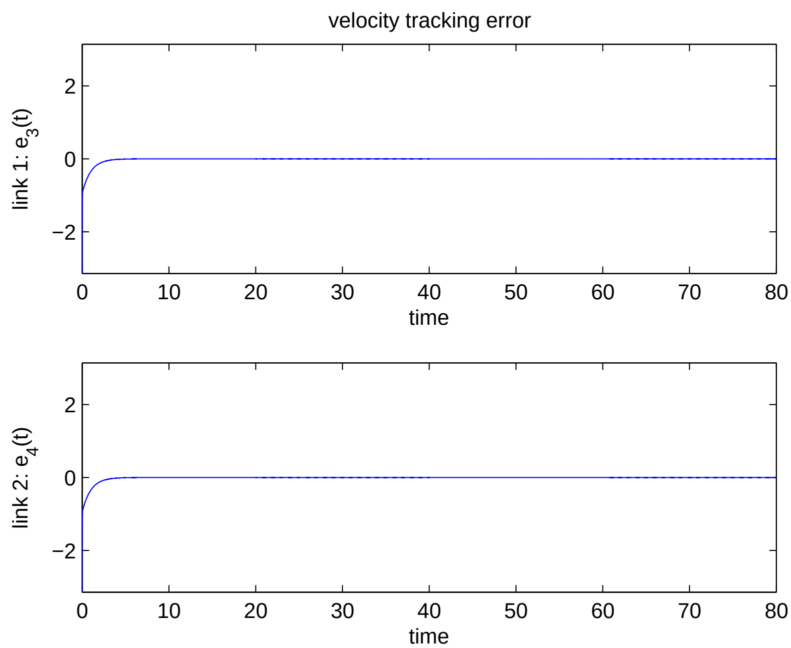

Figure 5.8: Velocity tracking error (Case II: nominal control)

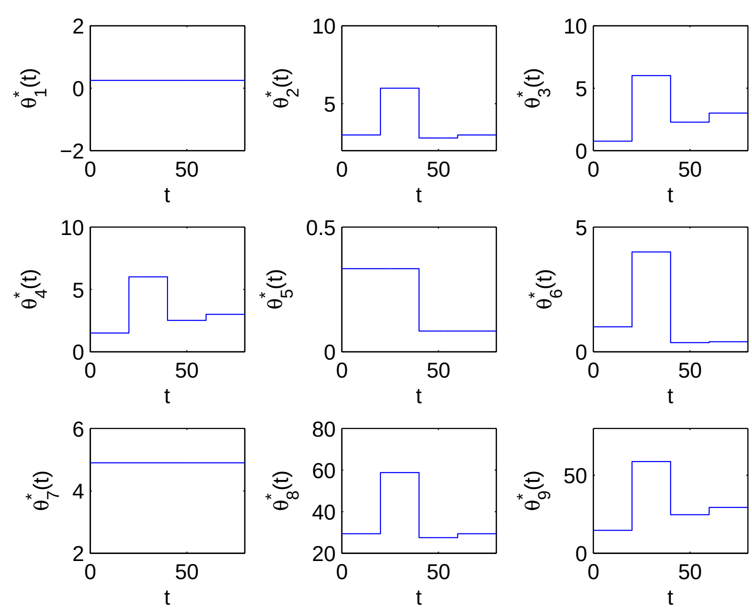

Figure 5.9: System parameters $\theta^*(t)$ (Case II: nominal control)

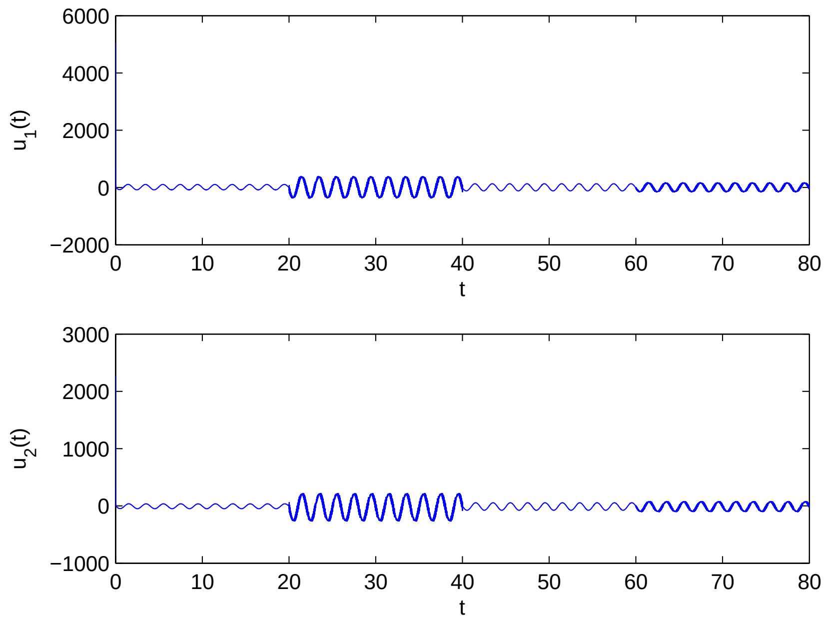

Figure 5.10: Control inputs (Case II: nominal control)

Figure 5.6 and 5.7 show the errors of joint angles and joint velocities. We can see that the plot converge to zero quickly, and remain stable to the end of the simulation. In figure 5.8, we present the variations of $\theta_k^*(t)(k = 1, 2, ..., 9)$. All θ_k^* that contain changing parameters in Table 5.1 have similar variation plots.

As shown in the Figures 5.6-5.8, the system is stable, and the tracking object is realized.

5.4 Adaptive Control System

For adaptive control, system parameters remain unknown. We first study on the control of two-link manipulators. We use estimated values $\theta(t)$ in Equation (5.17-5.18) in the controller structure. Also with $\hat{D}(q(t),t)$, $\hat{C}(q(t),\dot{q}(t),t)$, $\hat{g}(q(t),t)$ in Equation (5.14-5.16) and $a_q(t)$ in Equation (5.20), the controller structure is given in Equation (5.19) that

$$u(t) \;=\; \hat{D}(q(t),t)a_q(t) \;+\; \hat{C}(q(t),\dot{q}(t),t)\dot{q}(t) \;+\; \hat{g}(q(t),t)$$

Now, we design the adaptive law for updating $\theta(t)$ as

$$\begin{aligned} \dot{\theta}_{ki}(t) \;&=\; -\,\Gamma_{ki}((\hat{D}_i^{-1}(t)Y(t))^T B^T Pe(t))_k \chi_i(t) \\ &(k = 1,2,...,9,\; i = 1,2,...,l) \end{aligned} \tag{5.35}$$

where the notation $(\cdot)_k$ means the k th row of the matrix $(\cdot)$, $\chi_i(t)$ is the indicator function that is defined in Equation (5.1), i indicates the working mode, $\hat{D}_i^{-1}(t)$ is defined in Equation (5.26), Y is the regressor in Equation (5.24), A and B are defined in Equation (5.29), P comes from Lyapunov equation in Equation (5.30). Γ_{ki} is a positive constants for the k_{th} parameter in mode i.

Similarly, for general cases, we can express the update law for parameter estimates $\phi_{zi}(t)(z = 1,2,...,m)$ as

$$\begin{aligned} \dot{\phi}_{zi}(t) \;&=\; -\,\Gamma_{zi}((\hat{D}_i^{-1}(\phi_{zi}(t),q(t))W(t))^T B^T Pe(t))_z \chi_i(t) \\ &(z = 1,2,...,m,\; i = 1,2,...,l) \end{aligned} \tag{5.36}$$

where i indicates the working mode, $\hat{D}_i^{-1}(t)$ is the estimate of $D(q(t),t)^{-1}$ in mode

i, W is the regressor generated from Equation (5.13), A and B are defined in Equation (5.29), P comes from Lyapunov equation in Equation (5.30). Γ_{zi} is a positive constants for the z$_{th}$ parameter in mode i.

5.4.1 Stability Analysis

Before applying our algorithm, it is significant to examine stability. We first review the tracking error equation in Equation (5.31):

$$\dot{e}(t) \;=\; Ae(t) \;+\; B\sum_{i=1}^{l}\hat{D}_i^{-1}(\phi_i(t),q(t))W(q,\dot{q},\ddot{q})\tilde{\phi}_i(t)\chi_i(t)$$

where $e(t)$ is tracking error, A and B are defined in Equation (5.29), $\hat{D}_i^{-1}(t)$ is the estimate of $D(q(t),t)^{-1}$ in mode i, $W(q,\dot{q},\ddot{q})$ is the regressor generated from Equation (5.13). Specifically, for two-link manipulator, $\hat{D}_i^{-1}$ is given in Equation (5.26), $W(q,\dot{q},\ddot{q}) = Y(q,\dot{q},\ddot{q})$ is given in Equation (5.24), and $\tilde{\phi}(t) = \tilde{\theta}(t)$ is the parameter error.

With the adaptive update law in Equation (5.36), we choose Lyapunov function as

$$V \;=\; e^TPe \;+\; \sum_{i=1}^{l}(\Gamma_{1i}^{-1}\tilde{\phi}_{1i}^2+\Gamma_{2i}^{-1}\tilde{\phi}_{2i}^2+...+\Gamma_{mi}^{-1}\tilde{\phi}_{mi}^2) \tag{5.37}$$

with the Lyapunov matrix $P = P^T > 0$ satisfying $A^TP + PA = -Q$ for some $Q = Q^T > 0$, the tracking error equation in Equation (5.31), and the adaptive law in (5.31), we get $\dot{V}$ as

$$
\begin{aligned}
\dot{V} &= 2e^T P\dot{e} + 2\sum_{i=1}^{l}(\Gamma_{1i}^{-1}\tilde{\phi}_{1i}\dot{\tilde{\phi}}_{1i} + \Gamma_{2i}^{-1}\tilde{\phi}_{2i}\dot{\tilde{\phi}}_{2i} + ... + \Gamma_{mi}^{-1}\tilde{\phi}_{mi}\dot{\tilde{\phi}}_{mi}) \\
&= 2e^T P(Ae + \sum_{i=1}^{l}((B\hat{D}_i^{-1}Y)_1\tilde{\phi}_{1i}\chi_i(t) + (B\hat{D}_i^{-1}Y)_2\tilde{\phi}_{2i}\chi_i(t) + ... \\
&\quad + (B\hat{D}_i^{-1}Y)_m\tilde{\phi}_{mi}\chi_i(t))) + 2\sum_{i=1}^{l}(\Gamma_{1i}^{-1}\tilde{\phi}_{1i}\dot{\tilde{\phi}}_{1i} + \Gamma_{2i}^{-1}\tilde{\phi}_{2i}\dot{\tilde{\phi}}_{2i} + ... + \Gamma_{mi}^{-1}\tilde{\phi}_{mi}\dot{\tilde{\phi}}_{mi}) \\
&= -e^T Qe + 2\sum_{i=1}^{l}(((e^T PB\hat{D}_i^{-1}Y)_1\chi_i(t) + \Gamma_{1i}^{-1}\dot{\phi}_{1i})\tilde{\phi}_{1i} + ... \\
&\quad + ((e^T PB\hat{D}_i^{-1}Y)_m\chi_i(t) + \Gamma_{mi}^{-1}\dot{\phi}_{mi})\tilde{\phi}_{mi}) \\
&= -e^T Qe \\
&\leqslant 0
\end{aligned}
\tag{5.38}
$$

From the above equation and the tracking error equation in Equation (5.25), $\dot{V}$ equals to zero only when e is zero. We can conclude that $e(t) \in L^{\infty} \cap L^2$, $\dot{e}(t) \in L^{\infty}$. Then, with the Barbălat lemma, the position tracking error and velocity tracking error will converge to zero, that is, $\lim_{t\longrightarrow\infty} e(t) = 0$.

5.4.2 Simulation Study

Similar to the nominal control parts, we will study on the simulation of planar elbow robot manipulators in two cases. At this time, true values of system parameters remain unknown, and parameters we set are only for the simulation needs. They will not be used in our controller structure and the adaptive law.

Cases with Uncertain Repetitively Jumping Parameters

For better observation, we set two working modes, and make the modes switch back and forth every 10 seconds. The system parameters are given below. In mode 1, the manipulator is in normal conditions. In mode 2, the robot manipulator picks some commodities by link 2.

For simulation, the whole process is 100 seconds long. The initial values of

Table 5.3: Parameters of Case I (adaptive control)

Parameters (unit)	Mode 1	Mode 2
Mass of link 1: m_1 (kg)	1	1
Mass of link 2: m_2 (kg)	6	3
Length of link 1: l_1 (m)	1	1
Center of mass of link 1: l_{c1} (m)	1/2	1/2
Center of mass of link 2: l_{c2} (m)	1	1
Inertia of link 1: I_1 $(kg \cdot m^2)$	1/12	1/12
Inertia of link 2: I_2 $(kg \cdot m^2)$	4/5	2/5
Gravity acceleration: g (m/s^2)	9.8	9.8
K_0	diag{1000, 1000}	diag {100, 100}
K_1	diag{1000, 1000}	diag{100, 100}

joint variables are $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -\pi, -\pi)$. The desired joint variables are $q_{d1}(t) = q_{d2}(t) = \sin(\pi t)$. All $\Gamma_i = \text{diag}\{0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001\}$, $Q = \text{diag}\{0.01, 0.01, 0.01, 0.01\}$. For better observation, we set the initial parameter estimations to be 80% of the nominal values, that is, $\theta_{ki}(t)|_{t=0} = 80\%\theta^*_{ki}$.

Simulation results

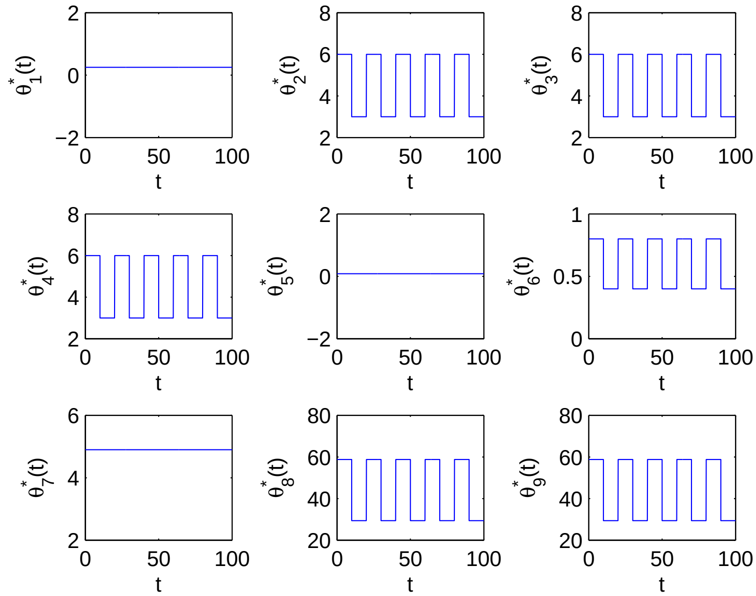

Figure 5.11: System parameters $\theta^*(t)$ (Case I: adaptive control)

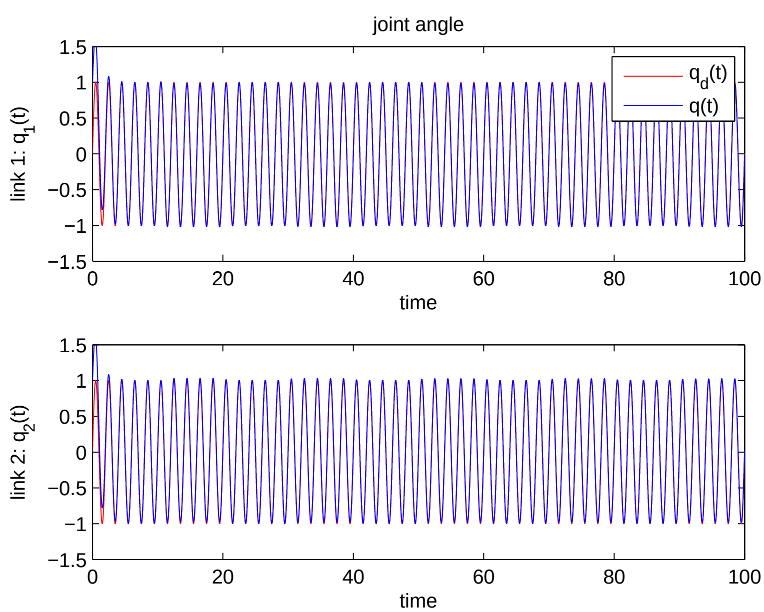

Figure 5.12: Joint angles contrast with desired joint angles (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

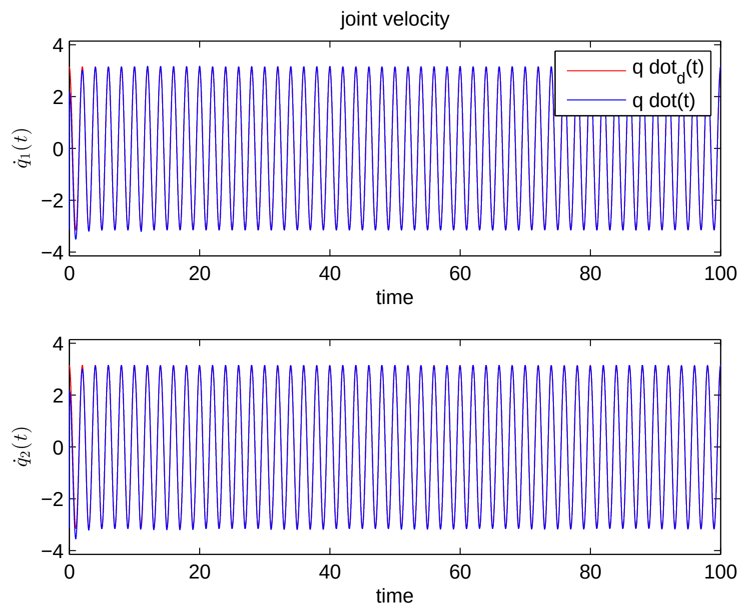

Figure 5.13: Joint velocities contrast with desired joint velocities (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

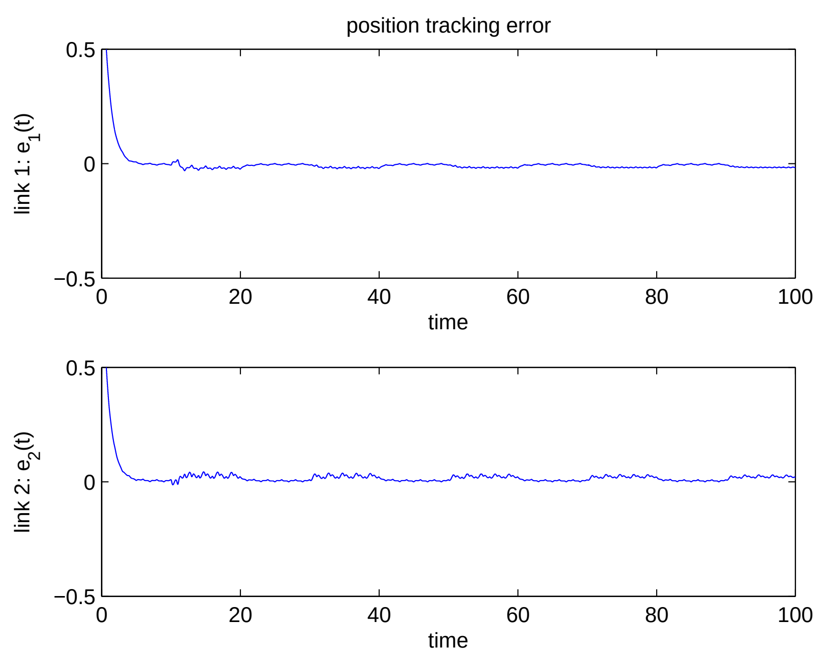

Figure 5.14: Position tracking error (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

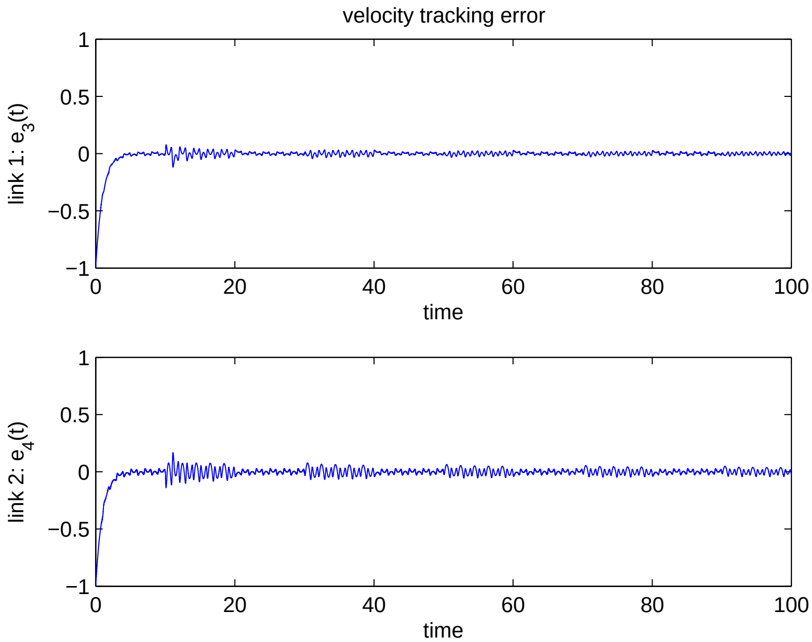

Figure 5.15: Velocity tracking error (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

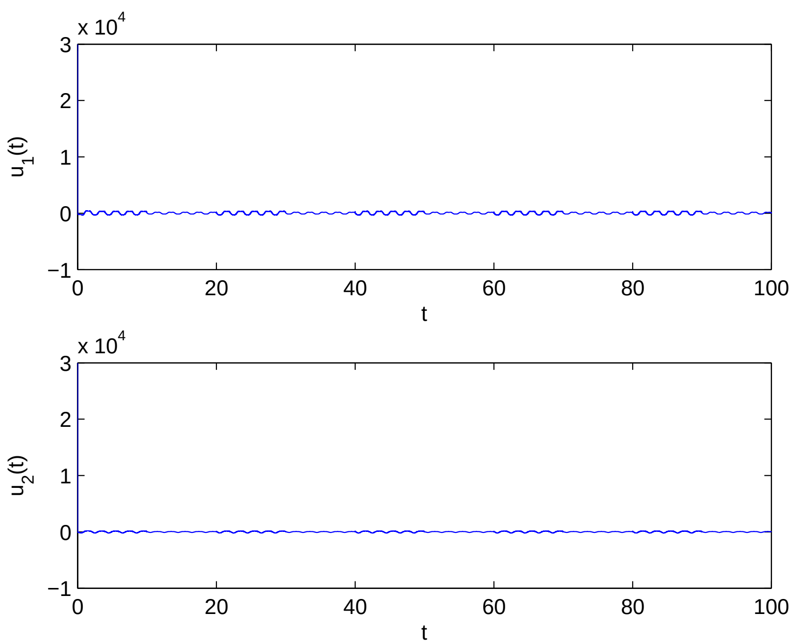

Figure 5.16: Control input u(t) (Case I: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

In Figure 5.9 and 5.10, the plots show that the real trajectory can track the desired trajectory. As shown in the Figure 5.11 and 5.12, all errors converge to zero. The real joint angles and velocities can track the desired ones asymptotically. Especially, we can see that disturbance is getting smaller even though the modes keep changing. We can conclude that our control algorithms realized the control objective.

Cases with Uncertain Time-Varying Parameters

We also have this four-mode simulation for a whole working procedure. We switch one mode to another every 10 seconds. The whole simulation is 40 seconds.

For simulation, the whole process is 40 seconds long. The initial values of

Table 5.4: Parameters of Case II (adaptive control)

Parameters (unit)	Mode 1	Mode 2	Mode 3	Mode 4
Mass of link 1: m_1 (kg)	1	1	1	1
Mass of link 2: m_2 (kg)	3	2.8	6	3
Length of link 1: l_1 (m)	1	1	1	1
Center of mass of link 1: l_{c1} (m)	1/2	1/2	1/2	1/2
Center of mass of link 2: l_{c2} (m)	1	0.9	1	1
Inertia of link 1: I_1 $(kg \cdot m^2)$	1/12	1/12	1/12	1/12
Inertia of link 2: I_2 $(kg \cdot m^2)$	2/5	0.37	4/5	2/5
Gravity acceleration: g (m/s^2)	9.8	9.8	9.8	9.8
K_0	diag{100, 100}	diag {1000, 1000}	diag{100, 100}	diag {1000, 1000}
K_1	diag{100, 100}	diag {1000, 1000}	diag{100, 100}	diag {1000, 1000}

joint variables are $(q_1, q_2, \dot{q}_1, \dot{q}_2)|_{t=0} = (1, 1, -\pi, -\pi)$. The desired joint variables are

$q_{d1}(t) = q_{d2}(t) = \sin(\pi t)$. All $\Gamma = \mathrm{diag}\{0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001\}$, $Q = \mathrm{diag}\{0.01, 0.01, 0.01, 0.01\}$

Simulation results

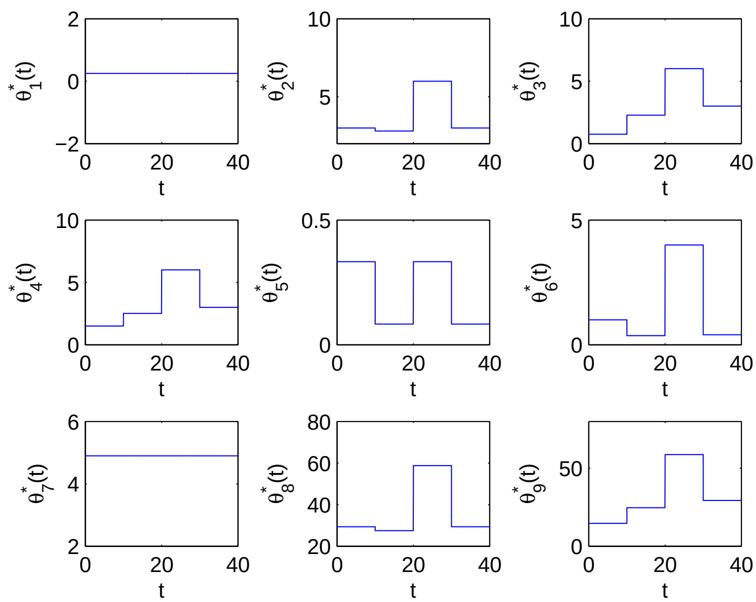

Figure 5.17: System parameters $\theta^*(t)$ (Case II: adaptive control)

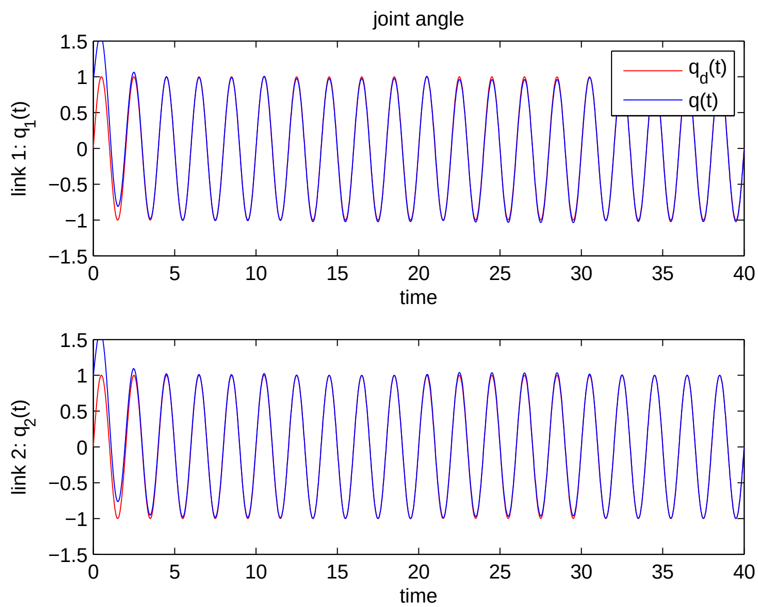

Figure 5.18: Joint angles contrast with desired joint angles (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

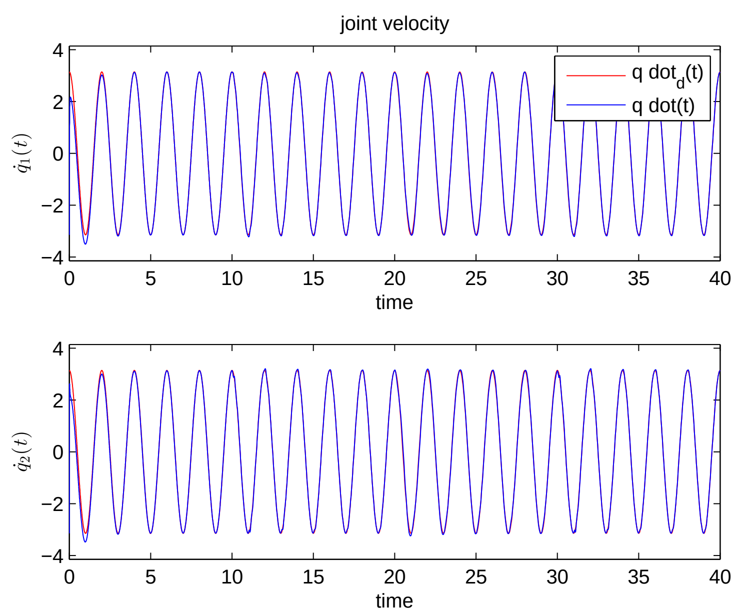

Figure 5.19: Joint velocities contrast with desired joint velocities (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

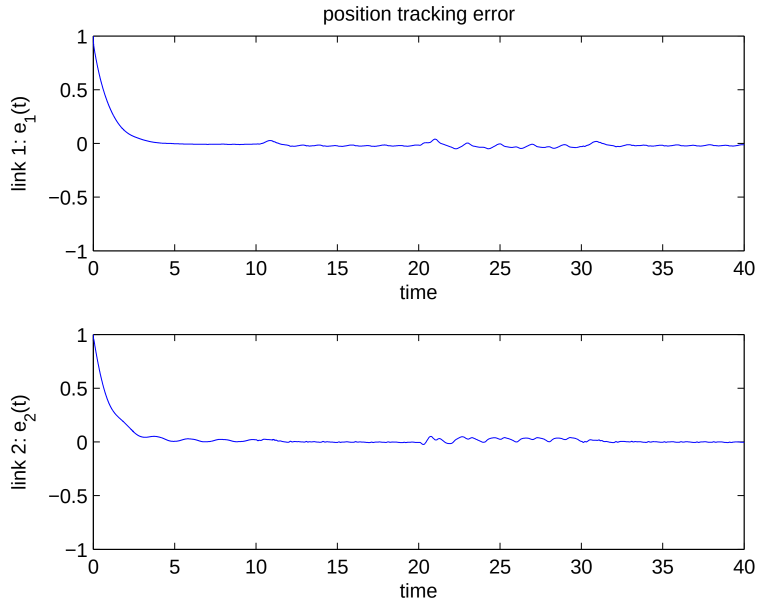

Figure 5.20: Position tracking error of adaptive control (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

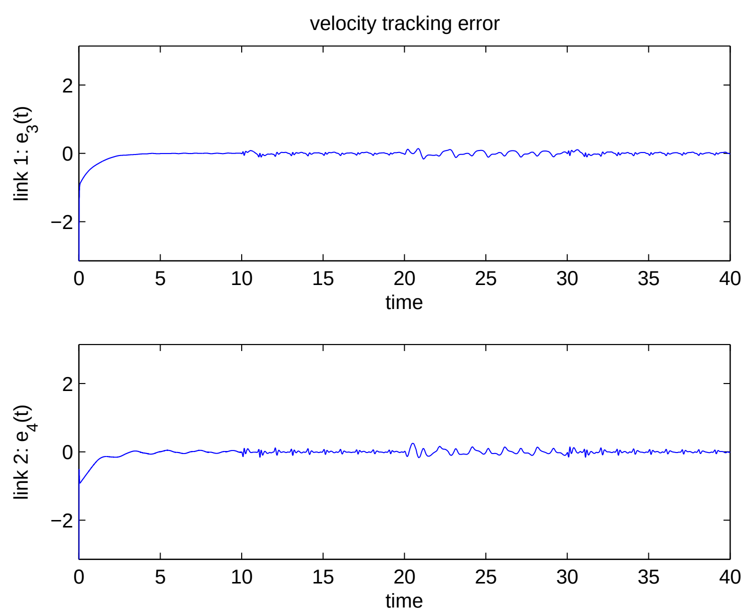

Figure 5.21: Velocity tracking error of adaptive control (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

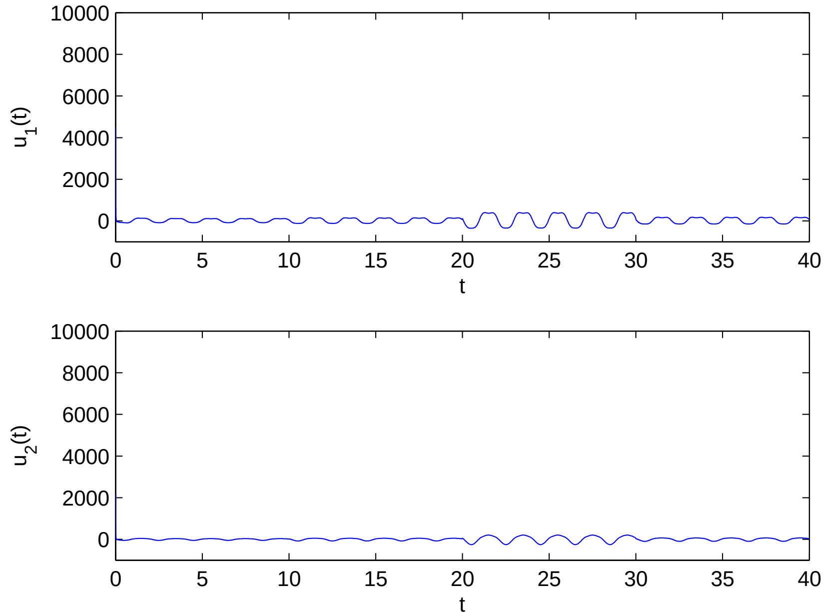

Figure 5.22: Control input u(t) (Case II: adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

As shown in Figure 5.13-5.16, the real joint variables can track the desired trajectory. Position tracking error and Velocity tracking error converge to zero in each modes. So, we can conclude that our control design realize the control objective.

5.5 Comparison Study

To have a better observation of the technical improvements in this chapter, figures below show the simulation results that use the existing control methods that were introduced in Chapter 4. The existing control methods did not aim at the problem of time-varying parameters, so they just have one controller for the whole working process. It results in repeating stabilization work of the controller.

For simulation, we use the same parameters in Table 5.3, which is for cases with uncertain repetitively jumping parameters. The initial estimate $\theta_i(t)$ are 80% of the true values θ_i^*. The mode also switches to another one every 10 seconds.

Simulation results

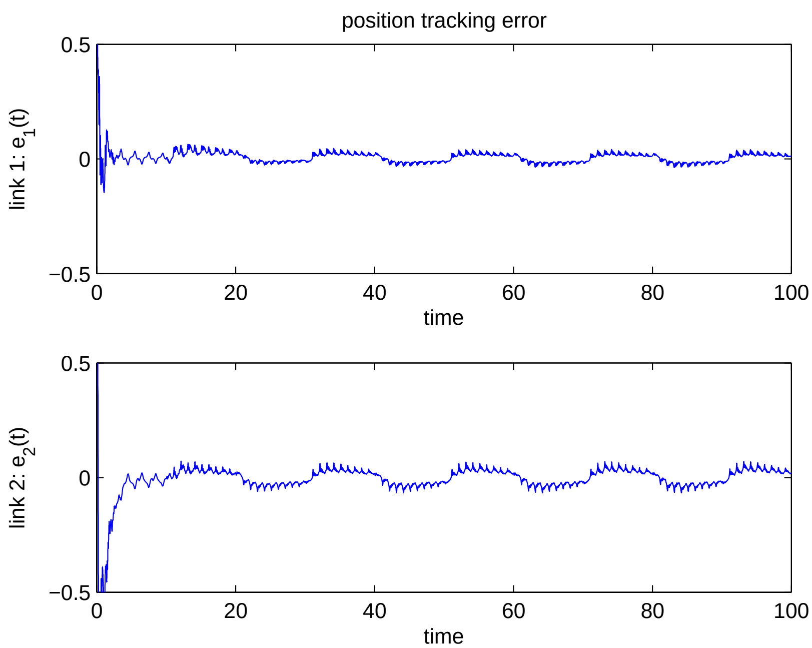

Figure 5.23: Position tracking error (Case I: the existing adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

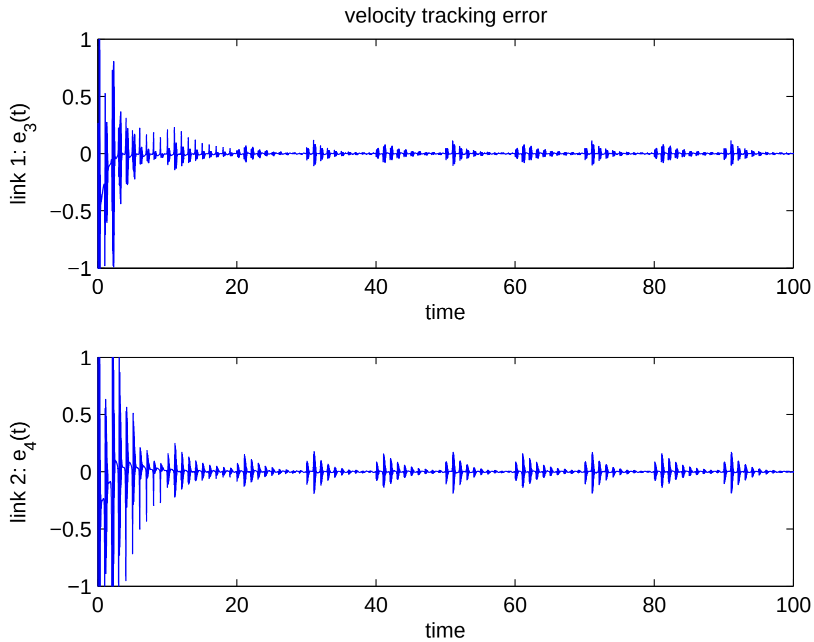

Figure 5.24: Velocity tracking error (Case I: the existing adaptive control, $\theta_i(0) = 80\%\theta_i^*$)

As shown in Figure 5.17 and 5.18, convergence progresses are repeating every time the modes changing. We can see that after the first period, the plots remain the same patterns. Compared with Figure 5.11 and 5.12, the disturbance vibration in above figures will not decrease as time goes on.

5.6 Summary

In this chapter, we first discussed the expressions of dynamic with jumping parameters and control objective in Section 5.1. Then, we studied on the nominal control scheme and adaptive control scheme in Section 5.2 and 5.3 respectively. In each of them, we present stability analysis and simulation study of two typical working cases. A comparison study of existing control scheme that introduced in Chapter 4 is also discussed in Section 5.4. The simulation of our control design shows that the controller structure and adaptive laws guarantee asymptotic tracking ability, and the disturbance of switching modes decreases as time goes on. Thus, we can conclude that the control objective is achieved.

Chapter 6

Conclusions and Future Topics

6.1 Summary and Conclusions

In this thesis, in order to solve the problems of time-varying system parameters for robot manipulators, we studied two conditions which are cases with and without joint acceleration measurements. In Chapter 5, the model of robot manipulators with time-varying and control method with joint acceleration measurement is given. Simulation results for nominal control and adaptive control were offered. We also show a comparison work of previous control methods. For all simulation works above, we use two-link planar manipulator to do the corresponding simulations, and the results illustrate the effectiveness of the adaptive control for jumping parameters.

From this research, we can obtain the following conclusions:

(i) Study of robot manipulator control with uncertain time-varying parameters is crucial, which can help reduce material fatigue, extend service life and enhance operation accuracy.

(ii) Our control designs with joint acceleration measurements achieve the control objective that the joint variables can track the desired ones asymptotically. Meanwhile, we can observe the disturbance gets smaller as time goes on.

6.2 Future Research Topics

In our thesis, we discussed adaptive control of robot manipulators with uncertain piecewise parameters, which is a promising area. However, the research area is not mature, and many challenges still need to be overcome.

(i) In our model, we did not include cases of flexible joint. It is a common model in today's robot manipulators. We can often see this structure in industrial robots. The study includes flexible joint control can increase the performance and meaning of control design.

(ii) The study of adaptive actuator failure compensation for robot manipulator control is a significant and promising topic now. It can greatly enhance the performance of control design and the robustness of the system.

(iii) In our thesis, we study the adaptive control for robot manipulators with uncertain variable parameters, which uses joint acceleration in the control design. In many cases, joint acceleration measurements are not available and hard to be accessed. So the study of adaptive control without joint acceleration measurements is also meaningful.

Part I

Bibliography

Bibliography

[1] P R Naik, J Samantaray, S K Pattanayak, B K Roy. *2-DOF Robot Manipulator Control Using Fuzzy PD Control With SimMechanics and Sliding Mode Control: A Comparative Study.* 2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE), Jun 12-13, 2015.

[2] Maximiliano Bueno Lopez, Daniel Marino Lizarazo. *A comparative analysis of adaptive visual servo control for Robots Manipulators in 2D.* 2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE), Jun 12-13, 2015

[3] H Hashimoto, K Maruyama, F Harashima. *A Microprocessor-Based Robot Manipulator Control with Sliding Mode.* IEEE Transactions on Industrial Electronics. Vol: IE-34, Issue: 1, pp. 11-18, 1987, Feb, 1987

[4] I Abdelmalek, N Gola. *A Non-Quadratic Fuzzy Stabilization and Tracking Approach to a Two-Link Robot Manipulator control.* Sixth International Conference on Intelligent Systems Design and Applications, Feb, Oct 16-18, 2006

[5] J Hao, G Tao, T Rugthum. *A Dynamic Prediction Error Based Adaptive Multiple-Model Control Scheme for Robotic Manipulators.* American Control Conference

(ACC), 2017, May 24-26, 2017

[6] A. Brahmi, M. Saad, G. Gauthier, B. Brahmi, W.-H. Zhu, J. Ghommam. *"Adaptive backstepping control of mobile manipulator robot based on virtual decomposition approach*. 2016 8th International Conference on Modelling, Identification and Control (ICMIC), Nov 15-17, 2016

[7] Z. Mao, G. Tao, B. Jiang. *Adaptive Compensation of Traction System Actuator Failures for High-Speed Trains*. IEEE Transactions on Intelligent Transportation Systems, Vol: 18, Issue: 11, pp 2950-2963, Mar 06, 2017

[8] D. Seo. *Adaptive Control for Robot Manipulator with Guaranteed Transient Performance*. 2016 IEEE 55th Conference on Decision and Control (CDC), Dec 12-14, 2016

[9] C. S. Lee, M. J. Chung, B. Lee. *Adaptive control for robot manipulators in joint and cartesian coordinates*. Proceedings. 1984 IEEE International Conference on Robotics and Automation, Mar 13-15, 1984

[10] John J. Craig, Ping Hsu, S. Shankar Sastry. *Adaptive Control of Mechanical Manipulators*. The international Journal of Robotics Research, Vol.6, No. 2, Summer 1987

[11] Q. Sang, G. Tao. *Adaptive Control of Piecewise Linear Systems: the State Tracking Case*. 2010 American Control Conference, Jun 30 - July 2, 2010

[12] T. Hsia. *Adaptive control of robot manipulators - A review*. Proceedings. 1986 IEEE International Conference on Robotics and Automation, Apr 7-10, 1986

[13] H. Wang. *Adaptive Control of Robot Manipulators With Uncertain Kinematics and Dynamics*. IEEE Transactions on Automatic Control, Vol.62, No.2, Feb, 2017

[14] J. Li, L. Liu, Y. Wang. *Adaptive hybrid impedance control of robot manipulators with robustness against environment's uncertainties*. 2015 IEEE International Conference on Mechatronics and Automation, Aug 2-5, 2015

[15] P. Hsu, M. Bodson, S. Sastry, B. Paden. *Adaptive identification and control for manipulators without using joint accelerations*. 1987 IEEE International Conference on Robotics and Automation, Mar, 1987

[16] J. Lee, P. H. Chang, M. Jin. *Adaptive Integral Sliding Mode Control With Time-Delay Estimation for Robot Manipulators*. IEEE Transactions on Industrial Electronics, Vol: 64, Issue: 8 Aug. 2017 pp 6796-6804, Apr, 2017

[17] J. E. Slotine, W. Li. *Adaptive Manipulator Control: A Case Study*. IEEE Transactions on Industrial Electronics, Vol: 33, Issue. 11 Nov 1988 pp 995-1003, Nov, 1988

[18] Romeo Ortega, Mark W. Spong. *Adaptive motion control of rigid robots: a tutorial*. Proceedings of the 27th IEEE Conference on Decision and Control, Dec 7-9, 1988

[19] C. Yuan, R. Yang, J. Na, F. Chen. *Adaptive RBFNN control of robot manipulators with finite-time convergence*. IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Oct 23-26, 2016

[20] N. Nahapetian, M. R. Jahed Motlagh, M. Analoui. *Adaptive robot manipulator control based on plant-controller model reference using soft computing and per-*

formance index analyzer. 2009 IEEE Symposium on Computational Intelligence in Control and Automation, Nov, 1988

[21] Ying-Jeh Huang, Tzu-Chun Kuo, Shin-Hung Chang. *Adaptive Sliding-Mode Control for NonlinearSystems With Uncertain Parameters*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol: 38, Issue: 2, Mar 7, 2008

[22] Xiao Xiaobo, Kang Yu , Xi Hongsheng, Wang Jun. *Adaptive tracking control for a class of uncertain nonlinear systems with Markovian jumping parameters*. Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788), Jun 15-19, 2004

[23] Jin Seuk Choi, Jong Hyun Yoon, Jong Hyeon Park, Pil Jun Kim. *A numerical algorithm to identify independent grouped parameters of robot manipulator for control*. 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Jul 3-7, 2011

[24] Chun-Yi Su, T.P. Leung, Qi-Jie Zhou. *Adaptive control of robot manipulators under constrained motion*. Adaptive control of robot manipulators under constrained motion, Dec 5-7, 1990

[25] Seung-Heui Lee, Do-Young Jeong, Ik-Soo Kim, Dong-In Lee, Dong-Yeon Cho, Min Cheol Lee. *Control of robot manipulator for storing cord blood in cryogenic environments*. Control of robot manipulator for storing cord blood in cryogenic environments, Dec 5-7, 1990

[26] Masakatsu Kemmotsu, Yasuhiko Mutoh. *Control System of Robot Manipulator using Linear Time-Varying Controller Design Technique*. Control Conference (CCC), 2014 33rd Chinese, Jul 28-30, 2014

[27] Buddhika Jayasekara, Keigo Watanabe, Kiyotaka Izumi. *Controlling a robot manipulator with fuzzy voice commands guided by visual motor coordination learning*. 2008 SICE Annual Conference, Aug 20-22, 2008

[28] Zhengui Xue, Cong Wang, Tengfei Liu. *Deterministic learning and robot manipulator control*. 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dec 15-18, 2007

[29] Stefan Kersting, Martin Buss. *Direct and Indirect Model Reference Adaptive Control for Multivariable Piecewise Affine Systems*. IEEE Transactions on Automatic Control, Vol: 62, Issue: 11, Mar 30, 2017

[30] Jiangping Li, Hongbin Ma, Chenguang Yang, Mengyin Fu. *Controlling a robot manipulator with fuzzy voice commands guided by visual motor coordination learning*. 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Jun 8-12, 2015

[31] Min Wang Anle Yang. *Dynamic Learning From Adaptive Neural Control of Robot Manipulators With Prescribed Performance*. Dynamic Learning From Adaptive Neural Control of Robot Manipulators With Prescribed Performance, Jan 6 2017

[32] Zilong Shao Gang Zheng Denis Efimov Wilfrid Perruquetti. *Modelling and control for position-controlled Modular Robot Manipulators*. IE2015 IEEE/RSJ In-

ternational Conference on Intelligent Robots and Systems (IROS), Sept 28 - Oct 2, 2015

[33] Zhong-Liang Tang, Shuzhi Sam Ge, Keng Peng Tee, Wei He. *Adaptive neural control for an uncertain robotic manipulator with joint space constraints*. International Journal of Control, Vol 89, Issue 7, Feb 25, 2016

[34] Jean-Jacques E. Slotine, Weiping Li. *On the Adaptive Control of Robot Manipulators*. The International Journal of Robotics Research, Sep 1, 1987

[35] M. Benzaoui, H. Chekireb, M. Tadjine. *Redundant robot manipulator control with obstacle avoidance using extended Jacobian method*. Control and Automation (MED), 2010 18th Mediterranean Conference, Jun 23-25, 2010

[36] John Studenny, Pierre R. Belanger. *Robot manipulator control by accelaration feedback: Stability, design and performance issues*. 1986 25th IEEE Conference on Decision and Control, Dec 10-12, 1986

[37] T.W. Martin, E. Yaz. *Robot manipulator control using adaptive computed torque technique*. Robot manipulator control using adaptive computed torque technique, Dec 10-12, 1986

[38] Mark W.Spong, Seth Hutchinson, M.Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons Inc., 2006

[39] G.Tao. *Adaptive Control Design and Analysis*. John Wiley & Sons Inc., 2003

[40] E. Lavretsky and K. A. Wise. *Robust and Adaptive Control with Aerospace Application*. Springer Inc., 2013

[41] J.J.E. Slotine and W.Li. *Applied Nonlinear Control*. Prentice Hall Inc., 1991

[42] C.Chen. *Linear System Theory and Design.* Oxford University Press Inc., 1995